The McGraw·Hill Companies

CAD/CAM: PRINCIPLES AND APPLICATIONS

3rd Edition

ABOUT THE AUTHOR

P Nageswara Rao received his BE degree in Mechanical Engineering from the Government Engineering College, Anantapur (Sri Venkateswara University, Tirupati) in 1970, ME degree from BITS, Pilani in 1973 and PhD in 1981 from IIT Delhi.

Currently, he is working in the Department of Industrial Technology, University of Northern Iowa, USA. Dr Rao previously worked in MARA University of Technology, Shah Alam, Malaysia, in the Department of Mechanical Engineering. Before that, he worked with IIT Delhi from 1975 till 1997, the final position being Professor in Mechanical Engineering Department between 1990 and 1997. Prior to joining IIT Delhi, he also worked at BITS Pilani for a period of two years.

His active areas of research are Manufacturing Engineering, CNC, CAPP, CAD/CAM, CIM, Tool Design, Maintenance and Injection Moulding. He is the author and co-author of more than 190 papers published in various national and international journals and conferences.

He has wide interaction within the industry through the process of consultancy work and conducting continuing education programmes on various aspects related to manufacturing.

Dr P N Rao has authored five books, beside the present one, all published by Tata McGraw Hill Education, New Delhi.

- *Numerical Control and Computer Aided Manufacturing*, 1985 (Kundra and Tewari as co-authors)
- *Manufacturing Technology: Vol. 1: Foundry, Forming and Welding*, 3rd Ed 2009
- *Computer Aided Manufacturing*, 1993 (Kundra and Tewari as co-authors)
- *AutoCAD 14 for Engineering Drawing Made Easy*, 1999
- *Manufacturing Technology: Vol. 2: Metal Cutting and Machine Tools*, 2nd Ed 2009

He is also a co-editor of *Emerging Trends in Manufacturing* (Proceedings of the 12th All India Machine Tool Design and Research Conference, 1986) published by Tata McGraw Hill Education, New Delhi.

His biographical information has appeared in *Marquis Who's Who in The World, Marquis Who's Who in Finance and Industry, Marquis Who's Who in Science and Engineering, Marquis Who's Who in America, Marquis Who's Who in American Education*, published by Reed Elsevier Inc., New Providence, USA, and *Dictionary of International Biography and Who's Who in Asia and the Pacific Nations*, published by The International Biographical Centre, Cambridge, UK.

The McGraw·Hill Companies

CAD/CAM: PRINCIPLES AND APPLICATIONS

3rd Edition

P N Rao
Professor
Department of Industrial Technology
University of Northern Iowa
Iowa, USA

Tata McGraw Hill Education Private Limited
NEW DELHI

McGraw-Hill Offices

New Delhi New York St Louis San Francisco Auckland Bogotá Caracas
Kuala Lumpur Lisbon London Madrid Mexico City Milan Montreal
San Juan Santiago Singapore Sydney Tokyo Toronto

Tata McGraw Hill Education Pvt. Ltd.

Published by Tata McGraw Hill Education Private Limited,
7 West Patel Nagar, New Delhi 110 008.

CAD/CAM: Principles and Applications, 3e

Copyright © 2010, 2004, 2002 by the Tata McGraw Hill Education Private Limited

No part of this publication may be reproduced or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise or stored in a database or retrieval system without the prior written permission of the publisher. The program listings (if any) may be entered, stored and executed in a computer system, but they may not be reproduced for publication.

This edition can be exported from India only by the publishers,
Tata McGraw Hill Education Private Limited.

ISBN (13): 978-0-07-068193-4
ISBN (10): 0-07-068193-7

Managing Director: *Ajay Shukla*
Head—Higher Education Publishing: *Vibha Mahajan*
Manager— Sponsoring: *Shalini Jha*
Assoc Sponsoring Editor: *Suman Sen*
Development Editor: *Devshree Lochab*
Executive—Editorial Services: *Sohini Mukherjee*
Sr Production Executive: *Suneeta S Bohra*
General Manager: Marketing—Higher Education & School: *Michael J Cruz*
Dy Marketing Manager: SEM & Tech Ed: *Biju Ganesan*
Asst Product Manager: *Amit Paranjpe*
General Manager—Production: *Rajender P Ghansela*
Asst General Manager—Production: *B L Dogra*

Information contained in this work has been obtained by Tata McGraw Hill, from sources believed to be reliable. However, neither Tata McGraw Hill nor its authors guarantee the accuracy or completeness of any information published herein, and neither Tata McGraw Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that Tata McGraw Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

Typeset at Bharati Composers, D-6/159, Sector-VI, Rohini, Delhi 110 085, and printed at Ram Book Binding, C-114, Okhla Industrial Area, Phase – I, New Delhi 110 020

Cover Printer: Rashtriya Printers

RAXQCRCZDXCQA

Dedication to

My elder brother
Parankusam
who appreciated and supported the importance of knowledge
and reading throughout his life

The McGraw·Hill Companies

PREFACE

It is my pleasure to note that the earlier editions of the book have gained wide acceptance among the academic community. The second edition of this book was published in 2004. Since then, many new software like AutoCAD 2010 and Invertor 2010 have come out in the field of CAD/CAM. As it is a constantly evolving branch of learning, so the text covering this subject needs constant updation and revision. Hence, in order to be in congruence with the contemporary technological advancements plus current syllabi and competitive requirements, revision of the text had become crucial.

The original intention of this book was to provide a user viewpoint of CAD/CAM such that the application aspect is covered in greater detail. The current syllabi of most of the Indian universities have significant coverage of CAD principles as a part of their courses. Many suggestions received have alluded to this fact, and as a result these have been substantially expanded in this edition.

The modifications done in this edition are listed below:

Chapter 23 on Computer Aided Quality Control is a new chapter added with details on Inspection and testing, CMM, Non-contact inspection, SQC, SPC, TQM, Six Sigma, Integration of CAQC.

Chapter 3 Additional details on data models, engineering data management system, expanded coverage of clipping methods, hidden surface algorithms, and colour and shading procedures

Chapter 4 Expanded coverage on wireframe modelling, parametric representation of curves and surfaces including b-splines, NURBS, surface of revolution and curve fitting techniques; additional topic on solid representation methods including CSG and b-rep

Chapter 8 Completely revised with additional topics on trusses, beams and plane stress methods in FEA

Chapter 12 Introduction to adaptive control added

Chapter 13 More examples on part programming including simulations were added to improve the understanding of part programming

Chapter 16 More examples of APT programs are added and CAM programming is updated with Mastercam X3

Chapter 18 Enhanced classification and coding, coding systems. Added MICLASS, DCLASS, CODE and KK-3, Enhanced PFA, rank order clustering method, cellular manufacturing, and cell formation methods

Chapter 19 Added details on production planning, capacity planning and shop floor data collection

In addition to these major additions, the book has been brought up-to-date making the necessary changes throughout the book. With these major modifications, I hope the book serves a much wider spectrum of people.

Salient Features

The topics are well structured and bifurcated into sub-topics to increase the understanding. Technical terms have been used without conciliating on the reader-friendly attribute of the text. A methodical approach has been followed by first outlining the objective in the beginning of each chapter. The concepts in each chapter are explained in a very comprehensive and coherent manner substantiated effectively with the help of solved examples, figures and photographs; and exercise problems. The contents of the chapter are summarised at the end which enables quick and handy revision. Thus, this edition has been thoroughly revised and updated in order to remain in conformity with the course requirements and provide the recent and contemporary technological progress in the respective areas.

Organisation of the Book

The book is divided into five parts containing 24 chapters in all. **Chapter 1** is an introductory chapter which discusses the basics of CAD and CAM.

This is followed by ***Part I*** which is on hardware and software components and contains Chapters 2 and 3. **Chapter 2** is on CAD/CAM hardware, and **Chapter 3** is about computer graphics.

Part II which is on design of industrial products has **Chapters 3 to 8** which discuss geometric modelling, CAD standards, drafting system, modelling systems, and finite element analysis.

Part III discusses manufacturing aspects of industrial products, and contains Chapters 9 to 16. Computer Numerical Control (CNC) is introduced in **Chapter 9**, followed by CNC hardware basics, CNC tooling, CNC machine tools and control systems, and CNC programming in **Chapters 10, 11, 12 and 13** respectively. Turning-centre programming is explained in **Chapter 14. Chapters 15 and 16** thereafter deal with advanced part-programming methods and computer-aided part programming in that order.

Part IV describes the role of information systems and has Chapters 17 to 19. Information requirements of manufacturing are dealt with in **Chapter 17. Chapter 18** discusses group technology and computer aided process control. Production planning and control are explained in **Chapter 19**.

Finally, ***Part V*** is on integration of manufacturing systems and contains Chapters 20 to 24. Communications, including communication methods, direct numerical control and communication standards are examined thoroughly in **Chapter 20**. Material-handling systems and flexible manufacturing systems are explained in **Chapters 21 and 22** respectively. Computer aided quality control is discussed in **Chapter 23**. Lastly, Computer Integrated Manufacturing (CIM) is taken up in **Chapter 24**.

Web Supplements

The web supplements can be accessed at http://www.mhhe.com/rao/cadcam/3e and contain the following material:

For Instructors

- Solution Manual
- Chapter wise PPT's

For Students

- Interactive Objective Type Questions
- Links to Reference Material
- Sample Chapter

Feedback

Readers are once again requested to send comments and suggestions, which will be taken care of in future editions.

P N Rao

Publisher's Note

Do you have a feature request? A suggestion? We are always open to new ideas (the best ideas come from you!). You may send your comments to *tmh.mefeedback@gmail.com* (kindly mention the title and author name in the subject line). Piracy-related issues may also be reported.

The McGraw·Hill Companies

The McGraw·Hill Companies

ACKNOWLEDGEMENTS

A book of this magnitude requires a large effort, which in many parts has been contributed by my many colleagues in various forms. It is my earnest duty, therefore, to acknowledge such contributions.

Many of my colleagues have significantly contributed in developing my ideas in CAD/CAM during their long association at the Indian Institute of Technology, New Delhi; Universiti Technology Mara, Malaysia; and University of Northern Iowa, USA. In particular, P S Nageswara Rao, A Subash Babu, N K Tewari, T K Kundra, Robert Bell, S Hinduja, U R K Rao, S R Deb, S Darius Gnanaraj, Salim, Julie Zhang, Ali Kashef and Nilmani Pramanik deserve a special mention in this regard.

I would also like to thank the Department of Electronics, Government of India and Bharat Heavy Electricals Ltd. (Basheer Ahmad, S N Daga, S Biswas and others) for having associated with me on various projects related to CAD/CAM.

I am grateful to the large number of postgraduate and research students at the Indian Institute of Technology, New Delhi, who worked with me and contributed a large amount of information presented in this book. A special note of appreciation is due to Jasthi Siva Rama Krishna, A V S R K Prasad and G Manidhar for their assistance in writing the part on CAPP.

A large number of companies have liberally provided information and illustrations for use in the book. I have made an effort to provide those references along with the relevant individual illustrations. However, a mention may be made about them here:

Amada Engineering, Japan
Boehringer Werkzeugmaschinen GmbH, Germany
Bridgeport Machines, USA
Chiron-Werke GmbH & Co., Germany
Dixi S A, Switzerland
DMG Asia Pacific Pte. Ltd., Singapore
Dr Johannes Heidenhain GmbH, Germany
EMAG Maschinenfabrik GmbH, Germany

Georg Fischer FMS, Switzerland
Huller Hille GmbH, Germany
IBM
Jobs S PA, Italy
Makino Milling Machine Co. Ltd., Japan
Mikron, Switzerland
Trimos, Switzerland
Virtual Gibbs, USA
Yamazaki Mazak Corporation, Japan

I am grateful to the authorities of the University of Northern Iowa, Cedar Falls, USA, who provided the excellent environment, opportunities and facilities to undertake this task.

I express my appreciation for all the reviewers, especially to M Ramulu, who provided valuable suggestions for revising parts of the manuscript. Their names are given below.

V K Jain	Indian Institute of Technology (IIT), Kanpur Kanpur, Uttar Pradesh
P M Pandey	Indian Institute of Technology (IIT), Delhi New Delhi
Vinay Marvel	Stani Memorial College of Engineering Technology (SMCET), Jaipur Jaipur, Rajasthan
Sankha Deb	Indian Institute of Technology (IIT), Kharagpur Kharagpur, West Bengal
S S Roy	National Institute of Technology (NIT), Durgapur Durgapur, West Bengal
T K Jana	Haldia Institute of Technology (HIT), Haldia Haldia, West Bengal
H K Raval	Sardar Vallabhai National Institute of Technology (SVNIT), Surat Surat, Gujarat
R Manu	National Institute of Technology (NIT), Calicut Calicut, Kerala
K Sankaranarayanasamy	National Institute of Technology (NIT), Trichy Trichy, Tamil Nadu
A Venugopal	National Institute of Technology (NIT), Warangal Warangal, Andhra Pradesh
V V N Satya Suresh	Mahatma Gandhi Institute of Technology, Hyderabad Hyderabad, Andhra Pradesh

It is a pleasure to express my heartfelt gratitude to my family members who have borne long hours of inconvenience during the preparation of the manuscript. I am also thankful to many of my students who have learned this subject from me, for their probing questions and comments through which the subject could be brought to this form.

P Nageswara Rao

CONTENTS

Part - I Hardware and Software Components

Part - II Design of Industrial Products

Part - III Manufacturing Aspects of Industrial Products

Part - IV Role of Information Systems

Part - V Integration of Manufacturing Systems

1

INTRODUCTION

Objectives

Computers are being used in all facets of our life. In this chapter, the application of computers for discrete manufacturing will be discussed. Overviews of computer applications that will be affecting the manufacturing industry are presented. After completing the chapter the reader will be able to

- Understand the various spheres of manufacturing activity where computers are used
- What is meant by product cycle with the differences between the conventional and computer-based manufacturing systems
- Definitions of various computer-based applications
- Discuss various facets of the design process
- Computer Aided Design and its applications
- Various types of manufacturing organisations
- Computer Aided Manufacturing and its applications
- Meaning of Computer Integrated Manufacturing

1.1 COMPUTERS IN INDUSTRIAL MANUFACTURING

With an increase in the need for quality manufacturing along with the factors of short lead times and short product lives and increasing consumer awareness regarding the quality of the product, it is becoming increasingly important for the manufacturers to initiate steps to achieve all these. View this against the fact that the developments in microelectronics in the recent past have made higher computational ability available at a low cost. Thus, it becomes imperative that manufacturing has to take advantage of the availability of low-cost yet more powerful computers. Hence, the use of Computer Aided Engineering, particularly for mechanical industries, should now be a realisable goal.

The role of computers in manufacturing may be broadly classified into two groups:

1. Computer monitoring and control of the manufacturing process
2. Manufacturing support applications, which deal essentially with the preparations for actual manufacturing and post-manufacture operations

In the first category are such applications where the computer is directly interfaced with the manufacturing apparatus for monitoring and control functions in the manufacturing process. For example, in a continuous process industry (chemical processing), a number of process parameters may be monitored. With built-in specifications in the computer memory, suitable actions may be initiated by the computer for the purpose of regulating the process. Alternately, a human operator looking at the process may initiate the controlling action. This is not restricted to the chemical process industry but could include any other type as well, such as packaging. The subject matter in this book may not go into this specific area in a great measure though some aspects of control will be dealt with in connection with the numerical-control machine tools and other material-handling equipment.

In the second category, are all the support functions that computers can provide for the successful completion of manufacturing operations. The types of support that can be envisaged are the following:

CAD—Computer Aided Design The use of computer methods to develop the geometric model of the product in three-dimensional form, such that the geometric and manufacturing requirements can be examined.

CADD—Computer Aided Design and Drafting Combining the CAD function with drafting to generate the production drawings of the part for the purpose of downstream processing.

CAE—Computer Aided Engineering The use of computer methods to support basic error checking, analysis, optimisation, manufacturability, etc., of a product design.

CAM—Computer Aided Manufacturing Generally refers to the computer software used to develop the Computer Numerical Control part programs for machining and other processing applications.

CAPP—Computer Aided Process Planning The use of computers to generate the process plans for the complete manufacture of products and parts.

CATD—Computer Aided Tool Design Computer assistance to be used for developing the tools for manufacture such as jigs and fixtures, dies, and moulds.

CAP—Computer Aided Planning The use of computers for many of the planning functions such as material requirement planning, computer aided scheduling, etc.

CAQ—Computer Aided Quality Assurance The use of computers and computer-controlled equipment for assessing the inspection methods and developing the quality control and assurance functions.

CAT—Computer Aided Testing Refers to the software tools that can take a system through its various phases of operations and examine the response against the expected results.

The use of computers in manufacturing is a methodological approach to the enterprise in order to improve industrial performance. This requires a range of broad technologies that have become realisable, thanks to the development in computer technology. The total components that can be assumed to consist of a number of interlinked domains are shown in Fig. 1.1.

1.2 DESIGN PROCESS

Design is an activity that needs to be well organised and should take into account all influences that are likely to be responsible for the success of the product under development. A product can range from a single component, which is functional in itself like a wrench to the assembly of a large number of components all

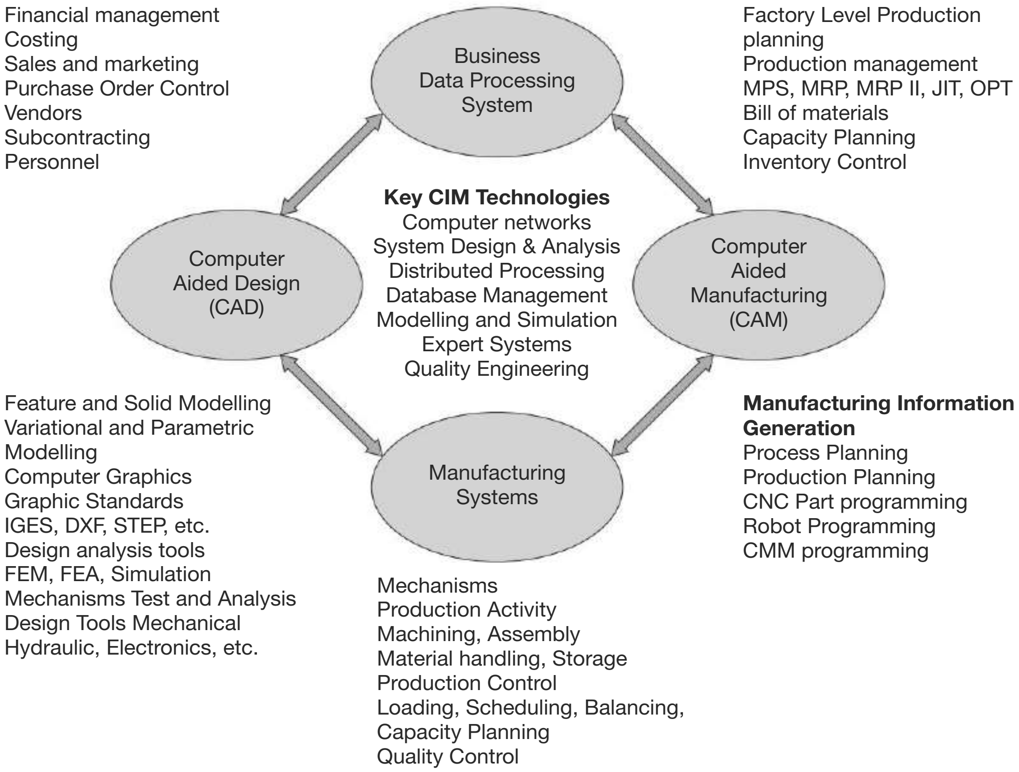


Fig. 1.1 *The influence of computers used in manufacturing environment*

of which will contribute to the functioning of the part, such as an automobile engine. The complexity of the design process increases with the number and diversity of components present in the final part.

Since there are such a large number of influencing factors, it is impossible to specify a design procedure for each component. The various faculties that are responsible for a successful product can be classified under two headings as follows:

Product Engineering

- Product functions
- Product specifications
- Conceptual design
- Ergonomics and Aesthetics
- Standards
- Detailed design
- Prototype development
- Testing
- Simulation

- Analysis
  - Strength
  - Kinematics
  - Dynamics
  - Heat
  - Flow
  - Design for manufacture
  - Design for assembly
- Drafting

Manufacturing Engineering

- *Process Planning*
  - Process sheets
  - Route sheets
- *Tooling*
  - Cutting tools
  - Jigs and fixtures
  - Dies and moulds
- *Manufacturing Information Generation*
  - CNC part programs
  - Robot programs
  - Inspection (CMM) programs
- *Production Organisation*
  - Bill of materials
  - Material requirement planning
  - Production planning
  - Shop-floor control
  - Plant simulation
- *Marketing and Distribution*
  - Packaging
  - Distribution
  - Marketing

Ideally, the designer should consider all these factors while finalising the design. It is impossible for a single individual to carry out all these functions, except in the case of simple parts. For complex systems, the product design function needs to be carried out by a team of specialists who have specified knowledge and experience in the individual areas as mentioned above. As identified earlier, the design process (Fig. 1.2) goes through well-structured stages to reach the stage of actual part production.

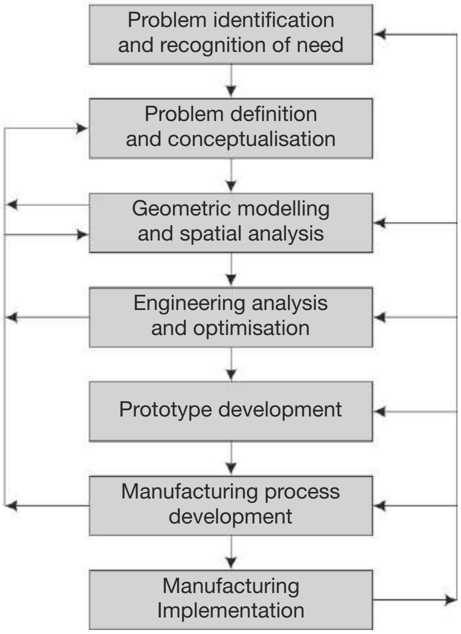


Fig. 1.2 *Stages in the design process*

1.2.1 Problem Identification

The starting point of the design process is the identification of the needs of an unsatisfied demand for a particular product or conceptually a new idea to start a fresh demand. At this stage, it is possible to identify some of the basic questions related to the product such as who, what, where, when, why and how many should be answered with fair accuracy. In order to provide answers to the above questions, the design team may have to explore a number of sources and methods as shown in Fig. 1.3.

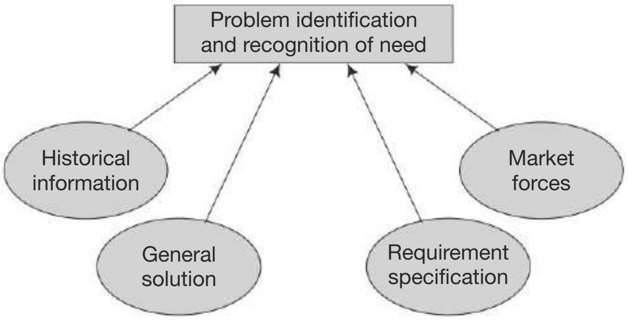


Fig. 1.3 *Processes involved in the problem-identification stage*

Historical Information This is related to the already existing information collected through the literature, marketing surveys, etc. This should be able to answer questions like

- The current technology
- Existing solutions (even competitor's product details)

Requirement Specification A clear definition of the requirements is specified at this stage. This helps in understanding the product from the current business practices and manufacturing resources of the plant. This also helps in understanding short-term or long-term potential of the new product introduction.

Market Forces Before going ahead with product design, it is also essential to consider the various market forces that will affect the product in one way or the other.

General Solutions Having identified all the requirements and the controlling factors, it would be possible to specify a general solution, which will be broad and would not contain too many details. This can be done by resorting to past designs, engineering standards, technical reports, catalogues, handbooks, patents, etc. This helps in its further evaluation and refinement at a later stage.

1.2.2 Problem Definition

The next stage in the design process is the clear definition of the problem and coming up with all possible ideas for solutions. This stage may be carried out in various forms of components as shown in Fig. 1.4.

Preliminary Design The necessary elements that will be important for the design process are identified at this stage. This basically identifies the likely difficulties to be faced in the design process as well as identify some important design elements that help in the design process. This is used further in developing the preliminary design solutions.

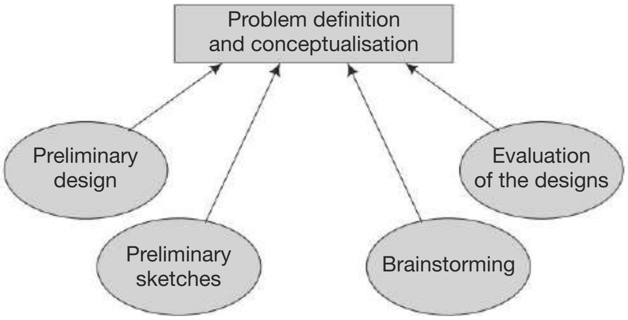


Fig. 1.4 *Processes involved in the problem-definition stage*

Preliminary Sketches The basic solutions that have been identified in the earlier stage are to be detailed with the necessary sketches to examine their suitability for finalisation. Also, some notes related to the design may be added to the sketches to clarify some ideas that cannot be shown by sketching alone. The effort here still is largely manual.

Brainstorming This is basically a group solving technique, where each one of the design team members spontaneously comes up with ideas. It is necessary to collect all the ideas during these sessions that are then be further processed to identify a final solution.

Evaluation of the Designs A number of concepts have been identified in the previous stage. It is necessary to evaluate each of the choices in terms of feasibility, cost, ergonomics and human factors, environment, maintainability, etc. At this stage, it should be possible to identify the final design based on all the factors such as market requirements, technical feasibility, economics, manufacturing expertise and resources available.

1.2.3 Geometric Modelling

In the next stage, the identified solutions are further explored for the final design solution. At this stage, it is necessary to employ computers at the various phases as shown in Fig. 1.5.

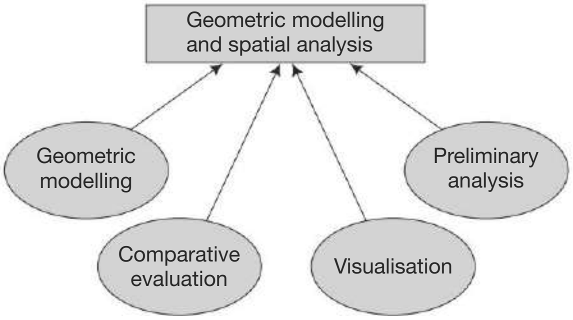


Fig. 1.5 *Geometric-modelling stage in the design process*

Geometric Modelling Geometric modelling provides a means of representing part geometry in graphical form. In fact, in many software packages, geometric modelling constitutes the most important and complex part. It is important that the geometric model generated should be as clear and comprehensive as possible so that the other modules of the modelling and manufacturing system are able to use this information in the most optimal way.

In view of such varied applications, the geometric-modelling technique used would have to provide all such facilities for interaction. The modelling system should be able to describe the parts, assemblies, raw material used, and the manufacturing requirements. From geometric models (of parts, assemblies, stock and tools), it would be possible to obtain manufacturing, assembly and inspection plans and command data for numerically controlled machine tools.

Visualisation One of the important requirements of modelling is the ability to visualise the part in actual service condition. This requires a range of shading facilities along with the ability to give various colours and surface textures to the part. This would allow the part to be visible in actual condition without really making the prototype.

Preliminary Analysis This will allow for the simple analysis techniques such as volumes and masses, inertia, spatial analysis, etc. Also, the human factors and ergonomic requirements can be analysed at this stage.

Comparative Evaluation Based on the data collected so far in terms of modelling, basic analysis and other factors, it would be possible to rate the various options in terms of technical feasibility, market acceptability and overall economics. This would allow for finalising the design, which can be conducted further thorough analysis in the next stage.

1.2.4 Engineering Analysis

In this stage of design process, a thorough analysis of the product is carried out to get as much of information as possible before committing to final manufacturing. For this purpose, a large number of computer aids are available as shown in Fig. 1.6. The analysis stage is basically an iterative one with modification to the geometric model being carried out until the desired end result is achieved.

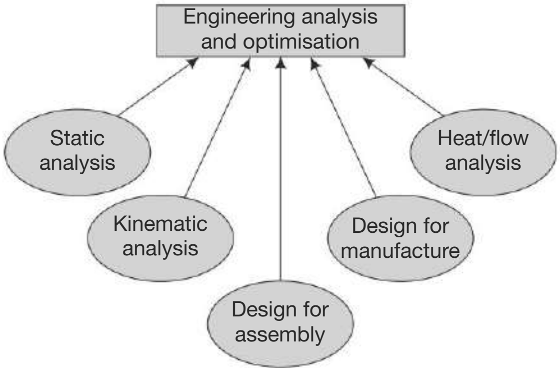


Fig. 1.6 *Analysis stage in the design process*

Strength Analysis It is necessary to obtain the stresses and strains in the component when it is in service. Analytical methods are feasible for simple shapes and configurations. However, for complex shapes, it is necessary to use finite element analysis methods. Finite Element Analysis (FEA) breaks a model into small uniform elements and applies the loading and boundary conditions for each of the elements. The stresses and strains thus derived are more representative of the final values.

Kinematic Analysis Many of the parts developed will have a number of components, some of which will also have relative motion requirements under service. Geometric modelling as described earlier will not be of much assistance in this regard. Kinematic-analysis systems allow the user to optimise the product performance by providing a fundamental understanding of how a design will perform in its real-world environment. This understanding, such as how an assembly will behave in motion and how the individual parts move under extreme conditions, provides the necessary insight for creating the best possible product design.

Dynamic Analysis For certain equipment that is likely to be operating under high speeds, it is necessary to extend the above system for dynamic conditions. Using this, engineers can evaluate the designs for vibration requirements by performing dynamic time, frequency, random, and shock-response simulations.

Heat/Flow Analysis This would allow for the evaluation of the part in terms of the heat-transfer analysis by evaluating the temperature, thermal stresses and the like. Similarly, it is also possible to evaluate the flow characteristics by employing the FEA techniques.

Design for Manufacture and Assembly One of the analysis methods that can be carried out in the early stages is the design for manufacture and assembly. This allows for a reduction of the assembly costs and component count along with a reduction of the overall costs while improving the reliability of the product. Boothroyd and Dewhurst have developed the methodologies and computer solutions for the same. The methodology to be adopted is shown in Fig. 1.7. The following three principles are recursively applied to all the assemblies to develop a low-cost assembly.

- During the operation of the product, does the part move relative to all other parts already assembled?
- Must the part be of a different material than or be isolated from all other parts already assembled?
- Must the part be separate from all other parts already assembled, because otherwise necessary assembly or disassembly of other separate parts would be impossible?

Each of the components is further analysed to see if the selected material and manufacturing process is the best or could a better, low-cost option be obtained. The concept of features helps in this process.

Some general guidelines that one may have to consider while carrying out the manufacturability analysis are as follows:

- Use standard processes and methods.
- Limit the manufacturing processes to those already available and that the plant has expertise in.
- Reduce the variety of manufacturing processes used.

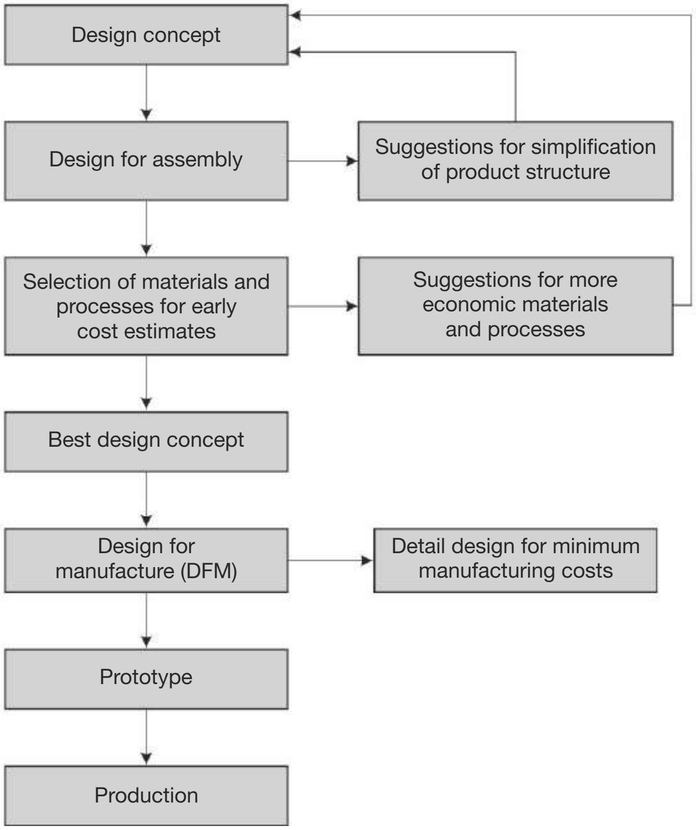


Fig. 1.7 *Methodology of design for manufacture and assembly*

- Use standard (off-the-shelf) components in the design.
- Provide liberal tolerances such that overall manufacturing cost could be lowered.
- Use materials that have better manufacturability.
- Since many of the secondary operations require additional cost, they should be minimised or avoided.
- The design process should be commensurate with the level of production expected of the part.
- When a particular process is identified, exploit the special features of the process to get better economies.

1.2.5 Prototype Development

Before committing the design to manufacture, it is also essential to carry out some physical tests on the part. This will be in addition to the computerised analysis carried out using various facilities as outlined in the earlier stages. The possible components in this stage are shown in Fig. 1.8. Using conventional methods for developing the physical models is often time consuming and expensive.

Rapid Prototyping (RP) Physical models which directly represent the component are much better than pure computer visualisation. A real component helps in obtaining the necessary information for manufacturing information generation as outlined later. Rapid prototyping is a means through which the product geometry as modelled in the earlier stages is directly utilised to get the physical shape of the component.

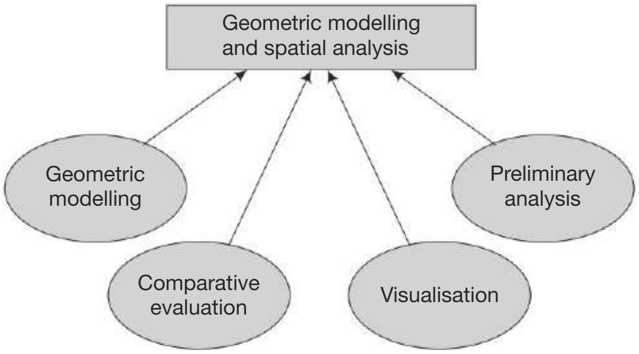


Fig. 1.8 *Prototype-development stage in the design process*

(a) Test and Evaluation In the earlier stage, the component has been designed to take care of the stresses likely to come when the part is in the real world. Sometimes, it may be necessary or desirable to carry out actual testing to verify the computer simulations. The actual prototype developed earlier can be utilised for this purpose.

(b) Design Refinement Having identified the final solution for the design, this stage helps in fine-tuning the design. A careful evaluation of each feature and capability embedded in the design is to be carried out in this stage. There will not be any major changes done at this stage, but only minor modifications and enhancements.

(c) Working Drawings These refer to the final hard copies of the drawings of the components and assemblies describing the dimensional details along with the assembly procedures. The main function expected to be served by this is to provide information for the downstream applications in the manufacturing.

1.2.6 Manufacturing Process Development

After finalising the product design, it is important to move the product to the manufacturing stage. Already the geometric models of the individual components as well as the assemblies are available both in electronic form as well as hard-copy form from the earlier stages. They are utilised for developing the necessary manufacturing processes again utilising computers to their fullest extent. The typical components present are shown in Fig. 1.9.

Process Planning Process planning is the function of determining exactly how a product will be made to satisfy the requirements specified at the most economical cost. The importance of a good process plan cannot be over-emphasized, particularly for mass production. A few minutes used to correct an error during process planning can save large costs that would be required to alter the tooling or build new tooling. Particularly in mass production, any small time saved per component would in the end mean large money saving.

Tool Design Since the geometric model is available; it is possible to develop tooling designs such as fixtures, injection mould cavities, mould cores, mould bases, and other tooling. For example, the cavity plate in the case of a simple injection mould can be conceived as a Boolean subtraction of the part at the parting line from a mould housing (a rectangular block) after providing for the necessary shrinkages and draft angles.

For example, some of the facilities provided in the software Pro/ENGINEER tool design option from Parametric Technologies Corporation are

- Create multi-cavity layout configurations, including single, rectangular, circular and variable
- Access the online components catalogue; then select, assemble and modify mould parts and mould bases, including DME, HASCO, Futaba, National, DMS and Progressive
- Automate the placement, trimming and clearance of holes for over 9,000 different ejector pins

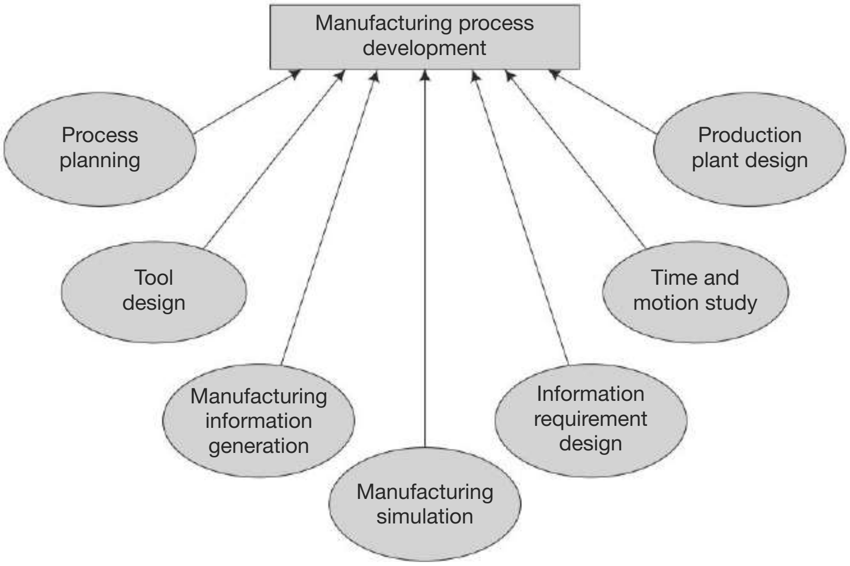


Fig. 1.9 *Manufacturing-process-development stage in the design process*

- Select and quickly assemble user-customised injection-moulding machine mock-ups in order to check for possible interference
- Create waterlines and instantly analyse for thin wall conditions
- Simulate the mould-opening sequence, including interference-checking
- Produce runners, gates and sprues instantly
- Dramatically shorten time-to-develop mould inserts, casting cavity and pattern geometry, while reducing modelling complexity
- Compensate for model shrinkage by enabling dimensioning or scaling of the entire model in X, Y and Z
- Eliminate the need to translate between part design, mould design and NC, due to seamless integration with other Pro/ENGINEER applications

Manufacturing Information Generation This aspect relates to the various part programs required during the manufacturing. They could be directly generated using the part model data. Some examples are CNC Part programs, robot programs and inspection (CMM) programs.

Software, often called the CAM systems or more appropriately called CNC programming systems, are used to develop the NC part programs directly from the CAD data. This method of programming allows the user to visualise the part on-screen during each phase of the programming process. Being able to verify each tool path in the computer instead of at the machine, reduces errors and saves valuable machine time. The computer calculates the mathematics involved in the part program and the post processor generates the G-code program, both of which produce more accurate and error-free part programs.

Manufacturing Simulation Many times the proving of the NC part programs has to be carried out using the actual CNC machine tools, which is an expensive and often time-consuming option. In such cases, it is desirable to carry out the actual simulation of machining on the computer screen, which saves a large amount of time and money.

Information Requirement Design This aspect relates to information pertinent to the manufacturing of the part that could be directly generated using the part model data. Some examples are bill of materials, material-requirement planning, production planning, shop-floor control and plant simulation.

Time and Motion Study This aspect needs to be done to see that the product's manufacturing cycle is optimised. Some of the times to be optimised are the material handling, manufacturing time, component and machine tool set-up time, etc.

Production Plant Design The actual plant to produce the design for the production volumes forms part of this.

As described above, engineers have been striving to improve productivity of the manufacturing process by optimising the various aspects of the process by taking all the information of relevance. In view of that, each process has to consider a large amount of information which will be difficult to organise through manual effort. Hence the application of computers in design and manufacturing of products is very wide and is ever expanding. Figure 1.10 indicates the various aspects of the design cycle as shown in Fig. 1.2, getting the benefit of computer assistance. Presently, many of the solutions as explained earlier are available from different sources and are often to be integrated through the help of data translators or neutral data formats. The day may not be far off when a truly totally integrated system from art to part in its true sense will be available.

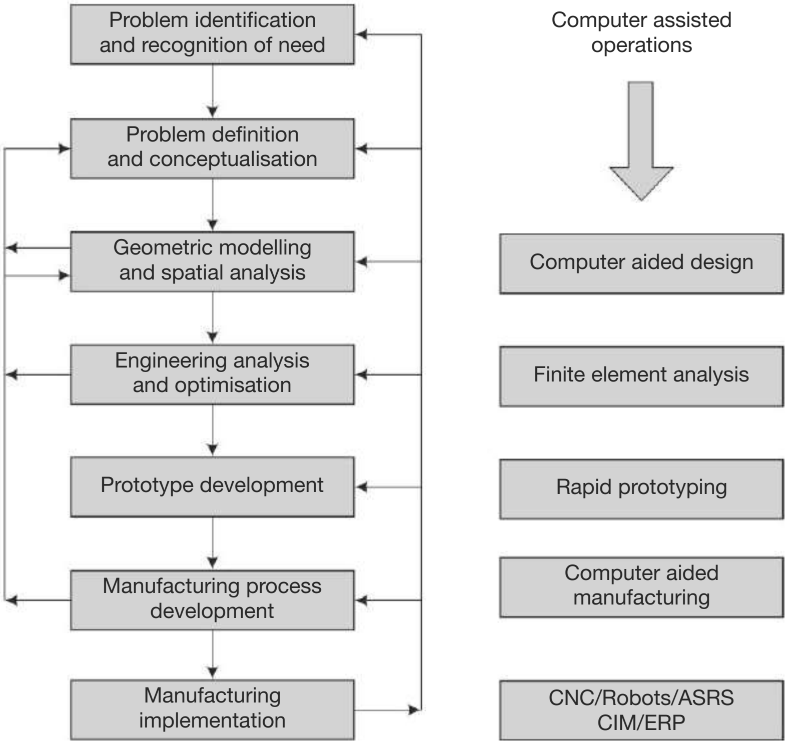


Fig. 1.10 *Computer assistance for the design cycle*

The product cycle with the computer assistance is shown in Fig. 1.11. One of the most important components for getting the various benefits associated with computer applications in manufacturing is the common databases associated with all aspects of manufacturing as shown in Fig. 1.12. In fact, all the modules in the CAM would actually be sharing the database created in any module. Any module would be able to modify the data as required for that particular application. This approach helps in reducing the work involved in maintaining the product database and at the same time includes the latest modifications for any aspect related to manufacturing. In contrast to the common database approach, it is possible that sometimes individual modules in the production aspects may be taken from different vendors, in which case care needs to be taken to see that information is properly transmitted between the modules and the data updating in all the modules takes place properly at the right time.

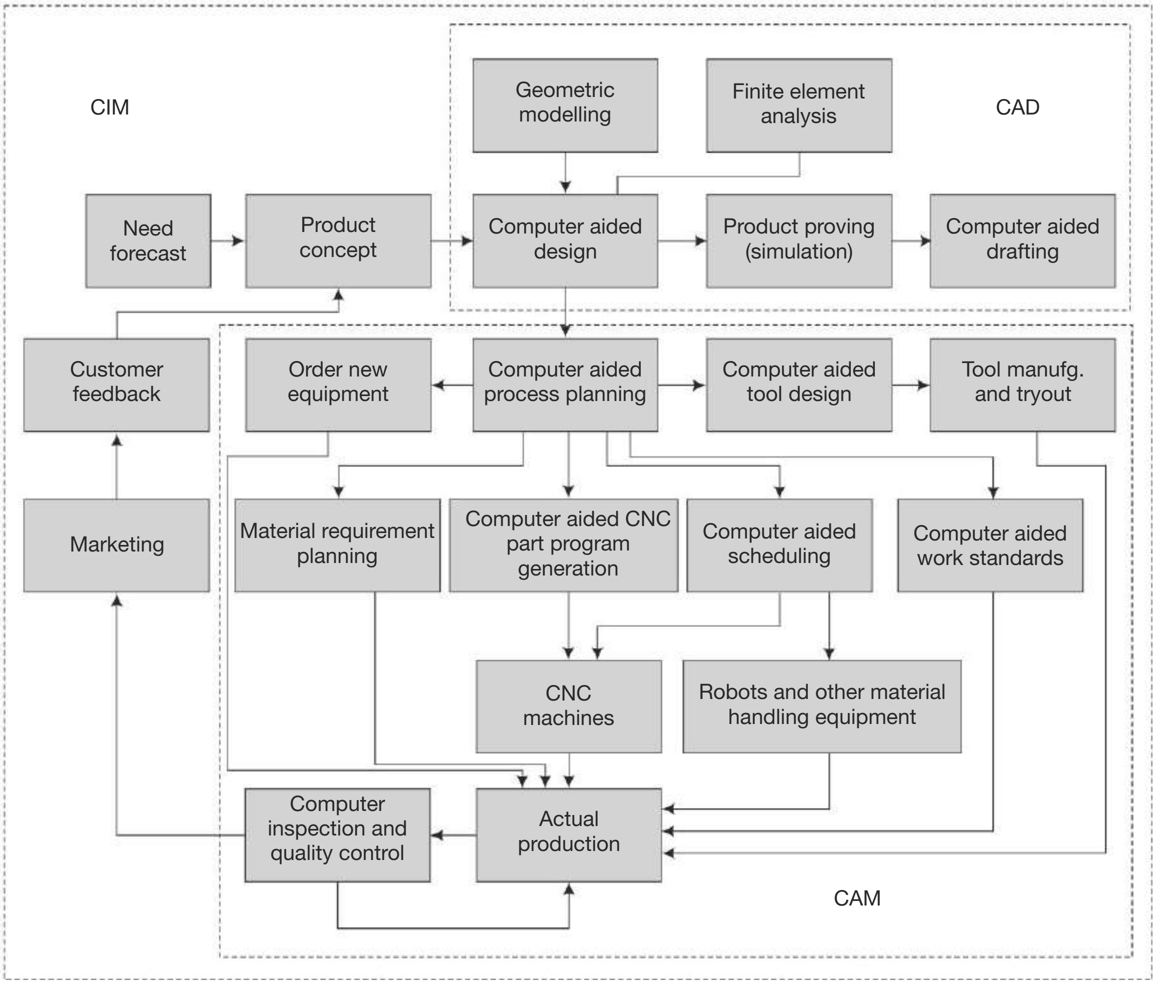


Fig. 1.11 *The product cycle in a computerised manufacturing environment*

1.3 COMPUTER AIDED DESIGN (CAD)

Computer aided design thus utilises the computer as a tool for all functions that are involved in the design process. The main functions that would utilise the computer are

- Layout design for the overall assembly
- Individual component modelling
- Assembly modelling
- Interference and tolerance stack checking
- Engineering drawings

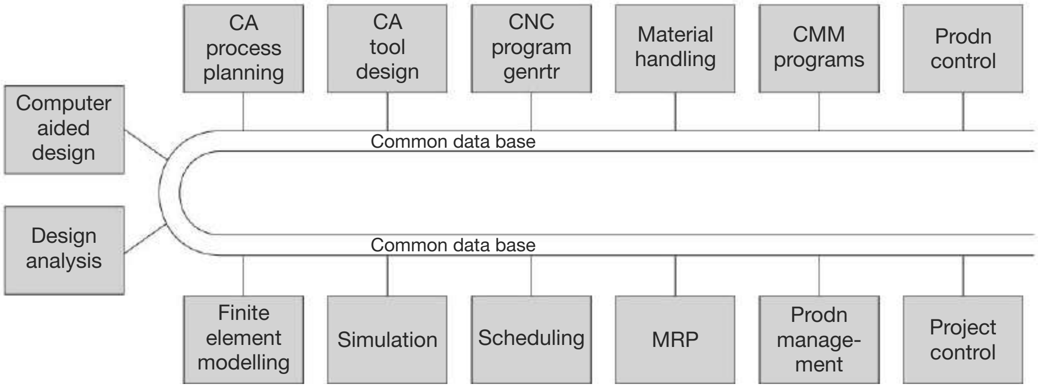


Fig. 1.12 *The common databases as linkage to the various computerised applications*

Today's CAD technology can provide the engineer/designer the necessary help in the following ways:

1. Computer Aided Design (CAD) is faster and more accurate than conventional methods.
2. The various construction facilities available in CAD would make the job of developing the model and associated drafting a very easy task.
3. In contrast with the traditional drawing methods, under CAD it is possible to manipulate various dimensions, attributes and distances of the drawing elements. This quality makes CAD useful for design work.
4. Under CAD, you will never have to repeat the design or drawing of any component. Once a component has been made, it can be copied in all further works within seconds, including any geometric transformation needed.
5. You can accurately calculate the various geometric properties including dimensions of various components interactively in CAD, without actually making their models and profiles.
6. With the constraint-based modelling methods that are prevalent in most of the commercially available CAD systems, it is possible to capture the design intent into the product model beyond the simple geometry. This will help in actually making modifications easily. Also, it is possible to try various options, thereby optimising the whole design process. Thus, the geometric modelling process can be driven by the physics of the process.
7. Modification of a model is very easy and would make the designer's task of improving a given product simple to take care of any future requirements.

8. Use of standard components (part libraries) makes for a very fast model-development work. Also, a large number of components and sub-assemblies may be stored in part libraries to be reproduced and used later.
9. Professional CAD packages provide 3D (3-dimensional) visualisation capabilities so that the designers can see the products being designed from several different orientations. This eliminates the need of making models of products for realisation and explaining the concepts to the team.

Not only this, several designers can work simultaneously on the same product and can gradually build the product in a modular fashion. This certainly provides the answer to the need of today's industry and the one emerging on the horizon.

1.4 COMPUTER AIDED MANUFACTURING (CAM)

We can broadly categorise the industrial manufacturing activity (for only mechanical engineering industries, i.e., making discrete components) into the following:

Mass Production – large lots, e.g., Automobiles In this, the volume of production is very high, ranging from a few thousand to millions per annum. The very high volume justifies the use of special-purpose machines and transfer lines to decrease the cost of production substantially. Also, these ensure that a very high degree of accuracy can be achieved with these systems. However, these manufacturing methods, once designed and fabricated, are very inflexible and can only be used for a single product. Further, the lead time taken from the product design stage to the setting up of the manufacturing facility is very large, varying with the product. Examples in this category are the automobiles, typewriters, etc.

Batch Production – Medium lot sizes, e.g., Industrial Machines, Aircrafts, etc. Batch production refers to the making of jobs in medium lots, say 100 to 1000, for a component type. Thus, transfer lines may not be used in their production, but special-purpose machines which can be easily modified by the use of jigs and fixtures for such jobs can be utilised.

Job-Shop Production-Small lots or one off, e.g., Prototypes, Aircrafts, etc. Job-shop production refers to the manufacture of very small lots, often of single jobs. This may be required in special situations for the purpose of proving a design, making prototypes, in tool making, or for special-purpose applications. In view of the very small lot, no special-purpose machines or tooling can be economically justified. Hence, the manufacture has to be carried on with the general-purpose machines and tooling, which is a very lengthy and often error-prone process.

These are graphically illustrated in Fig. 1.13.

Where does CAM find applicability? Practically, in all the ranges of production. However, its use is more important in categories 2 and 3 by virtue of the added amount of data processing needed in these. In particular, the present trend of large varieties and lower product lives requires that the total manufacturing lead times be smaller. The only way to ensure this is to improve manufacturing methods and CAM proves indispensable in this regard.

1.4.1 Advantages of Using CAM

Greater Design Freedom Any changes that are required in design can be incorporated at any design stage without worrying about any delays, since there would hardly be any in an integrated CAM environment.

Increased Productivity In view of the fact that the total manufacturing activity is completely organised through the computer, it would be possible to increase the productivity of the plant.

Greater Operating Flexibility CAM enhances the flexibility in manufacturing methods and changing of product lines.

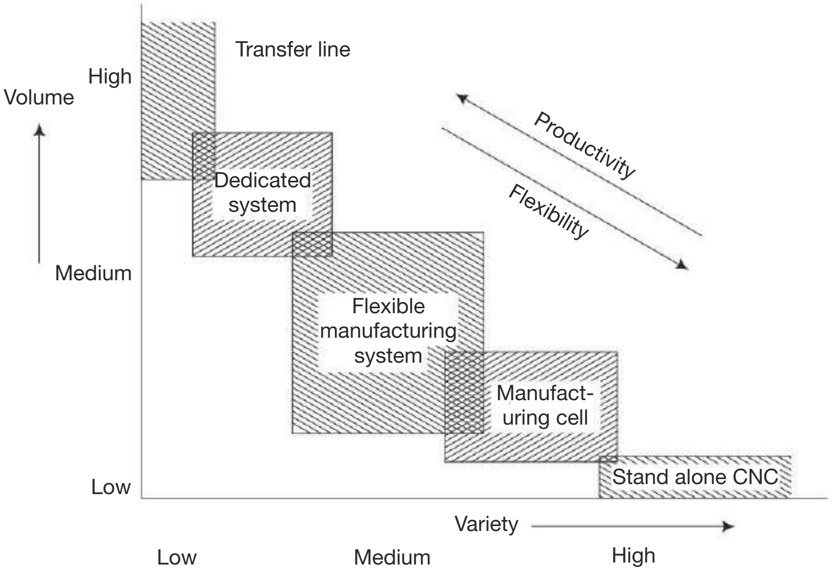


Fig. 1.13 *Manufacturing methods based on production quantity*

Shorter Lead Time Lead times in manufacturing would be greatly reduced.

Improved Reliability In view of the better manufacturing methods and controls at the manufacturing stage, the products thus manufactured as well as the manufacturing system would be highly reliable.

Reduced Maintenance Since most of the components of a CAM system would include integrated diagnostics and monitoring facilities, they would require less maintenance compared to the conventional manufacturing methods.

Reduced Scrap and Rework Because of the CNC machines used in production, and the part programs being made by the stored geometry from the design stage, the scrap level would be reduced to the minimum possible and almost no rework would be necessary.

Better Management Control As discussed above, since all the information and controlling functions are attempted with the help of the computer, a better management control on the manufacturing activity is possible.

All the above advantages when properly translated, would mean a lower total cost and consequently, higher final earnings. Therefore, any manufacturing activity can get the benefits of Computer Aided Manufacturing, be it a job-shop production or mass-scale manufacture. However, better results would be obtained when the design and manufacturing activities are properly integrated. Also, when there is a large variety of products or minor changes required in the existing production programme, CAM can easily manage the necessary alterations.

1.5 COMPUTER INTEGRATED MANUFACTURING (CIM)

As explained above, there are a number of advantages to be gained by employing computer applications in individual domains of the product cycle. It is possible to utilise computers in all aspects of the product cycle as shown in Fig. 1.10. However, the synergy can be obtained by integrating all the functions through the computer such that all the incremental improvements that are possible can be improved manifold. That is what is normally termed as Computer Integrated Manufacturing or CIM. More details of the architectures involved are discussed in Chapter 24.

Lean Manufacturing It is generally seen in spite of the fact that a number of improvements having been achieved through the employment of computer and other automation efforts, there is a large amount of waste involved in the production, which finally increases the unnecessary costs. Japanese manufacturers have recognised this fact, and developed methodologies that lead to the reduction of waste during mass-manufacturing operations. Taiichi Ohno [Womack *et al,* 2003] is credited with the development of methods to reduce the waste at all stages of manufacturing at Toyota, which is now generally called lean manufacturing. In this system, the products are manufactured as required and not for stocking. By following this philosophy throughout the product cycle, it is possible to reduce the amount of storage at each stage and thereby reduce a large amount of the hidden costs in the final product. Some interesting case studies that have employed lean thinking in their entire manufacturing operations are discussed in Chapter 24.

Six Sigma Six Sigma is considered the most important business tool that has transformed many ideas of management in the recent past. Six Sigma was originally started as a defection reduction process, but progressed into a comprehensive statistically based method to reduce variation in electronic manufacturing processes in Motorola Inc in the USA. Now it is practically adopted by a majority of industries as well as all walks of life such as government departments and hospitals. In industries even a new brand name of *Lean Sigma* is coined to combine the concepts of lean manufacturing and Six Sigma. The UK Department for Trade and Industry defines Six Sigma as ‘a data-driven method for achieving near-perfect quality. Six-Sigma analysis can focus on any element of production or service, and has a strong emphasis on statistical analysis in design, manufacturing and customer-oriented activities.’

The DMAIC model for process improvement is used as the basis for Six-Sigma implementations. DMAIC stands for

- Define opportunity
- Measure performance
- Analyse opportunity
- Improve performance
- Control performance

Further details of Six Sigma as employed in the entire manufacturing operations are discussed in Chapter 23.

Summary

- The application of computers in manufacturing has been to direct, monitor and control the processes as well as support the various functions of the operations.
- Computers are being applied in all aspects of manufacturing such as design and drafting, engineering, manufacturing, process planning, tool design, material requirement planning, scheduling, etc.
- Major developments in computer hardware and software are directly driving these applications to a greater extent.
- The conventional product cycle involves a number of interlinked operations. Information flows through all these operations to support them. Many of these support operations will be greatly affected by the application of computers.
- Since information will now be in the electronic form, databases become the backbone of the manufacturing operations.

- The conventional design process involves a large number of activities where a variety of processes need to interact while arriving at the best and economic design.
- Computer aided design deals with all the operations that deal with the development of the product. These could include such operations as design, analysis, testing, manufacturing information generation, etc.
- A number of advantages are gained by the use of computers in the design process. Using CAD reduces a number of unwanted repetitive operations, at the same time improving the accuracy, reducing the developmental time and cost.
- Manufacturing operations are organised broadly into mass production, batch production and job-shop production. Computerised manufacturing is involved in all these types of operations. A number of benefits can be achieved by the use of computer aided manufacturing.
- Computer Integrated Manufacturing tries to link all the operations that are used in manufacturing such that the information is shared between all the operations. This would mean the reduction of waste leading to lean manufacturing.

Questions

1. Explain the influence exerted by computers on the manufacturing scene.
2. Specify the various stages present in a conventional design process.
3. Give a schematic and explain how the application of computers helps in the overall improvement of the design cycle.
4. Explain the importance of engineering analysis in the design cycle.
5. Briefly describe the role of engineering analysis process in the product design cycle.
6. Briefly explain the computerised product cycle in the manufacturing environment.
7. What are the functions that get benefited by the use of computers in design and manufacturing functions?
8. Define CAD. Explain the reasons for adopting CAD in an engineering organisation.
9. Explain with an example various steps in the modern design process.
10. Explain the following steps in the design process: Problem definition, and Engineering Analysis.
11. Write about prototype development as part of the design process.
12. Write down the advantages to be gained by the adoption of CAM.
13. Briefly explain the various categories of manufacturing activities.
14. What do you understand by CIM and lean manufacturing?
15. What are the various processes that should be considered in getting the problem-identification phase of the product design?
16. Briefly explain about the importance of Six Sigma in manufacturing operations.

Part – I

HARDWARE AND SOFTWARE COMPONENTS

The McGraw·Hill Companies

2

CAD/CAM HARDWARE

Objectives

The developments that have fuelled the growth of the CAD/CAM industry are mostly based on the major developments in microelectronics, which have helped in leapfrogging the capabilities. The major component of any CAD/CAM system is the basic hardware that goes with the computational aspect, that is, the computer and the associated peripherals that go with it. After completing the study of this chapter, the reader should be able to get

- The idea of the basic structure of a computing system hardware as used in CAD/CAM systems
- The basic developments that are taking place in microprocessors and their speed as it affects the computing performance
- The different types of memories used in computers
- The varieties of graphic input devices used for controlling the graphic input information
- The types of graphic display devices, their capabilities and applications
- The graphic output devices such as printers and plotters used to obtain the hard copies of the graphic information
- The varieties of secondary storage devices used for storing and archiving the CAD/CAM data
- The CAD/CAM system configurations and the software systems that are used in multi-user environments where groups of designers will be working

This would help in understanding and finalising the details of the specifications that one has to look for while planning for a CAD/CAM system.

2.1 BASIC STRUCTURE

The computing system in operation can be compared to a human being in terms of its operating characteristics. The basic configuration of a typical computing system is shown in Fig. 2.1.

The heart of any computing system is the Central Processing Unit (CPU). It is in the CPU that all the necessary functions of a computer are carried out. The main functions performed in the CPU are arithmetic and logic operations. The CPU communicates with the external world through input/output devices or, in short, I/O devices. These are similar to the sensory organs by which a human being maintains contact with the outside environment. These are also collectively called *peripheral units*. Through an input device, the user would be able to communicate with the CPU, either to give certain data or to control the operation of the CPU. The output device is a means through which the CPU gives the results of the computations.

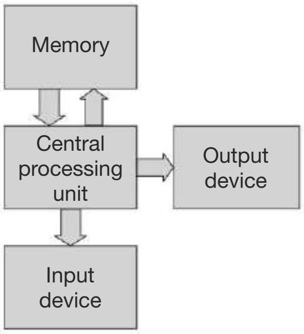


Fig. 2.1 *Organisation of a typical computing system*

Another important unit of a computer system is the memory unit. These are the areas where the necessary data or program (sequence of instructions) is stored. The type of memory and its amount determines the capabilities of a computing system.

The present-day computers work on a principle called the *stored-program concept.* It means that the sequence of operations to be carried out by the CPU is stored in the memory of the computer. The CPU therefore reads an instruction and executes it and continues doing so until it reaches the end of the program. This stored program is called *software* in computer terminology. Since it is the software, which runs any computer, the computer is as good as the program that is running at any given time. More details of software that would be of interest to us will be discussed in Chapter 3.

2.2 CENTRAL PROCESSING UNIT (CPU)

The central processing unit is the nerve centre for any computing system. Based on the software, it organises the information processing for any given application.

The flow of information in the CPU is presented in Fig. 2.2. The program containing the necessary sequence of instructions to the CPU resides in the main memory of the computer, the location of which is specified to the CPU in the form of a program counter. The instructions are in the form of low-level commands, which comprise the repertoire of any given CPU, called *instruction set*. The instruction set consists of instructions for moving the memory contents, performing arithmetic and logic operations and other miscellaneous commands. A CPU fabricated as a single integrated circuit (chip) is termed a microprocessor.

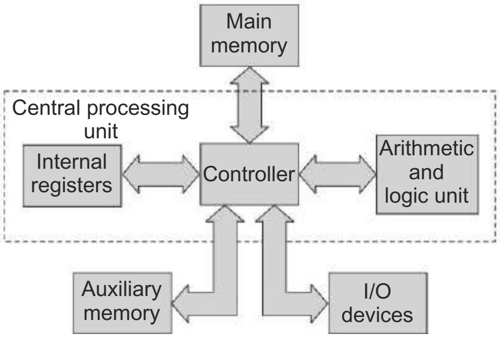


Fig. 2.2 *Organisation of central processing unit*

After being fetched from the main memory, the instructions are to be decoded for the actual action to be taken. This function is performed in the controller, which actually directs the operations to be done through the Arithmetic and Logic Unit (ALU).

The arithmetic and logic unit has the necessary ability to perform arithmetic operations such as addition, subtraction and logic operations as 'AND', 'OR', etc. The necessary values of the operands and their results are stored in certain locations close to the ALU called *registers*. Any operation associated with the registers would be done at the highest possible speed because they can be directly addressed by their names as part of the instruction set. Depending upon the type of CPU, these registers could be 8 bits long, 16 bits long or

32 bits long. Longer register sizes are available in mainframe computers or some special-purpose CPUs. Some of the registers are general purpose in nature so that any type of operation can be performed, whereas others such as stack pointer, accumulator, and address register, etc., carry out specific functions. The concept of the CPU with mainframe computers and minicomputers is essentially that of a Printed Circuit Board (PCB) consisting of a number of devices (chips). However, the era of microprocessors dawned with the increasing acceptance of calculators in the late 60's and early 70's. The first general-purpose microprocessor, Intel 4004, which was accidentally designed for a Japanese calculator company, was released in 1971, heralding the microprocessor era.

Though Intel 4004 was the first microprocessor, since it was designed keeping the calculator in mind, it was very limited in function. A greater use of microprocessors started with the availability of Intel 8008 in 1972. The market response for Intel 8008 was far beyond the expectations of the Intel Corporation. As a result, from then on there were rapid developments in microprocessors by a number of manufacturers, such as Motorola, Rockwell, Zilog, National, Texas Instruments, etc.

The first microprocessor, Intel 4004, was essentially a 4-bit microprocessor, whereas Intel 8008 was an 8-bit microprocessor and the subsequent development of 16- and 32-bit microprocessors took place in later years. What do '4' bit or '32' bit mean? It refers to the length of the word as used by the microprocessor, that is, the maximum length (number of bits) in which the information can be processed inside the CPU. The greater the length of the word, the faster would be the execution of a given program by the microprocessor. However, the complexity of the micro-circuitry that goes with the making of the microprocessor grows as the word length increases. Currently (in 2009), 32-bit and 64-bit microprocessors are generally in use for computing applications with all the others becoming obsolete. However, microprocessors with smaller bit sizes are used for embedded control applications.

There are a few things to be noted with microprocessors, such as address and data. Since the data is stored in memory locations, which are in large, it is necessary to know the address of the memory location for operating on the contents of that memory location. Hence, when the CPU tries to fetch data from any given location, it needs to know the correct location address. The width of the *address* specifies the maximum amount of memory locations that a microprocessor can directly address. For example, a 16-bit address would mean a linear address space of 2^{16} locations which is 65 536. This is represented as 64 k, where k stands for 1024 (2^{10}).

Data is the name given to any information that is written or retrieved by the microprocessor from the memory. Data can be a part of the program, constant parameter values required for the execution of the program, or results of computation.

Normally, the memory is not organised as individual bits, since it would increase enormously the problem of accessing data from it. Therefore, it is organised in multiples of bits. For example, many of the earlier systems were organised as 8 bits, known as a *byte*. The length of a data word normally depends on the length of the word used by the CPU, but there are exceptions to it. For example, though Intel 8088 is a 16-bit microprocessor, its memory is organised in 8-bit arrays. This was resorted for taking advantage of the relatively low-cost 8-bit peripheral devices that were available at that time. So for fetching a word of data, one has to perform two cycles of fetching of 8 bits each. Hence, memory operations become slower in such a situation.

Another important aspect to be considered with microprocessors is the speed at which they operate, termed the *system clock*. The faster the system clock, the faster would be the execution of instructions. Current microprocessors have clock rates upwards of 3 GHz. The current trend is not only to increase the clock speed but also to increase the number of cores.

The speed of the CPU operation is traditionally measured in terms of MIPS (Million Instructions Per Second). This is not a true indication of the overall performance of the system. Hence, efforts have been made to come up with a more objective evaluation of performance, taking into account not only the processor performance, but also other associated peripherals as well as the software available. A variety of benchmarks have been defined by SPEC (Standard Performance Evaluation Corporation) to take both the integer and floating-point performance of the computing system. These include the processor, memory hierarchy and compiler capability only and not the associated peripherals. In addition to these, there are a number of specific benchmarks developed by various other agencies based on specific applications in mind.

In CAD/CAM applications, the computational load being very high, a single CPU would not be able to take it up, thereby slowing down the whole system. This problem is overcome by the use of coprocessors, which have specialised capabilities. The main CPU would offload a particular job to that coprocessor which is best suited for it. Since CAD/CAM requires a large number of graphic operations, special Graphics Processors (GPU) are utilised. These relieve the main processor of most of the graphic computations that are required, such as graphic transformations. However, with the ultra large-scale integration in the microprocessors, the coprocessors are embedded in the current generation microprocessor itself.

The requirements of a typical CAD/CAM computer should be a microprocessor of at least 32 bits, but preferably 64 bits, with a clock rate above 1.0 GHz, an address of at least 24 bits but preferably 32 bits, and the support of the coprocessors described above. The MIPS rating may be above 250.

The Intel Pentium processor family is the first to provide the performance needed for mainstream workstations for CAD/CAM applications. The Pentium processor super scalar architecture can execute two instructions per clock cycle. Branch prediction and separate caches also increase performance. Separate code and data caches reduce cache conflicts while the remaining software is transparent. The Pentium processor has 3.3 million transistors and is built on Intel’s advanced 3.3V BiCMOS silicon technology.

The Intel Pentium 4 processor released in the year 2000, uses the Intel’s 0.13 micron process and the processor core contains approximately 42 million transistors. The maximum Pentium 4 processor speed available currently is at 3.2 GHz (mid 2009). This has been superseded by the multi-core processors. The dual-core Pentium processors in 2009 utilise the 45-nm (0.045 μm) process and as such have a much larger number of features present in the microprocessor, thereby ensuring very high performance.

As the processing requirements are increasing, hardware developers have problems in increasing the clock speed due to the associated problem of heat generated. An alternative way is to have two CPUs instead of one to boost the processing power, but that complicates the design of the motherboard thereby increasing its cost. Hence, they have come up with the concept of multiple cores in a single microprocessor, thereby increasing the processing power and at the same time keeping the motherboard economical.

By definition, a multi-core processor is an Integrated Circuit (IC) that has two or more processors to enhance the performance, reduced power consumption and more efficient simultaneous processing of multiple tasks. For example, a dual-core processor is comparable to having two separate processors installed in the same computer. However, since the two processors are part of a single IC, the connection between them is faster. Ideally, a dual-core processor is nearly twice as powerful as a single-core processor. In practice, performance gains are said to be about fifty per cent: a dual-core processor is likely to be about one-and-a-half times as powerful as a single-core processor. Currently (2009), quad-core processors are available in workstations, while 8-core processors are available for testing. All these processors are being manufactured using Intel’s 45-nm technology thereby allowing a large number of features embedded in the chip. Intel’s 32-nm technology is production-ready in 2009 and will be used for future processors coming from their stable. One thing to be noted with multi-core processors is that the software should be written for parallel processing to take full advantage of the capability.

The operating system (such as Windows XP) has a number of tasks to be done that will be scheduled. Since there are more CPUs, it is easier for the operating system to schedule these tasks, thereby improving the performance. Intel and AMD have their dual and quad-core processors available currently for use in the CAD/CAM workstations.

The Intel Pentium family, which is most widely used in PCs and other low-cost workstations, is normally called CISC (Complex Instruction Set Computing) processor. A CISC processor relies on improvements by adding more instructions to its repertoire. Thus, the system programmers will be provided a new set of instructions with each advancement to simplify some of the tasks which they are doing in the software. But in the process, the processor micro circuitry becomes more complex and requires more careful manufacturing processes.

As opposed to this is the RISC (Reduced Instruction Set Computing), which has a small range of instructions that are optimised for the given application. In view of the smaller number of instructions present, it is possible that the chip design is more efficient and hence is generally fast and more powerful. However, the system software developers will have to write more code. Typical examples of RISC processors are

- IBM POWERPC 750CXe running at 700 MHz (using 0.18 μm technology with copper-wiring technology)
- The 64-bit MIPS R12000 processor running at 375 MHz for Silicon Graphics
- Hewlett Packard PA 8200 Processor running at 200 MHz and 236 MHz
- Sun UltraSPARC-III Processor running at 750 and 900 MHz (with 29 million transistors)

With the advancement in multi-core processors, the interest in RISC processors has almost diminished, and not much development is taking place.

2.3 MEMORY TYPES

Memory is an integral part of a computer, storing data and programs, often called *main memory*. Traditionally, this used to be called *core memory* in the mainframe computer parlance. The name 'core memory' was given because the individual memory cells were formed by making a small magnetic core about 1.5 mm in size wound around a thin material of wires. Whether these cells would be magnetised or not depended upon the data to be written. By magnetising, they obtained a specific orientation, which would not be erased even when the electric supply was put off. However, this method was very expensive and is now hardly used due to the availability of semiconductor memory.

In semiconductor memory, the memory locations are organised as a series of small on/off switches (transistors). Developments in the semiconductor industry have decreased the cost of the memory units, apart from raising their capacity for storage. Presently, a single chip capable of storing 256 M bits of information is being used by a number of manufacturers. IBM and others have demonstrated the 1 G Bit units also, which will be available soon. Increase in the capability of holding a number of memory units in a single chip greatly decreases the complexity of the circuitry involved in the computer motherboard.

The various types of semiconductor memory units available are

• ROM	Read only memory
• PROM	Programmable ROM
• RAM	Random access memory in its various forms
• EPROM	Erasable programmable ROM
• EEPROM	Electrically erasable and programmable ROM
• Flash memory	

Read Only Memory (ROM), as the name implies, can only be read but cannot be written upon. Thus, the ROM when prepared can hold information, which need not be updated during the useful life of that information. Most of the system software is normally provided in the form of ROM. However, to be cost-effective, ROMs are to be manufactured in fairly large numbers, in millions.

For applications, which do not justify a large volume, PROM is used. Here, a general-purpose ROM from the semiconductor factory is specifically programmed by suitable means and then used as ROM. The PROM can be programmed only once. A ROM, which can be erased fully, by means of ultraviolet light and then reprogrammed using a high voltage (32 V) signal, is called EPROM. These can be programmed in part or in full a number of times, but the erasure when done would necessarily be full. EEPROM is an alternative to the EPROM in which information can be written and erased using the normal voltage signal (5 V) available for operating all the circuits.

Random Access Memory for RAM is not apt, since all semiconductor memories can be accessed randomly. It is essentially a read and write memory. The information can be read as well as written into the memory. However, the information stored can only be retained till the power supply stays on. Since most of the memory of a computing system is contained in the RAM, it is necessary to maintain the power supply throughout without even a momentary interruption, otherwise the data stored or computations carried out partially would all be lost.

Some RAM chips are coming up with a lithium battery embedded permanently in the casing of the RAM chip so that information present in the RAM is retained even when the external power supply is cut off. RAM chips come in two varieties, static and dynamic. In static RAM, the information is to be written only once whereas in dynamic RAM, the information after having been written would have to be continuously refreshed for merely being retained, even though there may be no change in it. Static RAM is more expensive compared to the dynamic RAM or DRAM.

There are currently three main types of DRAM technology used in computers: Fast Page Mode (FPM), Extended Data Out (EDO) and Synchronous DRAM (SDRAM). Also, the RAM can be fixed as parity, non-parity, Error-Checking-and-Correcting (ECC) or ECC-on-SIMM (EOS).

FPM DRAM Fast-Page Mode memory is a type of DRAM that allows for replicated memory access with minimum waiting for the next instruction.

EDO RAM EDO (Extended Data Out) It is a memory technology that can provide approximately a 5–30% boost in the memory subsystem speed versus Fast-Page Mode. Also known as Hyper-page mode DRAM, EDO provides increased performance by outputting data at the same time it is searching for new information. Fast-page memory has to wait between such operations, thus causing delays. EDO reduces the bottlenecking in data transfers between high-speed processors that need to get data quickly. Of critical importance, a high-performance computer system must be designed to take advantage of EDO memory to realise the benefits. The increase in system-level performance gained by EDO DRAMs is quite good considering that the performance benefit carries no increased cost.

SDRAM SDRAM, or Synchronous DRAM, is a fast, high-bandwidth memory designed to work best with systems employing high-performance PC chipsets and processors. This technology synchronises itself with the system clock that controls the CPU, eliminating time delays and improving processor efficiency. Offering bandwidths of up to 100 MHz—twice the bandwidth of EDO—SDRAM is a result of a major shift from EDO and Fast Page Mode DRAMs. As bus speeds increase, the performance difference increases. To use SDRAM, a computer system must be designed to support that kind of memory, and as most new chipsets support SDRAM technology, it has become the new RAM standard.

RDRAM Rambus Direct RAM is another technology, which is slowly gaining ground in the high-end workstations. It utilises a narrow, uniform-impedance transmission line, the Rambus Channel, to connect the memory controller to a set of RIMMs. This single channel is capable of supplying 1.6 GB/s of bandwidth. However, it is possible to use multiple channels, thereby increasing the bandwidth. This suits well with the new processors, most of which operate at or above 1 GHz. However, the RDRAM is still very expensive compared to the SDRAM, partly because it is a new technology. The usage of RDRAM is almost negligible in workstations.

DDR-SDRAM DDR-SDRAM (Double Data Rate) is another variation of SDRAM that is faster compared to normal SDRAM and is gaining ground and replacing the SDRAM in most workstations. The basic principle of DDR-SDRAM is very simple. It is able to transport double the amount of data by using the rising as well as falling edge of the clock signal for data transfers. The speed is further increased by quadrupling the transfer rate to 4 times using the quad-pumped technology known as DDR2. DDR-SDRAM has another important improvement over PC133 SDRAM. Its voltage supply uses only 2.5 V, instead of 3.3 V. This and the lower capacities inside the memory chips lead to a significantly reduced power consumption.

The next-generation SDRAM is DDR3 which doubles the data-transfer rate of DDR2 while reducing the power consumption. These also can be used at high speeds of up to 1600 MHz, and thereby improve the performance manifold. However, they are still used in lower volumes because of their high cost.

Parity and Non-parity Parity checking circuitry adds an extra bit to each 8 bits of memory data. The benefit of incorporating parity memory in a system is the ability to detect single-bit errors and send an error message before halting the system. After reading 8 bits of data, the memory controller can examine the parity bit to detect any single-bit errors. If an error occurs, the system notifies the user and usually shuts down immediately. With many business users dependent upon the accuracy of the data being processed, requiring parity memory is an important consideration. Non-parity memory is less expensive than parity memory. Parity checking is also limited to detecting only odd numbers of bit errors. Even though they are extremely rare, even numbers of bit errors are possible.

Error Checking and Correcting (ECC) ECC memory provides a vast improvement in memory integrity management over parity checking. ECC circuitry automatically detects and corrects single-bit errors and can detect highly unlikely multiple-bit errors. As with parity checking, ECC requires one extra memory bit for each byte of data. ECC memory requires more overhead than parity memory for storing data and causes approximately a 3% performance degradation in the memory subsystem and is also expensive. However, the resulting error detection and correction can make the trade-off well worth it. Differences in system failures when comparing the parity to ECC memory are shown in Table 2.1. Many new high-end computers include ECC circuitry in the memory controller. This allows the use of low-cost parity memory modules to provide advanced ECC functionality.

Table 2.1 *Comparison of memories with and without ECC*

Amount of Memory	*System Failures with Parity (in 5 years)*	*System Failures with ECC (in 5 years)*
16MB	0.32	0.00016
64MB	1.26	0.00064
128MB	2.52	0.00128

ECC-on-SIMM EOS or ECC-on-SIMM (Single In-line Memory Module) is a special type of ECC memory that performs error checking and correcting within the memory SIMM. Performance is not affected (over standard ECC) but provides ECC function to systems without an ECC memory controller.

Flash Memory It is a type of non-volatile memory that can be erased and reprogrammed in units of memory called blocks and not single bytes. Otherwise, the writing process is similar to EEPROM. It is also called 'flash RAM'. With the cost of production of flash memory becoming small, it is extensively used presently in the form of a separate drive that can be attached to the USB port, generally called *jump drive* or *thumb drive*. It can be used for transferring data similar to a floppy drive, but with much larger capacity and speed. Currently, these drives are available from 32 MB to 64 GB within reasonable cost.

2.4 INPUT DEVICES

These are the devices through which the user/ operator communicates with the computer for feeding it with the necessary information, both graphical and alphanumerical as required. The various devices used are

- Keyboard
- Mouse
- Light pen
- Joystick
- Digitiser
- Tablet
- Scanner

Keyboard The keyboard is the most basic input medium for all computers. The layout of keys on a keyboard generally consists of the traditional typewriter keys together with some special keys, which are used for controlling the execution of the program or the screen display (cursor movement). The presence of a higher number of keys would facilitate the interaction. But for CAD/CAM applications, a keyboard itself may not be sufficient by virtue of its being a digital interface. Hence, in addition to the keyboard, other analog input devices are also used for CAD/CAM applications. These devices may be used for entering graphic data in a convenient form or for selecting an item from the menu displayed on the screen.

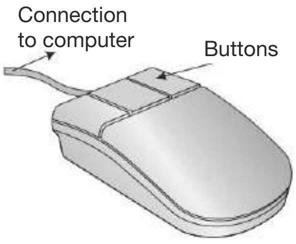


Fig. 2.3 *Mouse*

Mouse The mouse is a pointing device, which has been gaining importance with the advent of the microprocessors, and the pull-down menus associated with the application software. The mouse operates on three basic principles—mechanical, optical and opto-mechanical. The *mechanical mouse* contains a free floating ball with rubber coating on the underside (Fig. 2.3) which when moved on a firm plane surface would be able to follow the movement of the hand. The motion of the ball is resolved into X- and Y- motions by means of the two rollers pressed against the ball. They in turn control the cursor on the screen, which can then be utilised for any desired applications by means of the clicking of the buttons on the mouse. This can only suffice to point on the screen but not for giving positional data. Further, the mouse is a relative device and not an absolute pointing device as the digitisers to be discussed later. Many a mouse available in the market contains two buttons for its operation, though mice with a larger number of buttons (3 and 4) are also available but have gained only limited acceptance.

In the case of the *optical mouse*, a special reflective plane surface with etched fine grids is required. The LEDs present inside the mouse (in place of the rubber ball) would reflect the number of grid lines crossed in the X and Y directions, thereby showing the distance moved. The life of the optical mouse is high since it has no moving parts, but it has not gained as much acceptance as the mechanical mouse because of the special surface needed for its operation. The operation of the *opto-mechanical mouse* is similar to that of the mechanical mouse, but the position resolvers used are based on the optical principle.

Laser Mouse Laser tracking responds to the hand movement with a lot more accuracy than with LED. Also, it can track any surface, such as high-gloss surfaces that LED-based mice simply can't negotiate. Laser illumination being pointed and coherent reveals the structure of a surface that an LED cannot recognise. The coherent nature of laser light creates patterns of high contrast when its light is reflected from a surface. As a result, a laser mouse has no difficulty in tracking even plain surfaces. In future, practically all mice may be converted to use laser.

Light Pen A light pen resembles a fountain pen in the method of holding, but it works on the principle of light rather than ink, hence the name. Light pens are not used for writing on the screen as is erroneously believed by many, but actually only to detect the presence of light on the screen, as shown in Fig. 2.4, with the help of a light-detecting resistor. Their normal use in graphic applications is to identify objects or locations on the display screen for possible graphics handling. These are to be used only with refresh-type display devices. The resolution of the light pen is poor, as the field of view of the photo-sensitive element is conical.

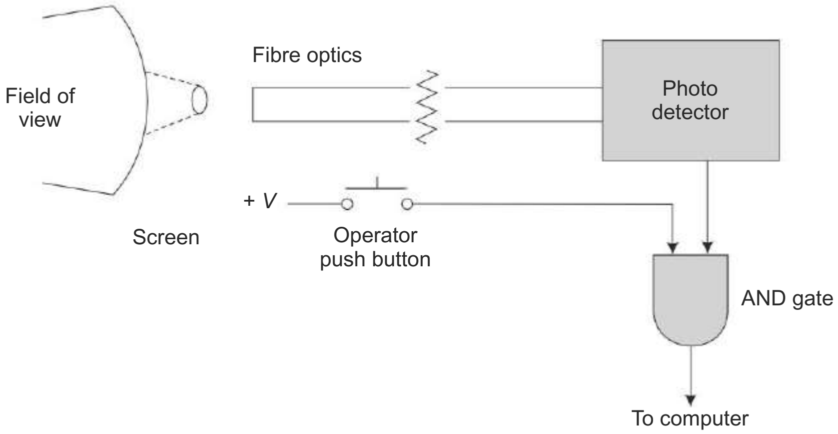


Fig. 2.4 *Light pen*

Since the light pen points to the graphic display directly, it is a natural graphic interactive tool. However, as the operator has to hold the light pen against the gravity along with its cable connecting the graphic adapter card for making any selection, ergonomically it is inconvenient to use it over long periods.

Joystick A joystick can also be used to control the on-screen cursor movement as a mouse does. A joystick can indicate the direction, speed and duration of the cursor motion, by the movement of the stick, which contains a ball seated in a spherical cavity, held in position by the operator. Generally, the response of the joystick would be quicker compared to other cursor-tracking devices. Hence, they are more suited for video games.

Digitiser A digitiser is the most widely used input medium by the CAD designer. It is used for converting the physical locations into coordinate values so that accurate transfer of data can be achieved. Very high resolution in the order of 0.1 mm or better can be achieved. A *tablet* is essentially a low-resolution digitiser having a small work area. The work area corresponds to the full CRT screen.

The designer can work with any pointing device similar to a pen, and do normal writing on the tablet as if he were doing so on a drawing board. The movement of the pen tip would be communicated onto the screen, which the designer can modify depending on the software at his disposal. Since it gives natural feel to a designer for free form sketching (which can be straightened if necessary by the software), it is generally preferred as a pointing device in CAD applications. Another kind of pointing device used in tablets is a *puck*, which has a cross-hair line cursor and a number of buttons, as shown in Fig. 2.5, and is used normally with other digitisers. This is useful for the menu selection. The buttons can be assigned for different auxiliary functions.

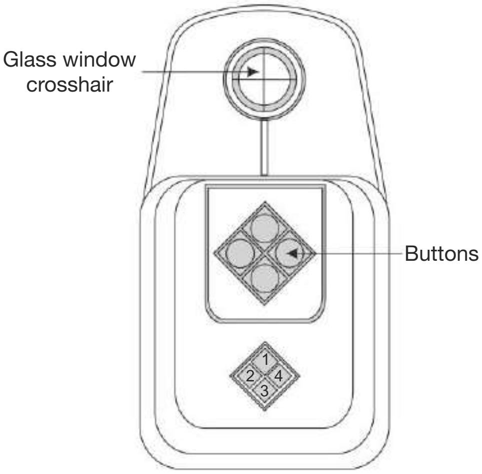


Fig. 2.5 *Puck or pointing device used with tablets and digitisers*

A digitiser consists of a rectangular smooth surface as a draughting board, as shown in Fig. 2.6. Underneath this surface is a position-sensing mechanism. The designer interacts through the handheld locator (or puck as shown in Fig. 2.5) which contains a number of buttons. The designer can move the puck to the desired position and then by pressing one of the buttons to initiate a certain action. A digitiser is an absolute measuring device.

Electromagnetic force is the measuring means that is most generally employed in digitiser construction. It contains a printed circuit board, which is etched with evenly spaced traces on both sides, one representing the *X*-axis and the other, the *Y*-axis. The locating device used (such as puck or a pen type) would contain a coil which can act as a receiver or a transmitter. In the transmitter mode, electronic signals on the board can be measured for their strength, and the highest strength signifies the location of the desired point. The digitiser controller would measure the position of *X*- and *Y*-axes alternatively.

Fig. 2.6 *Digitiser*

The other technology used in the digitiser is resistance. The digitiser is made of two sheets separated by a number of spacer dots. One sheet is evenly deposited with a resistive film, whereas the other is coated with a conductive film. A small voltage is applied across the resistive film. When pressure is applied by the puck or stylus on the digitiser pad, the potential of the conductive sheet would be proportional to the distance from the end, which gives the locational data. The measurement is again to be done alternatively in *X*- and *Y*-axes. The same resistive technology may be effectively used in touchpad-type applications. In a touch screen, a transparent digitiser composed of conductive glass film is present on top of the CRT screen, such that when pressure is applied at a given point on the screen, it can be sensed. This is comparable with the application of a light pen.

The digitiser comes in a large number of sizes, from 250 × 250 mm to as high as 1000 × 3000 mm. The quality of a digitiser can be measured in terms of resolution, accuracy, linearity and repeatability. Linearity is the variability of accuracy over large areas of the digitiser. A digitiser communicates with the computer in single-point mode, as the coordinates of the point are transmitted when a button on the puck is pressed. This is used for digitising discrete points such as an already existing drawing. However, when continuous sets of points are required, the puck can be moved continuously over that line, and the coordinates of the points are sent to the computer at a specified sampling rate, such as 200 coordinate points per second. Some typical digitiser specifications are presented in Table 2.2.

Table 2.2 *Digitiser Specifications*

	GTCO DrawingBoard VI- DB6-2024	*CalComp CADPro*
1. Digitising method	Electro-magnetic	Electro-magnetic
2. Digitising area, mm × mm	508 × 610	152 × 229
3. Resolution, mm	0.002	0.006
4. Accuracy, mm	0.05	0.20
5. Interface	USB, RS 232	USB, RS 232
6. Maximum transmission speed for serial communication, baud rate	38 400	38 400
7. Operating modes	Point, line, run, track, increment, prompt	Run, increment, prompt

Another class of input devices sometimes used is the 3D digitiser, which has the ability of converting any 3D object into its dimensional form. These are also sometimes referred to as *space digitisers*. The method of digitising could be to manually move a stylus along the desired 3D object, but this is time-consuming. Therefore, optical scanning is done for such imaging work. A thin plane of light is projected onto the 3D image which, when reflected, is received by a camera through a system of mirrors and is then recorded.

Scanner A scanner digitally scans images or text present on a paper optically and converts it into a digital image as a bit map. The scanner consists of a CCD (Charge Coupled Device) array which takes the image in the form of dots of very high resolution (300 to 4800 dpi). A lamp illuminates the document. The light reflected from the document reaches the CCD array through a series of mirrors, filters and lenses. Depending upon the application, a 2D image is normally converted into its raster format of the requisite resolution. This is the medium that is normally used when old information is in the form of hard copy and can be converted into electronic form. The type of conversion depends upon the type of document in question, for example, when character- or image-based information can be simply converted using OCR (Optical Character Recognition) software for converting the input into a word processing program. If the document is an old drawing then the raster image can be converted into vector format by software tools for the purpose. Then engineers can spend a little time to clean the converted drawing and store it in an appropriate location for future reference.

A scanner typically consists of a scan head which moves over the scanned object and sends a bitmapped representation of the object to the scanning system. The processing software present in the scanner system converts and sends it to the application where it is further processed. The scan head has the ability to discriminate between levels of lightness and darkness, shading, and colours.

The most common type of scanner used is the *flat-bed scanner*. This is capable of scanning a variety of objects that can be accommodated on its glass platform. In appearance, it looks like a photocopy machine and has a glass platform where the objects to be scanned are placed face down. Alternatively, in a sheet-fed scanner individual pieces of paper can be fed at a high rate. Another form of a scanner that is sometimes used is the handheld scanner. It is much smaller and inexpensive, but has the ability to scan only a small area.

Another type which is quite frequently seen in industry is a 3D scanner which is used for digitising three-dimensional objects. There are many technologies used for digitising the 3D objects. The one that is commonly used in industry is the *non-contact laser scanner*. The scanner sends light in the form of a laser and receives a part of the reflected light and measures the coordinates on the surface of the object. These scanners work on the principle of stereo vision, in that the same point is measured simultaneously by two probes which provide the third dimension. They are much more expensive compared to the flat-bed scanners.

2.5 DISPLAY DEVICES

The display device forms the most important element in a CAD/CAM system, since on this most of the design work and simulation of manufacturing can be graphically displayed. The display media that are used are

- Cathode-Ray Tube (CRT) display
- Plasma panel display
- Liquid Crystal Display (LCD)

Of these three methods, it is the CRT displays that are the most advanced and extensive in use, in spite of their bulkier size.

2.5.1 CRT Display

In a CRT display, the heated cathode emits electrons, which are formed into a stream, accelerated and focussed onto a point on the display screen. The display screen contains a phosphor-coated surface as shown schematically in Fig. 2.7, which gets illuminated when the speeding electrons hit the surface, displaying the point. The electron beam is controlled by means of deflection plates for accessing any point on the surface of the screen. Changing the beam current changes the intensity of the spot created on the screen.

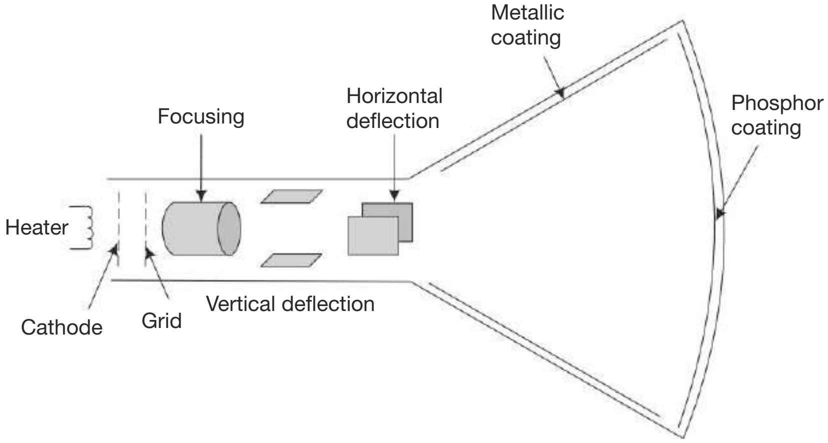


Fig. 2.7 *The cathode-ray tube principle*

There are basically two types of image-drawing techniques that are used in graphic displays. They are

- Stroke-writing
- Raster scan

In a *stroke-writing display*, the electron gun directly draws the vectors on the screen to generate the image, whereas in the *raster scan*, the whole display surface is divided into a matrix of small dots called *pixels* (picture elements) and the electron beam scans the whole surface area line by line, as in that of a home television.

Irrespective of the writing technology employed, the phosphor glow created by the electron impingement on the picture screen is short-lived. Therefore, some means are to be devised for overcoming this shortcoming and achieving a static image on the screen.

One method of maintaining a static display is by using storage tube technology developed by The Tektronix in 1972 called the Direct View Storage Tube (DVST). Here, the display is generated by the impingement of electrons as in the conventional CRT. However, a cathode grid would be part of the screen surface, which, once excited by the electron beam would continuously emit electrons which would maintain the image on the screen. This is desirable because there being no need for refreshing the image, substantial overheads on the display electronics are averted. The resolution that can be achieved is in the order of 3000 × 4000 addressable points on a 19-inch tube and it results in a clear image without any flicker. This explains the popularity of this technology with the earlier CAD/CAM system.

However, the disadvantage of this system is that once written, partial erasing of an image is not possible. Any necessary modifications could only be made after completely erasing the picture and then redrawing, which takes a lot of time, particularly for complex images if they were to be altered a number of times. Animation is not possible, as it relies on the erasing and drawing of parts of the image on the screen. Also, the image can be obtained in monochrome only. Presently, this type of display device is almost obsolete as far as the CAD/CAM sphere is concerned, though some TEKTRONIX terminals have achieved both colour-filling as well as partial erasure facilities in the storage tubes.

The second method of technology used is that of direct stroke writing with a direct refreshing tube or vector refresh tube. In this, the image is generated on the screen by direct drawing of straight vectors on the screen. As the phosphor glow is short-lived, it is continuously refreshed by repeated stroke writing at a rate fast enough to eliminate flicker from the screen. To maintain a flicker-free vision of the image, it is necessary to refresh the whole screen at a rate of about 60 times every second (60 Hz). Since the image is continuously refreshed, it is possible to erase and modify parts of the display to any extent desired. A major disadvantage of the vector refresh devices is that the display starts to flicker if a large amount of data is on the screen. When the image contains more than 4000 vector inches of graphics, the refresh rate may drop below 30 Hz and the image would start flickering. These are also very expensive.

It is also possible to obtain colour display in refresher tubes. Here, the phosphor coating on the screen contains three different dots (red, green and blue) arranged side by side at the same spot. The CRT contains three electron guns, each corresponding to the primary colour. Information regarding the intensity of each of the colours would control its gun, the beams of all three guns being simultaneously focussed onto the screen through a shadow mask from where they diverge to fall on the corresponding phosphor dots giving rise to the desired colour and intensity on the screen. The glow of each of the phosphors at a given point controls the colour at that point. With only a red glow, a red dot will be displayed, but red and green in combination may present orange or yellow depending upon the strength of each colour, which can be controlled by the beam strength. With proper control of the display electronics, it should be possible to display a large number of colours on the screen. Typically, the number of colours displayed is 4, 16, 64, 256 to 4096 or larger depending upon the capability of the display device.

In the raster scan displays, the complete screen is divided into a matrix of pixels from a typical (320 × 200) to as high as (1280 × 1024) or more as shown schematically in Fig. 2.8. Each square in Fig. 2.8 represents one pixel. The electron beam generates a single dot at the centre of this square. The distance between the squares is called *dot pitch* and it indicates the fineness of the display. A typical dot pitch used is 0.28 mm in the current-day low-cost display monitors. However, a dot pitch less than 0.25 mm is preferable for a sharper display image.

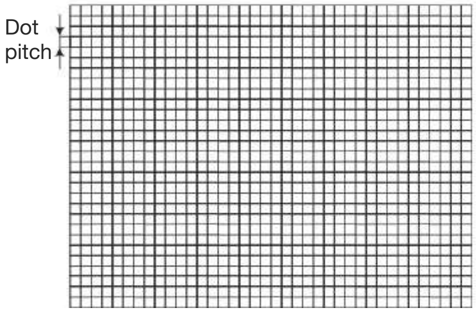


Fig. 2.8 *Dividing the screen into small points called pixels. Each square represents a pixel.*

The display is generated by identifying which pixels need to be bright for a given vector and then the full-screen display is obtained by scanning the complete screen horizontally line by line as shown in Fig. 2.9. This is similar to the process in the domestic television. The refresh rate is to be maintained sufficiently high so that no flicker in the image is perceivable. This normally amounts to 60 times a second and is represented as the refresh rate of 60 Hz. This means that the whole screen is to be completely written in 1/60th of a second. This is called *sequential* or *non-interlaced refreshing*. The refresh rate also depends on the resolution of the screen, with higher resolutions requiring faster refresh rates. Typical refresh rates used in Pentium-based workstations is given Table 2.3.

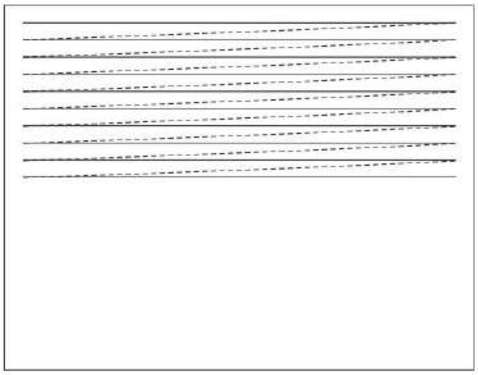


Fig. 2.9 *Raster-scan display for continuous scanning of lines*

Table 2.3 *Typical resolutions used in Pentium-based workstations*

Specification	*Resolution in pixels*	*Minimum Vertical refresh rate, Hz*	*Minimum Horizontal scan rate, kHz*
VGA	640 × 480	85	43
SVGA	800 × 600	85	54
XGA	1024 × 768	85	69
SXGA	1280 × 1024	75	80
UXGA	1600 × 1 200	72	89

The process of writing on a refresh-type monitor requires that the electron beam will pass through all those points which will require it to be bright and as a result, the vectors in the drawings are to be converted into its equivalent pixel points. This process is termed as *rasterisation*. The raster images of lines and circles for typical orientations are shown in Fig. 2.10 and Fig. 2.11.

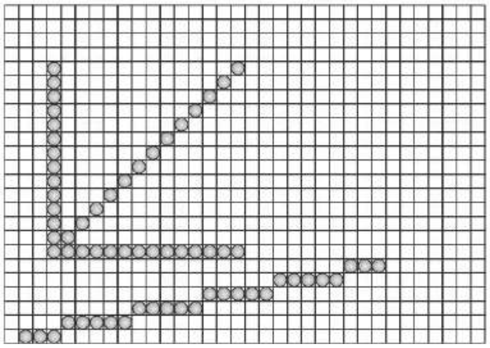


Fig. 2.10 *Raster display of lines*

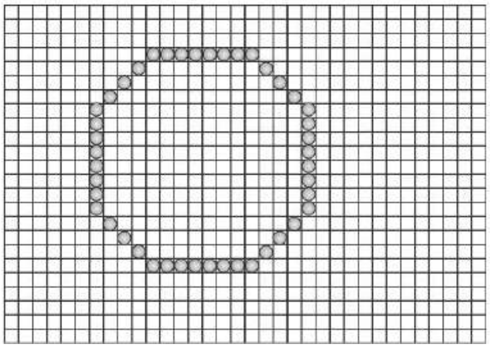


Fig. 2.11 *Raster display of circles*

In some low-cost display devices, decreasing the refresh rate to half at 30 Hz reduces the cost of the monitor. This gives rise to a flickering of the image as in each of the cycles only half of the screen image is refreshed, instead of the full one by omitting alternate lines. This is termed *interlaced refreshing*. In this mode, in the first cycle, alternate lines are refreshed as shown in Fig. 2.12 whereas in the second cycle, the other lines are refreshed. This reduces the overheads on the display control and consequently the costs, but is not suitable for dynamic displays where the display changes fast.

A typical configuration of the raster scan display is shown in Fig. 2.13. Here, the frame buffer contains the complete dot-by-dot image of the display, which is required. From the frame buffer, this information is accessed by the sweeper, which in turn controls the display device. In the earlier systems, a part of the main computer stored the frame buffer, which imposed severe overheads on the performance of the system. As a result, the frame buffer was arranged separately along with a host processor to read from and write onto the frame buffer. The main processor controlled the graphic processor in turn. If there were no separate graphic processors then the main processor would directly communicate with the frame buffer.

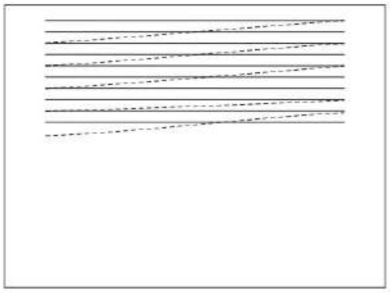


Fig. 2.12 *Interlaced scanning in refresher-type CRT*

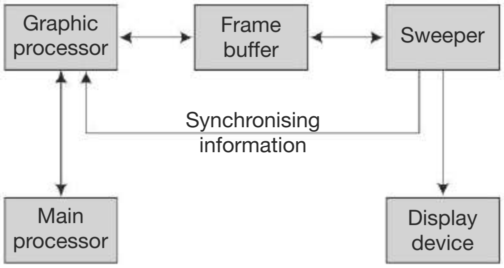


Fig. 2.13 *Information organisation in a raster-scan display*

What should be the capacity of the frame buffer? It depends upon the screen resolution required. Consider a monochrome screen resolution of 1024 × 1024. For each of the pixels to be represented on the screen, one bit of information is needed, so the frame buffer capacity should be 1 M bits or 128 kilobytes. However, for colour display, the required memory gets increased.

The colour information is contained in the form of the intensity of each of the primary colours (red, blue and green) for each colour to be generated. Then the number of bits required per pixel to store all possible colours increases enormously. Normally, the colour display adapter cards contain a look-up table or tables which sequentially codify the colours in terms of a number in each table which contains the necessary colour-mixing information as shown in Fig. 2.14. Thus, whenever a particular colour is to be displayed, the only information to be stored in the frame buffer for each pixel is the colour number and the look-up table. With 2 bits for each pixel, we can generate 4 colours, and with 8 bits per pixel, one can generate 256 simultaneous colours. Typical sizes of colour depth used in some of the available graphic

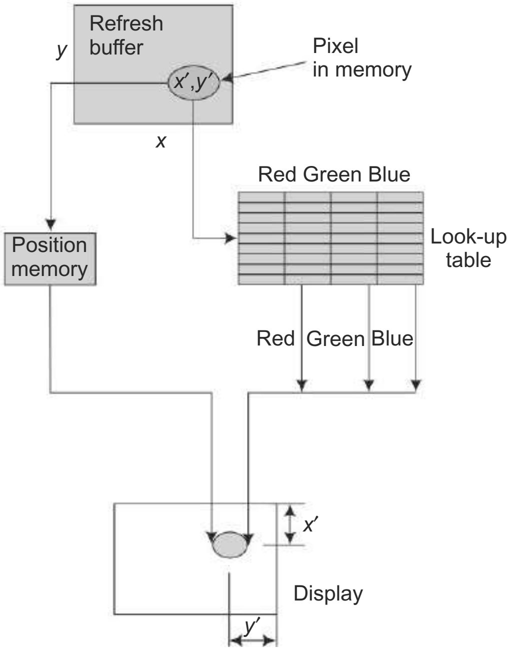


Fig. 2.14 *Colour look-up tables in a raster-scan display*

cards for PC-based workstations are given in Table 2.4. Hence, a colour monitor with a resolution of 1024 × 1024 and displaying 256 colours simultaneously, would require a frame buffer of 1 M bytes.

Table 2.4 *Typical buffer memory sizes for various colours required*

Resolution	*Colour depth*		
	256 colours (8-bit)	*65,000 (high colour, or 16-bit)*	*16.7 million colours (true colour, or 24-bit)*
640 × 480	4 MB	4 MB	4 MB
800 × 600	4 MB	4 MB	4 MB
1024 × 768	4 MB	4 MB	4 MB
1280 × 1024	4 MB	8 MB	8 MB
1600 × 1200	8 MB	8 MB	16 MB

Example 2.1 Examine a colour graphics display of 1280 × 1024 resolution with 256 colour display with a refresh rate of 75 Hz.

256 colours require 8 bits (1 byte) per pixel.

Buffer memory size = 1280 × 1024 = 1280 kb = 1.25 MB

With a refresh rate of 75 Hz, the memory will be accessed 75 times per second.

The memory read rate = 75 × 1.25 = 93.75 MB/s

$$\text{Time available for displaying one line} = \frac{1}{75 \times 1024} = 13.02\ \mu\text{s}$$

$$\text{Time for reading one byte} = \frac{1}{93.75 \times 1024 \times 1024} = 10.17\ \text{ns}$$

The above shows the need for extreme fast rates of the memory to be used for graphic displays. The sweeper would access the frame buffer periodically for scanning one horizontal line or part of the line depending on the design of the sweeper electronics. After the scanning is completed, it would again access the frame buffer for the next data. This process would so continue. At the same time, the graphic processor would access the frame buffer to write the display. There should be no contention between the graphic processor for writing the information and the sweeper for reading the information from the frame buffer at the same time, and hence the need for synchronisation. If this synchronisation fails then it is possible for the frame buffer to get corrupted.

The time slice available for data manipulation of the frame buffer by the computer is severely limited as the resolution of the display is increased since more time would be required for screen refreshing. To obviate this problem, dual-ported frame buffers can be made using video RAMs. They contain a small shift register in the range of 256 bits into which any memory contents can be shifted while other portions of the memory are accessed in the normal manner. The sweeper would access the frame buffer through these shift registers whereas the graphic processor would access in the normal manner. Thus, video RAMs (VRAM) enable further increasing the resolution of the raster-scan display devices.

A comparison of the salient aspects of the various CRT display devices is presented in Table 2.5.

Table 2.5 *Comparison of graphic terminals*

	DVST	*Stroke writing*	*Raster Scan*
1. Image generation	Stroke-writing	Stroke-writing	Raster Scan
2. Resolution	4096 × 4096	4096 × 4096	2048 × 2048
3. Picture quality	Excellent	Excellent	Good
4. Contrast	Low	–	High
5. Selective erase	To a limited extent	Yes	Yes
6. Interactive	No	Yes	Yes
7. Colour capability	To a limited extent	Yes	Yes
8. Refresh memory	Not required	Medium	High
9. Animation	No	Yes	Yes
10. Price	Average	High	Low

2.5.2 Plasma Panel Display

Though the CRT display is highly refined, it is sometimes not suitable particularly for portable applications because of the depth that is necessary for the cathode-ray tube. In such situations, the plasma panel has been found to be useful though in a limited sense. Plasma uses a neon gas in a glass envelope with electrodes fore and aft to display the image. Though they are small and flat, they consume a large amount of power and also the resolution is not very good.

2.5.3 Liquid Crystal Display (LCD)

Liquid crystals exist in a state between liquid and solid. The molecules of liquid crystal are all aligned in the same direction, as in a solid, but are free to move around slightly in relation to one another, as in a liquid. Liquid crystal is actually closer to a liquid state than a solid state, which is one reason why it is rather sensitive to temperature. The array of liquid crystals becomes opaque when the electric field is applied, for displaying the image. Their use as display devices has been made popular by their widespread use in portable calculators and in laptop or portable computers. Their full screen size with reasonably low power consumption has made them suitable for portability. Another advantage is that they occupy very small desktop space while reducing the power consumption.

The working of LCD is different from that of a CRT. LCD makes use of a type of liquid crystal that exists in what is called the *nematic phase*. A nematic liquid crystal is a transparent or translucent material that causes the polarisation of light waves to change as the waves pass through the liquid. The intensity of the applied electric field changes the extent of polarisation. Nematic comes from a Greek prefix '*nemato',* meaning 'threadlike' and is used here because the liquid crystal molecules align themselves into a threadlike shape.

Twisted nematic (TN) liquid crystal has a natural twist at the molecular level, and the degree of twist is highly controllable by applying an electric current to the liquid crystal. By applying electric current in varying degrees, the liquid crystal reacts predictably on a molecular level in such a way as to allow or disallow the passage of light. A typical nematic liquid crystal produces a 90-degree shift in the polarisation of the light passing through when there is no electric field present. When an electric field is produced in the liquid, it affects the orientation of the molecules. This causes the polarisation shift to be reduced. The effect is slight at low voltages, and increases as the voltage increases. When the applied voltage reaches a certain level, the polarisation shift disappears entirely. This affects the light transmitted by the liquid crystal. Because their light transmission properties can be deliberately varied as a function of applied external voltage, nematic liquids are used in alphanumeric liquid-crystal displays.

LCD consists of two sheets of polarised glass. Each sheet of glass has a film on one side which gives the glass its polarising properties, while a special polymer is applied on the other side which will set small, microscopic grooves into the surface. The grooves must be in the same direction as the polarisation of the glass. A coat of nematic liquid crystal is added on top of this polymer such that the molecules of the crystal align with the direction of the microscopic grooves. Then, the two pieces of glass are arranged in such as way that the polarisation of the second sits at a right angle from that of the first. Since the liquid crystal is twisted nematic, each molecule is slightly twisted as compared to the level right below it. The uppermost level of molecules will be aligned with the grooves on the upper glass, making them positioned at a 90-degree angle from the ones on the bottom.

If the molecules are twisted then the light will pass while no light will pass through when they are untwisted, because light will not pass through the polarisation film. By applying an electric charge to the TN, it will untwist. Electrodes are embedded into the LCD to administer the electric current. When a current passes through them, the liquid crystal untwists, blocking the light and when the current is not passed, the reverse takes place. Since the LCD does not produce light and only transmits or hides it, it needs to have a light source. LCD monitors are backlit using either built-in flourescent bulbs or LEDs. A white panel behind the LCD diffuses the light so that it is spread out evenly over the entire display surface.

For an LCD to produce colour, each pixel on the screen has to have three sub-pixels, each being a primary colour (red, blue and green). The light from the LCD passes through colour filters (like CRT monitors). A transistor applies voltage to liquid crystals that sets their spatial alignment. Light changes its polarisation angle when it passes through the ordered liquid crystal molecular structure and depending on its new polarisation angle, it will be absorbed completely or partially. This generates the required colour.

There are two types of displays used in LCDs. In the *active matrix display*, a polysilicate layer provides thin-film transistors at each pixel, allowing for direct pixel access and constant illumination. As a result, this is also known as a Thin-Film Transistor (TFT) display. It is more popular because of its low manufacturing costs. However, it has some drawbacks. The biggest problem is that black looks more like dark grey on the old panels that result in poor contrast. The second problem occurs when a transistor dies leaving a bright dead pixel on the screen, which is much more noticeable than the black dead one.

Alternatively, the *passive matrix LCD* has a grid of conductors with pixels being located at each intersection in the grid. In order to light a pixel, current is sent across the two conductors. Active matrix requires less current to control the luminance of a pixel. As a result, the current in an active matrix display can be switched on and off more frequently, improving the screen-refresh time, for example, the mouse will appear to move more smoothly across the screen.

The prices of these displays are falling rapidly, and they are thus becoming popular for desktop applications as well. Also, large screen LCD monitors of up to 24 inches in size are available currently at reasonable prices, thus replacing the CRT terminals in most cases.

Aspect Ratio CRT terminals were all produced with an aspect ratio (ratio of longer side to the shorter side) of 4:3. However, the larger LCDs (> 19 inch) have a wide format with a 16:9 aspect ratio. This format provides a longer screen while reducing the height, thereby reducing the amount of scrolling one has to do for wider documents. This will be good for some applications where the width is more, for example, for draughting work. Also, the users can watch any widescreen DVD movie in its original filmed format.

2.6 HARD-COPY DEVICES

Once the output is finalised on the display device, it can be transformed into hard copy using:

- graphical printers,
- plotters, or
- photographic devices.

2.6.1 Graphical Printers

This is the fastest way of getting graphical output at low cost. The three principal technologies that are currently used are the following:

Impact Dot-Matrix Printer In this printer, the print head consists of a vertical bank of needles (9, 12 or 24) which move horizontally over the paper. At each of the horizontal positions, any of the pins in the print head can make ink marks by hitting the paper through a ribbon. Thus, from the image on the screen, each of the pixels corresponds to the pin position as it moves over the entire page. The resolutions that are available vary but range from 60 dots per inch to 240 dots per inch. Their cost is comparatively low, but a major disadvantage is their noise because of the impact of the pins on the paper. They are almost obsolete for graphic printing.

Thermal-Transfer Printer This is similar to the dot matrix printer in operations, but is not to be confused with the normal thermal printers where sensitised paper is used for output. It uses a special ribbon positioned between the paper and print head. The ribbon is a roll of thin polymer material. Spots of the dye are transformed from the heat-sensitive ribbon to the paper underneath. Though they are relatively noiseless, with fewer moving parts and a low weight, the cost of the special ribbon to be used is high and it is still a developing technology. It is normally used for field applications where portability is required. Colour thermal transfer printers from a number of vendors are to be found in the market with resolutions of 150 to 400 dots per inch.

Inkjet Printer This does not make use of any ribbon but shoots a jet of ink directly onto the paper, as the pin impact of the dot-matrix print head. Normally, there would be a bank of ink nozzles positioned vertically, as the pins of an impact dot matrix print head. Otherwise, the rest of the mechanism is identical to the impact dot-matrix printer. These are almost noiseless in operation. The print head of the Hewlett-Packard inkjet printer consists of an ink cartridge holding 3 ml of ink in a cylindrical unit of 40-mm length which is enough to last for 500 A4 size papers. Ink from the reservoir terminates in 12 tiny holes, arranged vertically facing the paper. As the print head moves horizontally, the droplets of ink are shot wherever required. Behind each hole is a small heating element which when turned on vaporises the ink partially, causing a force inside the cartridge, to eject the ink onto the paper. Resolutions can be in the order of 300 to 1200 dpi (dots/inch). The only requirement is that the paper used should be sufficiently absorbent, so that the droplet upon reaching the surface of the paper dries quickly. It is also possible to have full colour printing using the inkjet technology by incorporating the primary colour inks for each of the dots.

Each ink cartridge in a Lexmark 7000 series printer has an integral laser-crafted print head, which delivers 1200 × 1200 dpi. It consists of 208 nozzles in the print head of the black cartridge, and 192 nozzles in the colour and photo cartridges. These nozzles let the 5700 create very small, precise dots of ink on the page, giving sharp text and fine line detail.

Canon bubble jet printers have a Black BJ Cartridge with 608 nozzles, the colour BJ Cartridge with 240 nozzles (80 nozzles × 3) and the photo BJ cartridge with 480 nozzles (80 nozzles × 6) giving a print resolution of 1200 × 600 dpi.

The Calcomp CrystalJet printer comes with four print heads, one for each of the four process colours: cyan, magenta, yellow and black with each print head having 256 nozzles. The heads are staggered in stairstep fashion and mounted on a gold-plated nozzle plate. The print heads are spaced at 1/180th-inch intervals, yielding a print swath of 1.4 inches per colour. The printer can have a resolution of 180, 360 or 720 dpi.

One of the major problems with inkjet printers is the cost of the inkjet cartridge used per page.

Laser Printer The laser printer is essentially an electrostatic plain-paper copier with the difference that the drum surface is written by a laser beam. A semiconductor laser beam scans the electrostatically charged drum with a rotating 18-sided mirror (560 revolutions per minute). This writes on the drum a number of points (at 300 dots per inch), which are similar to pixels. When the beam strikes the drum in the wrong way round for printing a positive image, reversing it, then the toner powder is released. The toner powder sticks to the charged positions of the drum, which is then transformed to a sheet of paper and bonded to it by heat. Though it is relatively expensive compared to the dot-matrix printer, the quality of the output is extremely good and it works very fast at 8 to 16 pages per minute (A4 size). The current resolution available is 600 and 1200 dpi. Currently, the size is limited only to A3 or A4, though higher size electrostatic plotters with a slightly different technology are available but are very expensive. Operationally, the laser printers are fast, as well as the copy cost is low compared to inkjet printers. However currently, the output paper size is the limitation.

Colour Laser Printers Colour laser printers operate in the same principle as that of the monochrome laser printer. They are provided with 4 toners: black, cyan, yellow and magenta. The colour is printed by three additional passes through each of the toner colours. It is far more complicated than a monochrome printer since the paper needs to go through multiple passes for all the colours to be registered. Any small misalignment during these individual passes will result in unintended colour fringing, blurring, or light/dark streaking along the edges of the coloured regions. The new colour laser printers have the separate drum and toner assembly for each of the colour. The paper passes through each of the colours like an assembly line getting the required colour in the process. They also require large memory since the raster data for the colour images have to be generated for the individual colours before printing can begin.

2.6.2 Plotters

The plotter is the widely accepted output device for the final output. A large range of plotters of varying sizes and prices (see Table 2.6) are available. The accuracies achievable are very high and the plots can be made on all types of media such as paper, tracing paper and acetate film. Normally, all plotters have a range of pens available, which can be changed under program control. Pens of any colour or of different width writings can be used depending upon the output desired. The types of pens used are fibre tip, roller ball or liquid ink. They are the slowest of all the high-resolution plotters since the speed is dependent upon the pen's ability to draw lines.

Table 2.6 *Various sizes for plotters*

Designation	*Size of drawing, mm*
A0	841 × 1189
A1	594 × 841
A2	420 × 594
A3	297 × 420
A4	210 × 297

Pen Plotters Essentially, plotters are of two types—flat-bed and drum-type. In the flat-bed plotter, the paper is held in a fixed position by means of a vacuum or electrostatic force. The pen carriage moves in both X- and Y-axes for making the necessary plot. Its chief advantage is that any kind of paper is acceptable because of the simple nature of the plotter. However, these are very expensive compared to a similar size drum plotter. Moreover, the plot size is limited by the bed size of the plotter.

The drum plotter is slightly more complex, the *Y*-motion of the plot being obtained by a rotating drum on which the paper is held with the help of sprocket holes of a standard size. The *X*-movement of the pen is arrived at by moving it in a direction perpendicular to the drum motion. As the paper is moved during the plotting, the size of the plot to be obtained can be varied by the program. As a result, the drum plotter would be cheaper for a given drawing size compared to a flat-bed plotter. The disadvantage lies in the use of special paper with proper sprocket holes for the plotter.

The plotters of the present generation are almost all of the drum type with the variation that the paper feeding is done by means of friction feed (pinch rollers). The pen plotters are low in initial purchase cost, and produce accurate drawings. However, they are slow and require a high level of maintenance.

Electrostatic Plotters The electrostatic plotter uses the pixel as a drawing means, like the raster display device. The plotter head consists of a large number of tiny styluses (as high as 21 760) embedded into it. This head traverses over the width of the paper as it rolls past the head to make the drawing. These styluses cause electrostatic charges at the required dot positions to make the drawing. The resolutions available may be of 100 to 508 dots per inch. They are normally very fast with plotting speeds of 6 to 32 mm/s, depending on the plotter resolution.

The speeds of different plotters vary with the acceleration of the pen as it draws the line and with the pen up/down cycle time. The factors to be considered while selecting the plotters are plotting area, number of pens, type of pens used, drawing speed, resolution, accuracy and drawing protocol (plotting graphics language such as HPGL) it observed.

Inkjet Plotters With the rapid developments in the inkjet technology in terms of the resolution, speed and cost, a majority of the plotters are now using the inkjet for writing rather than a felt-tip pen as was the case earlier with the plotters.

2.6.3 Photographic Devices

The photographic recording devices are essentially cameras in front of a CRT display. The only difference is that they use a display device other than that used as the monitor. They normally have a smaller built-in screen inside the recorder, which is connected to the CPU through the serial communication port (RS 232c). The image is obtained on a flat-faced monochrome CRT with a 4-position filter wheel to provide separate exposure for red, green and blue images. The image from the main computing system is received by these recorders, and then separate graphic processing is effected to remove the jagged edges on type fonts, circles and lines. It would normally be possible to enhance the image obtained on the display device through the software of the recording devices. The resolutions that are attainable vary from 500 lines to 4000 lines depending on the cost of the device and the presence of any other hardware that is required along with the device.

2.7 STORAGE DEVICES

Permanent storage of programs and of data generated during various sessions of CAD/CAM requires a large amount of storage space. This is normally denoted as auxiliary storage and the various devices used are

- Floppy disks
- Winchester disks
- Magnetic tapes
- Magnetic tape cartridges

- Compact disk ROMS
- DVD

The *floppy disk* is the most convenient medium for handling data that is either temporary or permanent. It consists of a storage disk, which is magnetically coated on both sides. It is permanently enclosed in a square cover, lined internally with a special low-friction material. The floppy disk rotates at very high speeds, and the read/write head moves radially to read or write any data randomly from any location. The disk is divided into concentric rings called tracks. They are usually specified in tracks per inch (TPI) and are generally 48 TPI or 96 TPI. The tracks are further subdivided into radial sectors. Each sector can store information of about 250 bytes. They normally come in 3 standard sizes—3.5, 5.25 and 8 inches. The storage capacities range from 360 k to 1.5 M bytes (formatted) storage. The formatted storage is the actual value of storage in bytes available for the user. Because of the limited storage, a floppy disk is almost obsolete with very few computers using it.

The *Winchester disk* is a thin, rigid metal disk on both sides of which the magnetic medium is coated. Several such hard disks are put together aligned on a central shaft disk pack as shown in Fig. 2.15. Separate read/write heads for each of the disks are permanently aligned and then the disk is sealed. Though there are more than one head, at any given time only one head would be accessing the disk for data input/output. This has a large storage capacity and has been extensively used because of its low access time, low cost and compact size. The disk is fixed inside the drive and therefore, the storage is fixed to the capacity by the drive. Typical storage sizes available presently are from 100 GB to 1000 GB. However, removable disk packs are available whereby a large amount of data can be stored and used for back-up purposes.

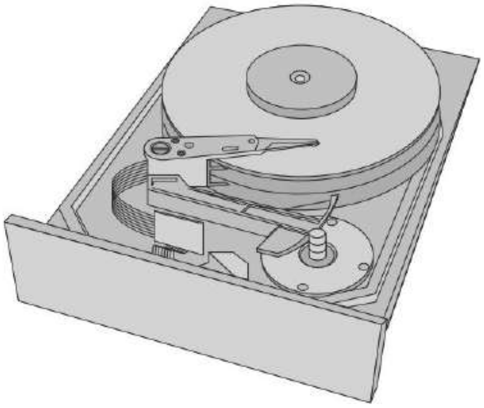


Fig. 2.15 *Hard disk*

The other kind of mass storage medium associated with computers is the 0.5-inch *magnetic tape*. The greatest limitation of magnetic tape is the serial nature of the storage, necessitating all the tape before to be wound for accessing any inside information. As a result, the magnetic tape is used only for data exchange or back-up, as a large amount of storage is possible. A typical storage density is 6250 bits per inch in 9 tracks. A 10.5 inch reel can store about 180 M bytes of data.

Another magnetic tape, 0.25 inch in size, in the form of a cartridge as in an audio cassette is also used by microcomputers and minicomputers as a mass storage medium. These are essentially used as a back-up medium for taking periodical back-up from the hard-disk drive.

Current developments in mass storage of very large capacities are based on the optical technology, rather than on the magnetic principle, which was used in all the previous storage devices. In this, a small aluminium compact disk, 120 mm in size, contains a number of pits in the size range of about 1.5 microns. A non-contact laser reads the information present on the disk. As a result, the storage densities can be very high. For example, a 120-mm compact disk can store 650 M bytes of data and is very easy to handle. Since the reading is done by the focussed laser onto the pits, the information stored is permanent. For writing onto the disk, using the same technology, a high-intensity laser is required to evaporate the material on the disk. This can only be done once and the drives used for this purpose are called WORM (Write Once and Read Many) drives and are now commonplace in most of the computers.

The major disadvantage of this technology is that once the disk is written by making a pit, it cannot be erased. As a result, it becomes like a ROM and hence the name for these devices as CD ROM. Thus, these devices cannot be used as a regular auxiliary storage device with the computer but can be used for only database purposes. However, current research provides a better medium for storage in the form of magneto-optical media. The recording-medium orientation changes, causing the light passing through it to change in brightness. These media would provide a convenient means of erasing and storing large amount of data as a regular auxiliary storage medium.

The media of these types of storage systems consist of a polycarbonate platter in which is embedded a layer of aluminium backing, overlaid with a magneto-optical substrate. This substrate crystal orientation is changed magnetically to erase or write. The writing process consists of three passes. In the first pass to erase the written matter at a particular location, a magnetic field is applied on the platter for erasing the stated direction. A high-power laser heats the platter to a temperature called *Curie point* at which the crystals in the substrate orient magnetically to the surrounding magnetic field, which is a 0. Thus, the data in the target location in the platter is erased. Next, the applied magnetic-field orientation changes to the writing orientation, i.e., 1. The laser again heats the same location to the Curie point such that the crystal orientation is now altered in the direction 1. The third pass is required to verify that what was written is the same as the data.

For the purpose of reading, the magnetic field is removed and a low-intensity laser beam is directed, which passes through the substrate and is reflected by the aluminium backing. The crystal orientation in the magneto-optical substrate alters the polarisation of the reflected beam, called the *Kerr effect*. The reflected beam is passed through a polarising filter onto a photo-detector, the intensity of which determines whether a 0 or 1 is present at the target location. Presently, commercial drives are available which can store 650 MB on a 5.25-inch removable optical disk using this technology. This can also be read by modern CD-ROM drives. These are called CD-RW drives and are currently available. These are slowly replacing the existing floppy diskettes for data transfer and archiving.

Another storage medium which is becoming increasingly popular is the *DVD*, originally named Digital Video Disk', then 'Digital Versatile Disk', and now simply 'DVD'. There is no official three-word equivalent to DVD. Like CD-ROM, the DVD is read by an infrared laser focused through a protective plastic layer onto the disk's reflective layer. The transparent layer is 1.2 mm thick on a CD-ROM, but only 0.6 mm on a DVD-ROM. The beam reflects off pits burned into the reflective layer by the recording laser and is passed through optics to the pickup. The laser beam used on a CD-ROM player has a wavelength of 780 nanometres. DVD players employ a laser with a wavelength of 635 and 650 nanometres, designed to read through the thinner 0.6-mm transparent layer. This makes it possible to focus on smaller pits of digital data, about half the physical size of pits on a CD-ROM—effectively doubling the density of pits on a DVD-ROM. More data is squeezed onto the disk by recording tracks closer together and closer to the centre hole, as well as improving the error-correcting decoding algorithms.

DVD disks come in capacities of 4.7, 8.5, 9.4 and 17 GB. Some of the early disks are single-sided, but the specification includes dual-layered and double-side versions that define the four levels of storage capacity. DVD data is read by a variable-focus laser; on dual-layered disks, a lens shifts the beam's focus from the pits on the outer layer to the pits on the inner layer. A comparative evaluation of the CD-ROM and DVD technologies is given in Table 2.7.

Currently, there are three different types of DVD drives that have been defined:

- **DVD-ROM** These are the drives with only reading capability. They are used basically for removable mass storage for large volumes of data such as encyclopaedia, and are currently available.

Table 2.7 *Comparison of CD-ROM and DVD storage characteristics*

	CD-ROM Disk	*DVD Disk*
Disk diameter	120 mm	120 mm
Disk thickness	1.2 mm	1.2 mm
Centre-hole diameter	15 mm	15 mm
Disk structure	Single substrate	Two bonded 0.6-mm substrates
Laser wavelength	780 nm	650 and 635 nm
Lens aperture	0.45	0.6
Track pitch	1.6 micron	0.74 micron
Shortest pit length	0.834 micron	0.4 micron
Average bit rate	0.15 MB/s	4.7 MB/s
Data capacity	650–680 MB	Single-side/single-layer: 4.7 GB Single-side/dual-layer: 8.5 GB Double-side/single-layer: 9.4 GB Double-side/dual-layer: 17 GB
Data layers	1	1 or 2
Video compression	MPEG-1	MPEG-2
Audio compression	MPEG-1	Dolby digital

- **DVD-R** These are drives with write-once capability. DVD-R drives are also called ‘Write Once, Read Many (WORM) drives and are currently available. These are similar to the CD-R drives with WORM capability.
- **DVD-RAM** These are drives with both read and write capability. DVD-RAM drives are also called Write Many, Read Many (WMRM) drives. Unfortunately, there is no agreed format in this category. As a result, there are a number of different formats that are being pushed by the various groups in the DVD forum. What was approved by the forum is a phase-change design that can hold 2.6 GB of data per side on single- or double-sided disks. The single-sided disks will come in removable cartridges, but to protect the sensitive recording layer, double-sided disks will be permanently mounted in cartridges.
- **DVD+RW** This is supported by Hewlett-Packard, Philips and Sony. DVD+RW’s single-layer phase-change disks have more capacity than DVD-RAM disks—4.7 GB per side, and use a higher-density recording process. The DVD+RW format does not rely on cartridges to hold the disks.
- **DVD-R/W** This is being put forward by Pioneer and will be the first one to be available commercially. It will use random-access media that holds up to 4.7 GB. One of this technology’s key characteristics is that its phase-change media have a higher reflectivity, and as a result, can be read in existing DVD-ROM drives and DVD players without modification.

Currently, there is no standardisation in the DVD formats. Both the +R and −R are currently available, and there is not much of difference between the two formats. Many drives are available that can write in both the formats, so that the user has little to worry about in terms of the format and usability.

The next formats for DVDs have been developed that promise much higher storage capacities. There are two formats, HD DVD and Blu-ray. HD DVD is invented by Toshiba and is approved by the DVD forum. Blu-ray is developed by a group of companies that include Apple, Dell, Hitachi, HP, JVC, LG, Mitsubishi, Panasonic, Pioneer, Philips, Samsung, Sharp, Sony, TDK and Thomson. A comparison of these different formats is given Table 2.8.

Table 2.8 *Comparison among different DVD characteristics*

Parameters	*Blu-ray*	*HD DVD*	*DVD*
Storage capacity	25 GB (single-layer) 50 GB (dual-layer)	15 GB (single-layer) 30 GB (dual-layer)	4.7 GB (single-layer) 8.5 GB (dual-layer)
Laser wavelength	405 nm (blue laser)	405 nm (blue laser)	650 nm (red laser)
Numerical aperture (NA)	0.85	0.65	0.60
Pit size	0.13 μm	0.20 μm	
Disk diameter	120 mm	120 mm	120 mm
Disk thickness	1.2 mm	1.2 mm	1.2 mm
Protection layer	0.1 mm		0.6 mm
Hard coating	Yes		No
Track pitch	0.32 μm		0.74 μm
Data transfer rate (data)	36.0 Mbps (1x)	36.55 Mbps (1x)	11.08 Mbps (1x)
Data transfer rate (video/audio)	54.0 Mbps (1.5x)		10.08 Mbps (<1x)
Video resolution (max)	1920 × 1080 (1080p)	1920 × 1080 (1080p)	720 × 480/720 × 576
Video bit rate (max)	40.0 Mbps	40.0 Mbps	(480i/576i) 9.8 Mbps

The main advantage of HD DVD is that it is very similar to the DVD technology, and as a result, it is possible to migrate with very minimal cost for the production of HD DVDs. The downside is that HD DVD has lower capacity of 30 GB for the dual-layer disk. Also, the HD DVD players are less expensive. In contrast blu-ray DVD has a higher capacity of 50 GB for a dual-layer disk with a possibility of going up to 200 GB with 8 layers. The cost of the players is higher; however, with volume sales it is expected to come down. By mid-2008, Toshiba withdrew its support to the HD DVD format. Thus, for high storage only the blu-ray format is the de-facto standard.

2.8 SOFTWARE

Software represents that segment of the computing system which determines the way the computer is to be used. Better software makes for a better utilisation of the computing system. Various software items, which form part of any computing system, are shown in Fig. 2.16.

2.8.1 System Software

This represents the essential part of the software without which no computer system can operate. The *operating system* generally forms part of the hardware, and together with it provides for the use of all the hardware elements in an optimal manner. The operating systems are generally proprietary with the hardware that is being used. Examples are, VM for IBM computers, VMS for DEC computers, CP/M, PC-DOS (MSDOS), Windows 95 and Windows 98 for microcomputers, Windows NT, UNIX and LINUX. UNIX and LINUX are the operating systems, which are generally considered independent of hardware.

The *editors* are meant, as the name indicates, for creating and modifying disk files; a *linker* is to be used for linking the object modules that are developed into a coherent executable module; a *debugger* is to be used in program development to identify the logic and run-time errors.

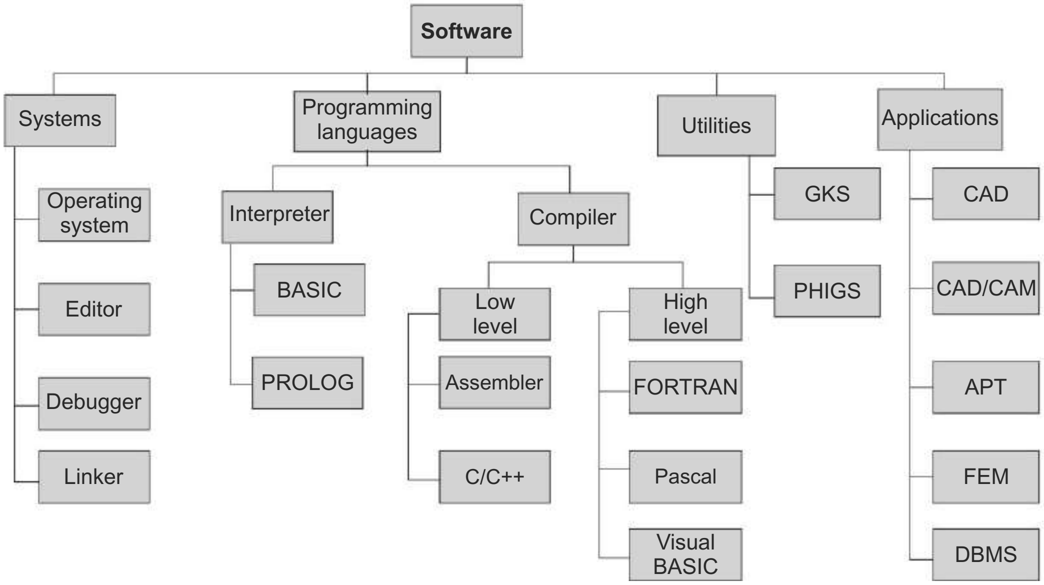


Fig. 2.16 *Software classification*

2.8.2 Programming Languages

The next major segment of software is the programming languages, through which software development takes place. Various languages have been developed to meet the different requirements of the applications. The programming languages are essentially translators and can be broadly classified into interpreters and compilers. *Interpreter* is a system in which the language is translated and then executed immediately. Thus, if the same statement is encountered a number of times during the execution of a given program, it has to be translated on every occasion. However, in a *compiler*, the complete program is translated once into the machine language and it can be executed any number of times. Interpreters are good for program development since the execution can take place immediately, but they are slow in view of the repeated translations that are needed. It is preferable to develop the program using an interpreter, and when it is bug-free it may be compiled to generate the directly executable machine language code for faster usage. The present trend in most of the programming languages is to provide an integrated environment consisting of an interpreter, compiler and context-sensitive editor for the particular language. This would help in the rapid development of programs since no compilation and linking of the programs takes place during the development stage.

Traditionally, FORTRAN (FORmula TRANslation) has been used as a programming language by the scientific community from the beginning. But now, BASIC (Beginner's All purpose Symbolic Instruction Code), PASCAL and C are being used for CAD/CAM program development with C being the one used for system development in view of its tight code and faster execution on various systems. With artificial intelligence being increasingly used by the scientific community, LISP (LISt Processing) and PROLOG have also been used for some modules related to CAD/CAM.

Object Oriented Programming (OOP) has now become the norm of most of the programming. The objects are basically reusable code that can be used in many of the programs. By the careful designing of objects, it is possible to reduce the system-development time greatly, particularly for large programming projects.

2.8.3 Utilities

The utilities refer to a set of small programs which the applications developer can incorporate in his program for performing any specific task. Examples could be the numerical procedures, matrix operations, etc. These are essentially the many repeated applications which can once for all be made available as utilities and linked with any particular program, rather than trying to develop them anew every time they are needed.

2.8.4 Applications

Finally, these refer to the application programs, which are generally stand-alone programs that are meant for doing specific tasks. Examples are, word processing, database management, computer aided design, etc. In the later chapters, we will discuss more of these that relate to CAD/CAM.

2.9 SYSTEM CONFIGURATION

Traditionally, computers have been classified as mainframe, minicomputers and microcomputers. The classification was based on the word length used, main memory available, and other such features. The developments in microprocessors that have taken place in the last few years have gradually decreased these differences with the lower-end microcomputers acquiring the higher-end facilities such as larger word length, higher memory addressables, and so on.

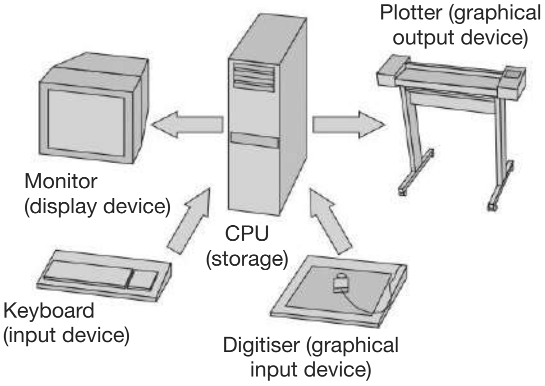


Fig. 2.17 *Typical configuration of a workstation*

The stand-alone workstation which normally is a 32- or 64-bit microprocessor-based system having all the necessary hardware facilities at the local level (see Fig. 2.17) is now commonplace among design professionals. The microprocessor that is most popular in CAD/CAM workstations is the Intel Pentium family processors running up to 3 GHz.

The Intel Pentium based workstations are available at a low cost because of their large volume production. Based on the discussion in the earlier pages, it would be possible to form the configuration of a suitable system based on the needs of CAD/CAM. A typical system for graphical applications may be as shown in Fig. 2.17. The hard-disk storage needs to be high because of the large amount of storage required for the drawings, particularly if one thinks in terms of sub-assemblies and assemblies involving a large number of individual parts. Further, the hard disks have faster access times for writing and retrieving the data.

The digitiser becomes a necessity for not only transferring data in the case of existing drawings, but also for having standard menu items being permanently displayed. Normally, the menu would be shown on the display screen, but because of the small size of the screen, it becomes difficult to show all the menu items at the same time. As a result, the display needs to be modified and this slows down the design activity once the user is familiar with the designing system. Hence, the menu can be made as part of an overlay on the digitiser, which can then be easily accessed by the help of the puck or digitiser stylus.

In a typical design office, there may be a number of designers who would like to work simultaneously on various aspects. Hence, there would be a need for a number of workstations. When duplication of workstations is done, it is not necessary to duplicate all their resources as that would be prohibitively expensive. It therefore becomes necessary that such resources that will be used by all the users will be grouped together and connected to a high-speed computer termed as server as shown in Fig. 2.18. Typical facilities that will form

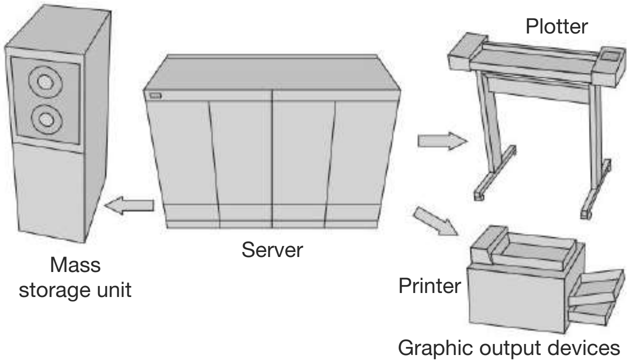


Fig. 2.18 *Configuration of a server*

part of the server are mass storage units along with backup facilities. Also, the large volume graphic printing facilities form a part of such a server.

All the facilities of the server may have to be continuously accessed by any one workstation. Further, it is possible that data may have to be transferred from one workstation to another. It would therefore be imperative that all the workstations be connected in some manner so that all of them play a coherent part in the design process. A typical form in which such a set-up can be achieved is shown in Fig. 2.19, where the individual workstations, termed *clients* will be connected to the server through some kind of a networking arrangement.

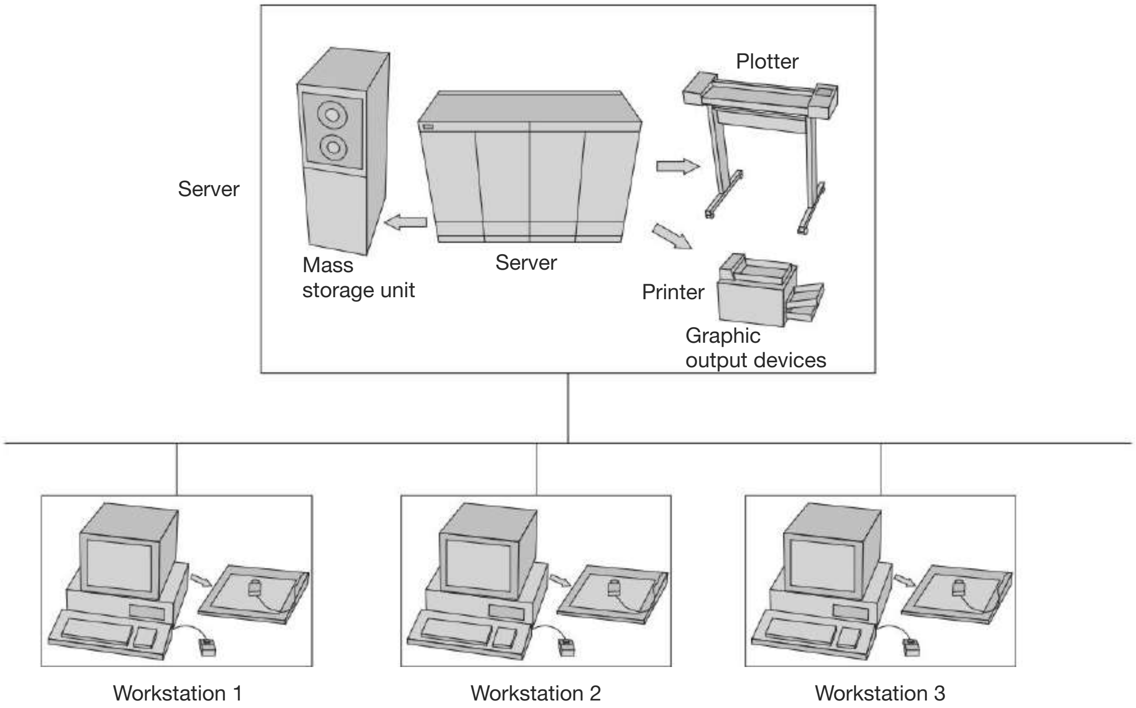


Fig. 2.19 *Configuration of local workstations*

The joining of a number of computing systems is termed *networking*. Local Area Networking (LAN) is a special case of networking where all the items forming the network are located close to one another, compared to the general term where the computing system can be located anywhere.

There are essentially three types of network layouts or topologies that are commonly used. The differences are mainly in the ways of switching, the speed of transmission and the types of cabling used.

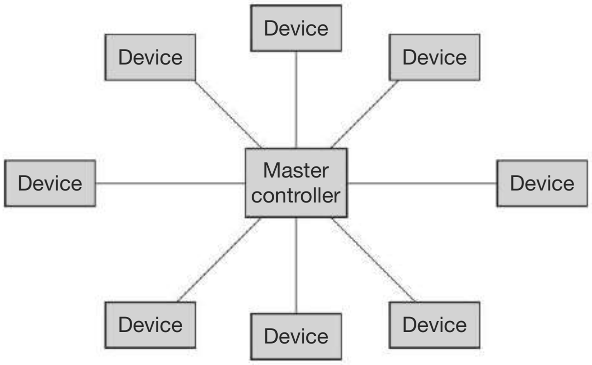


Fig. 2.20 *Star-network topology*

A *star network* is one in which all the links are connected to a central point, which is essentially a small exchange that switches the information from one machine to the other and is shown in Fig. 2.20. This type is preferable only when the system does not have to handle a lot of high-speed traffic. The only disadvantage is that if the central serving station (switching) develops a fault, the whole network fails. Hence, it is not generally used.

The other types of networks offer distributed control, so the risk of total network failure is reduced. The *bus network* is the most commonly used networking system mainly because of the success of the Ethernet developed by Xerox in the early 1970s. A bus network looks like a branch of a tree with the computing equipment attached like leaves to the twigs (Fig. 2.21). Each of the equipment has its own unique address by which the information flowing in the bus is picked up. Thus, any equipment (e.g., computing terminal, normally called a node) can send data at any moment to any address. However, if the bus is occupied by the data being sent (collisions occurring between two packets of data) the information will be notified and the equipment has to resend the data. The data transmission in this type of network is of one of the highest speeds available, up to 10 Mbits per second.

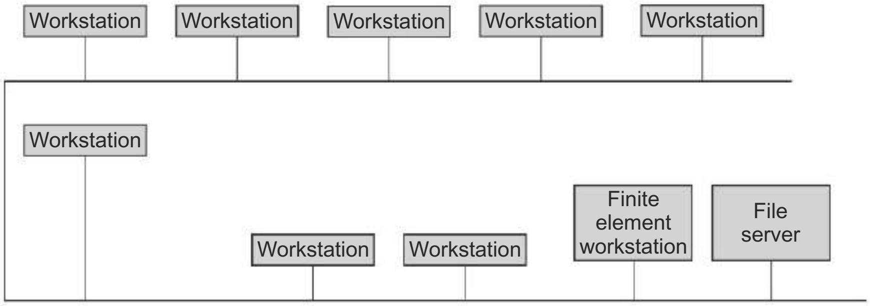


Fig. 2.21 *Bus network*

Ring networks are one of the cheapest options available as they are based on simple and inexpensive cables. The devices forming the network are attached in a ring-like structure as shown in Fig. 2.22. The data moves in the ring and when the correct address comes up, it is picked by the device. There is no single terminal to act as a server and all the nodes can transmit and receive data through the network. Some definitions of the terms that are generally encountered in communication between computers and peripherals are presented in Table 2.9.

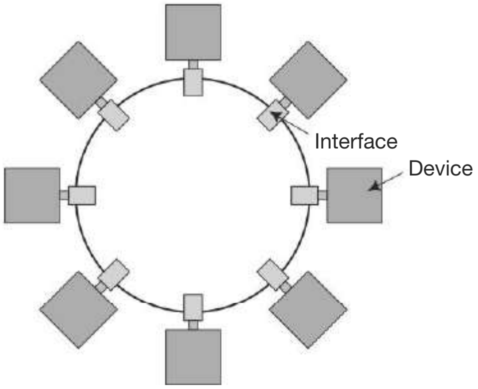


Fig. 2.22 *Ring network*

Table 2.9 *Some definitions in communications*

The following are terms and definitions one frequently comes across in communication between workstations.

Multiplexers This is a hardware unit, which helps a number of computers in sharing a single transmission line.

Communications processor An interface between the CPU and the multiplexer. It assembles the information coming from the data highway in a proper manner and passes it on to the CPU.

Concentrator or buffer This is normally used with asynchronous devices, though it can also be used with synchronous devices. It temporarily stores information from the CPU and passes it on to the device or vice versa, depending on the speed of transmission desired.

ASCII It stands for American National Standard Code for Information Interchange. It represents a method of binary coding for information characters, which are to be either stored by the computer or used for transmission across.

Data transmission Data transmission between any two devices can take place in either synchronous or asynchronous manner. In synchronous or parallel transmission (e.g., Centronics parallel, GPIB, IEEE 488), a complete character is transmitted at a time. However, in asynchronous or serial transmission (e.g., RS 232c), the data is transmitted bit by bit for each of the characters. This involves sending identifying codes for the character bits (start bit and stop bits) which relatively decrease the speed of transmission.

Broadband Broadband splits the transmission bandwidth into channels such that more than one transmission can occupy a line at any single point of time. It is more reliable and is generally used for outdoor transmission.

Baseband Baseband carries a single transmission at any given time without any disruption. It is employed for Local Area Networks (LAN).

The detailed specifications for the software items will be discussed in Chapter 3.

Summary

- The capability of a computer system makes a lot of difference in terms of the ease with which a CAD/CAM system can be operated. Careful choice of the specifications for the system is an important task for planning a CAD/CAM system.
- The basic structure of a computer is modular with the central processing unit, memory, input and output devices being the major components.
- Microprocessor capability has been increasing at an exponential rate, with the latest 32- and 64-bit processors running above 3 GHz and having direct memory-addressing capabilities in the order of GB.
- Continuous developments have been taking place in the main memory with the DDR RAM that is currently the main type used in all workstations.
- Though there are a number of graphic input devices that have been tried, the mouse is the universally used input device with the digitiser being used for specialised work.
- The most common output device is the cathode-ray tube type with very high resolution as well as size. However, liquid crystal displays are increasingly becoming common in view of their short footprint and low energy consumption.
- Among the hard-copy devices, plotters are used for final output for archival purposes, while graphic printers such as inkjet and lasers are extensively used for regular and draft outputs.
- A variety of storage options are available involving magnetic and optical technologies. The size of the storage is also increasing to cater to the large file sizes that are associated with the CAD. Optical storage, particularly the DVD, is now becoming the mainstream usage with low cost and large size (4.7 GB) compared to a floppy disk (1.4 MB).
- CAD workstations are organised into networks to facilitate group working and information sharing.

Questions

1. What is the structure of a computing system?
2. What do you understand by the CPU?
3. Describe the functioning of a central processing unit with the aid of a block diagram.
4. What is the importance of a CPU in a computing system?
5. How do you distinguish between a CPU and a microprocessor?
6. Explain the following terms with reference to microprocessors:
   (a) Word (b) System clock
   (c) Address
7. Describe the various types of semiconductor memory devices used in microcomputers.
8. What are the input devices more commonly employed for general graphics applications?
9. Describe the following input devices:
   (a) Digitiser (b) Tablet
   (c) Mouse (d) Light pen
10. Specify a digitiser for CAD applications and justify your choice.
11. What are the various constructional methods employed in the making of a digitiser?
12. What are the various display devices that are used for displaying graphic information? Present their merits and demerits.
13. Explain the functioning of Liquid Crystal Display terminals as used in CAD.
14. Explain the importance of aspect ratio in computer display terminals.
15. What do you understand by the word 'interlacing' in connection with display terminals? Explain its importance.
16. What do you understand by raster scanning? Why is it preferred to the storage tube in the display of graphics information?
17. Give a brief description of scanners as used in engineering applications.
18. Explain the following terms:
   (a) Screen buffer (b) Scanning
19. Explain the factors which inhibit the use of very high resolution and a large number of colours for display in the case of raster-scanning display devices?
20. What are the types of printers that would be useful for printing graphic information?
21. What are the types of plotters?
22. How do you specify a plotter for graphics application?
23. Briefly describe the types of storage devices used in computers.
24. How would you classify software?
25. Discuss the following terms in relation to software.
   (a) Operating system
   (b) Utilities
   (c) Programming languages

Problems

1. How long does it take a dot-matrix printer with a speed of 200 characters per second to print all the characters displayed on a 640 horizontal × 480 vertical pixels monitor? Assume that the text is generated on the display using a grid of 8 × 20. How does this change if the text grid changes to 8 × 12?
2. A 60-Hz non-interlaced colour display terminal has a resolution of 1024 horizontal × 768 vertical pixels with the ability to display 256 colours simultaneously. Find
   (a) the RAM size of the bitmap (frame buffer)
   (b) the time required to display a scan line and a pixel

   (c) the optimal resolution design if the bitmap size is to be reduced by half

3. The resolution of the monitor of a computer is 512 horizontal × 320 vertical pixels, and it is mapped to a display of 300 × 200 mm. Show how a line of 80 mm length at 45° is to be drawn on the screen while starting with coordinates of (20, 20) looks in terms of pixels.
4. A physical space of (50, −15) to (185, 90) in mm is to be displayed on the screen with a resolution of 1280 × 1024 pixels with an aspect ratio of 1. Find the necessary mapping required from physical coordinates to pixels. Also, determine the pixel coordinates (only end points and no interpolations) of a line whose physical coordinates of end points are (125, 55) and (140, 60).
5. Make a study of the workstations you are using in the laboratory and prepare its specifications. Compare them with that of any PC in the laboratory.
6. If the frame buffer is limited to 256 Kbytes of RAM, what is the reasonable resolution given to the aspect ratio of 1? The display has an 8-bit colour plane and the screen size has an aspect ratio of 4 : 3.
7. A computer display system has a resolution of 800 horizontal × 600 vertical pixels. If the screen aspect ratio is 4 : 3, show how a square box of 400 pixels can be drawn.
8. A physical space of (0, 0) to (225, 125) in mm is to be displayed on a screen with a resolution of 1280 × 1024 pixels with an aspect ratio of 1. Find the necessary mapping required from physical coordinates to pixels. Also, determine the pixel coordinates (only end points and no interpolations) of a line whose physical coordinates of end points are (65, 55) and (160, 80).

3

COMPUTER GRAPHICS

Objectives

One of the important elements in any CAD/CAM system is the component of software related to the manipulation of geometric elements. The functionality of the user interaction of the CAD/CAM system is greatly affected by this component. After completing the study of this chapter, the reader should be able to

- Convert vector straight lines to raster images to be displayed on a raster terminal utilising the pixel information
- Understand the problems associated with displaying vectorial information on a raster terminal
- Various types of coordinate systems used in displaying CAD information
- The data requirements of a graphic image and the database storage methods used
- Learn about engineering data-management systems
- Different types of geometric transformations used during CAD geometry generation and display, and their evaluation
- Mathematics required to display a 3D image on the 2D screen of the display device
- Understand the problems associated with the display of graphic images in the display screen such as clipping and hidden-line elimination
- Learn about adding colour and shading to the display for better visualisation.

3.1 RASTER-SCAN GRAPHICS

It has been discussed earlier that raster-scan displays are the most widely used monitors and, therefore, the graphic software has to provide the necessary components of that as part of the software. This involves the conversion of the vectorial information of the drawing into its equivalent raster format such that the frame buffer can be filled with that information. This process is termed as *rasterisation* and involves one of the most important basic components of a graphic software suit. The two most common forms of geometric elements present in a graphic display are straight lines and circles. The other geometric elements can be converted into either of these forms. Hence, the algorithms are developed for these elements only.

Converting a line vector into its equivalent pixel positions is an arduous task involving a large amount of computation. Each drawing consists of a large number of vectors to be displayed. Hence, there is need to have a simplified method by which these could be done at a faster rate with little computing overhead.

3.1.1 DDA Algorithm

DDA or Digital Differential Analyser is one of the first algorithms developed for rasterising the vectorial information. The equation of a straight line is given by

$$Y = m\,X + c \tag{3.1}$$

Using this equation for direct computing of the pixel positions involves a large amount of computational effort. Hence, it is necessary to simplify the procedure of calculating the individual pixel positions by a simple algorithm.

For this purpose, consider drawing a line on the screen as shown in Fig. 3.1, from (x_1, y_1) to (x_2, y_2). Then,

$$m = \frac{y_2 - y_1}{x_1 - x_2} \tag{3.2}$$

and

$$c = y_1 - m\,X_1 \tag{3.3}$$

The line-drawing method would have to make use of the above three equations in order to develop a suitable algorithm.

Equation 3.1, for small increments can also be written as

$$\Delta Y = m\Delta X \tag{3.4}$$

By taking a small step for ΔX, ΔY can be computed using Eq. 3.4. However, the computations become unnecessarily long for arbitrary values of ΔX. Let us now work out a procedure to simplify the calculation method.

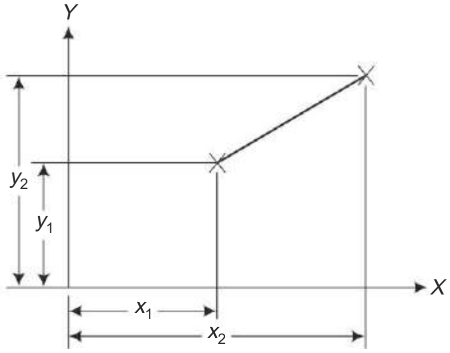


Fig. 3.1 *A straight-line drawing*

Let us consider a case of line drawing where $m > 1$. Choose an increment for ΔX as unit pixel.

Hence $\Delta X = 1$

Then from Eq. 3.4,

$$y_{i+1} = y_i + m \tag{3.5}$$

The subscript i takes the values starting from 1 for the start point till the end point is reached. Hence, it is possible to calculate the total pixel positions for completely drawing the line on the display screen. This is called the DDA (Digital Differential Analyser) algorithm.

If $m \le 1$ then the roles of x and y would have to be reversed.

Choose $\Delta Y = 1$ (3.6)

Then from Eq. 3.4, we get

$$x_{i+1} = x_i + \frac{1}{m} \tag{3.7}$$

The following is the flowchart showing the complete procedure for the implementation of the above procedure.

Example 3.1 Vectorise a line to be drawn from (10, 20) to (150, 125) mm on a display, which is mapped to approximately (300 × 250 mm). The resolution of the screen is 640 × 480 pixels.

Solution The mapping from user coordinates of (150, 125) to the device coordinates is to be first calculated as follows:

$$\text{Scale factor for } X \text{ direction: } \frac{640}{300} = 2.1333 \text{ pixels/mm}$$

$$\text{Scale factor for } Y \text{ direction: } \frac{480}{250} = 1.92 \text{ pixels/mm}$$

From these two, choose the one which does not violate the requirement, i.e., the smaller value of 1.92 pixels/mm as the scale factor for conversion from user coordinates to device coordinates.

Now using this scale factor, we can convert the start and end points into the pixel equivalents.

For start point: (10 × 1.92, 20 × 1.92) = (19, 38) pixels

Similarly, for end point: (150 × 1.92, 125 × 1.92) = (288, 240) pixels

From the above, $\Delta X = (288 - 19) = 269$ pixels

From the above, $\Delta Y = (240 - 38) = 202$ pixels

$$\text{Hence the slope of the line, } m, \frac{\Delta Y}{\Delta X} = \frac{202}{269} = 0.751$$

From the above values now we can compute the necessary pixel values based on the above algorithm as shown in the following table:

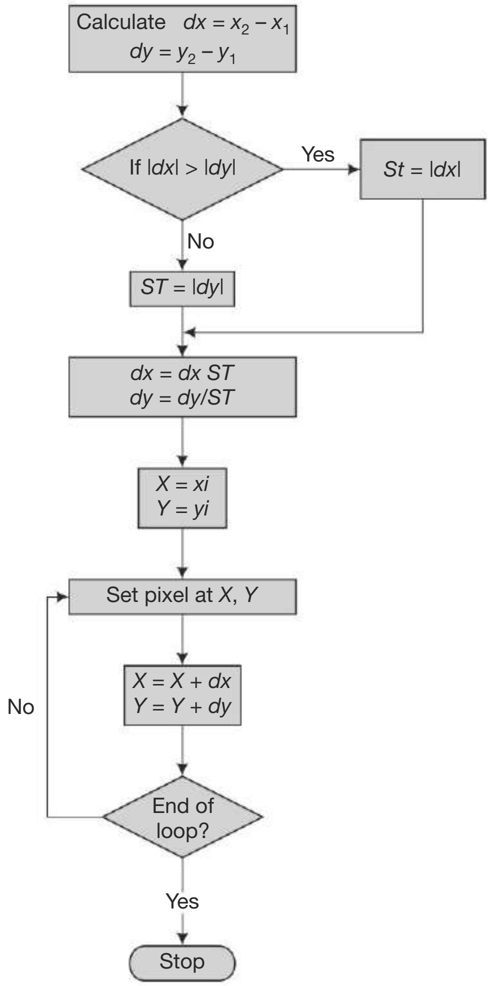


Fig. 3.2 *Flow chart for line-drawing calculation procedure*

Table 3.1

X	*Y calculated*	*Y rounded*
19	38	38
20	38.751	39
21	39.502	40
22	40.253	40
23	41.004	41
24	41.755	42
25	42.506	43
26	43.257	43

Contd..

Contd..

27	44.008	44
28	44.759	45
29	45.51	46
30	46.261	46
31	47.012	47
32	47.763	48
33	48.514	49
34	49.265	49
35	50.016	50
36	50.767	51
37	51.518	52
38	52.269	52

3.1.2 Bresenham's Algorithm

The above DDA algorithm is certainly an improvement over the direct use of the line equations since it eliminates many of the complicated calculations. However, still it requires some amount of floating-point arithmetic for each of the pixel positions. This is still more expensive in terms of the total computation time since a large number of points need to be calculated for each of the line segments as shown in the above example.

Bresenham's method is an improvement over DDA since it completely eliminates the floating-point arithmetic except for the initial computations. All other computations are fully integer arithmetic and thus is more efficient for raster conversion.

As for the DDA algorithm, start from the same line equation and the same parameters. The basic argument for positioning the pixel here is the amount of deviation the calculated position is from the actual position obtained by the line equation in terms of d_1 and d_2 shown in Fig. 3.3.

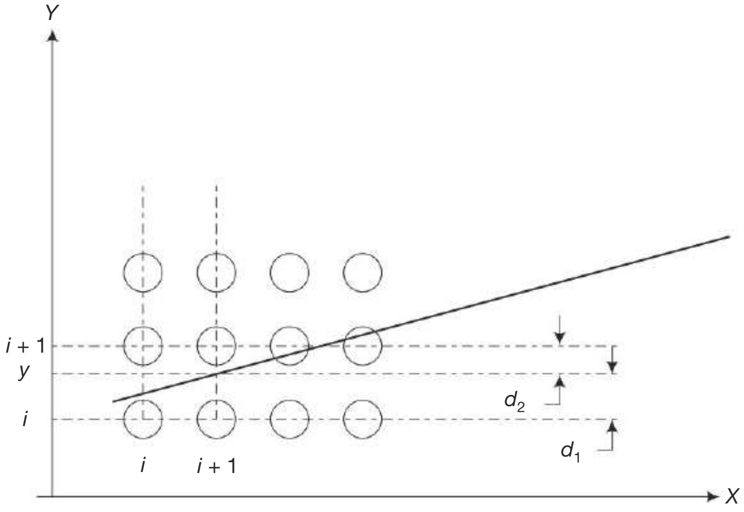


Fig. 3.3 *Line drawing using Bresenham algorithm*

Let the current position be (X_i, Y_i) at the ith position as shown in Fig. 3.3. Each of the circles in Fig. 3.3 represents the pixels in the successive positions.

Then $$X_{i+1} = X_i + 1 \tag{3.8}$$

Also $$Y_i = m\, X_i + c \tag{3.9}$$

and $$Y = m\,(X_i + 1) + c \tag{3.10}$$

These are shown in Fig. 3.3. Let d_1 and d_2 be two parameters, which indicate where the next pixel is to be located. If d_1 is greater than d_2 then the y pixel is to be located at the +1 position, else it remains at the same position as in the previous location.

$$d_1 = Y - Y_i = m\,(X_i + 1) + c - Y_i \tag{3.11}$$

$$d_2 = Y_{i+1} - Y = Y_{i+1} - m\,(X_i + 1) - c \tag{3.12}$$

$$d_1 - d_2 = 2m\,(X_i + 1) - 2Y_i + 2c - 1 \tag{3.13}$$

Equation 3.13 still contains more computations and hence we would now define another parameter P which would define the relative position in terms of $d_1 - d_2$.

$$P_i = (d_1 - d_2)\,\Delta X \tag{3.14}$$

$$P_i = 2\,\Delta Y\, X_i - 2\,\Delta X\, Y_i + b \tag{3.15}$$

where $$b = 2\,\Delta Y + 2c\,\Delta X - \Delta X$$

Similarly, we can write

$$P_{i+1} = 2\,\Delta Y\,(X_i + 1) - 2\,\Delta X\, Y_{i+1} + b \tag{3.16}$$

Taking the difference of two successive parameters, we can eliminate the constant terms from Eq. 3.16.

$$P_{i+1} - P_i = 2\,\Delta Y - 2\Delta X\,(Y_{i+1} - Y_i) \tag{3.17}$$

The same can also be written as

$$P_{i+1} = P_i + 2\,\Delta Y - 2\,\Delta X\,(Y_{i+1} - Y_i) \tag{3.18}$$

The same can also be written as

$$P_{i+1} = P_i + 2\,\Delta Y \quad \text{when } Y_{i+1} = Y_i \tag{3.19}$$

$$P_{i+1} = P_i + 2\,\Delta Y - 2\,\Delta X \quad \text{when } Y_{i+1} = Y_i + 1 \tag{3.20}$$

From the start point (x_1, y_1)

$$y_1 = m\, x_1 + c$$

$$c = y_1 - \frac{\Delta Y}{\Delta X} x_1$$

Substituting this in Eq. 3.15 and simplifying, we get

$$P_1 = 2\,\Delta Y - \Delta X \tag{3.21}$$

Hence, from Eq. 3.17, when P_i is negative then the next pixel location remains the same as the previous one and Eq. 3.19 becomes valid. Otherwise, Eq. 3.20 becomes valid.

Using these equations, it is now possible to develop the algorithm for the line drawing on the screen as shown in the following flowchart. The procedure just described is for the case when $m > 1$. The procedure can be repeated for the case of $m < 1$ by interchanging X and Y similar to the DDA algorithm.

Example 3.2 Repeat the calculations for the above example using Bresenham's algorithm.

Solution

X	P	Y
19	135	38
20	1	39
21	–133	40
22	271	40
23	137	41
24	3	42
25	407	43
26	273	43
27	139	44
28	5	45
29	–129	46
30	275	46
31	141	47
32	7	48
33	–127	49
34	277	49
35	143	50
36	9	51
37	–125	52
38	279	52

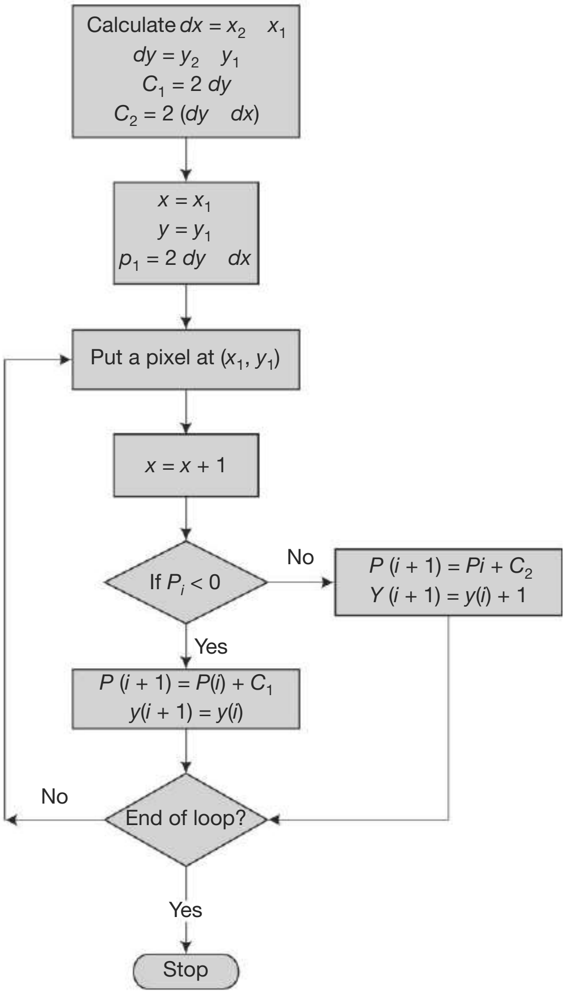


Fig. 3.4 *Flowchart for line-drawing calculation using the Bresenham procedure*

3.1.3 Anti-Aliasing Lines

The rasterisation algorithms discussed earlier will be generating the pixel points by rounding off to the nearest integer. As a result, the inclined lines have the jagged effect often called the staircase effect as shown in Fig. 3.5. The effect will be more pronounced in the case of the lines with small angles as shown in Fig. 3.5. It is possible to improve the appearance by increasing the screen resolution as shown in Fig. 3.6.

The effect can be decreased by anti-aliasing based on the sampling theory. Each of the geometric elements has a certain thickness compared to the size of the pixel. As can be seen in Fig. 3.7, the finite line thickness is overlapping the pixel with different areas. In this method, the intensity of the pixel is made proportional to the area of the pixel covered by the line thickness. Though this improves the appearance of the line, it is computationally more intensive.

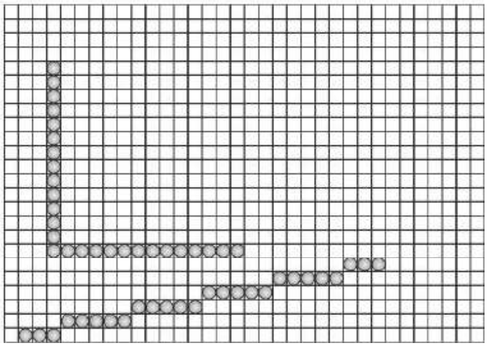


Fig. 3.5 *The staircase effect of pixels when drawing inclined lines*

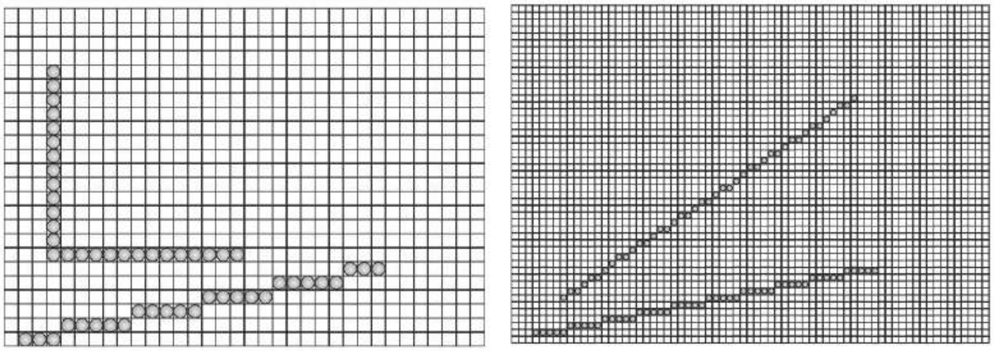


Fig. 3.6 *The staircase effect of pixels when drawing inclined lines decreases with increased resolution*

Raster display of lines utilises unequal number of pixels to represent lines depending upon their orientation in space. For example, from Fig. 3.8, it can be seen that the same number of pixels are representing a small length when it is horizontal or vertical while the length of an inclined line is more for the same number of pixels. This makes the horizontal or vertical lines more bright compared to the inclined lines. This can also be taken care of by making the brightness of the pixels different depending upon the inclination of the line. For example, the pixels on lines at 45° will be made with the highest brightness while the horizontal or vertical lines will be drawn with approximately 50% of the brightness.

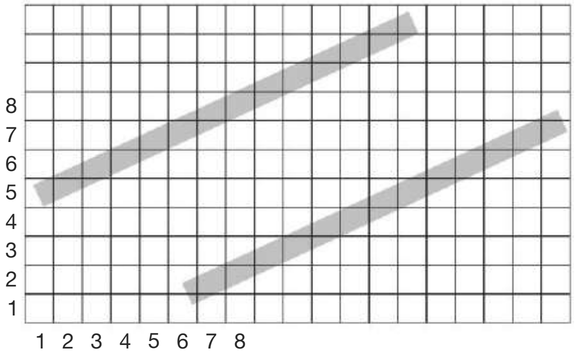


Fig. 3.7 *Anti-aliasing of pixels proportional to the portion of pixel occupied by the line*

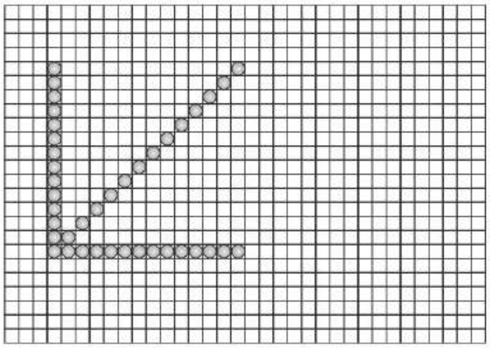


Fig. 3.8 *Unequal number of lines displayed with the same number of pixels*

3.2 COORDINATE SYSTEMS

The right-handed Cartesian coordinate system is used for defining the geometry of the parts. In order to specify the geometry of a given solid, it is necessary to use a variety of coordinate systems. They are the following:

World Coordinate System This refers to the actual coordinate system used as a master for the component. Sometimes, it may also be called the model coordinate system. In this book, we will call it the World Coordinate System, or WCS. Figure 3.9 shows a typical component, which needs to be modelled. Figure 3.10 shows the component with its associated world coordinate system, X, Y and Z. This is basically the coordinate

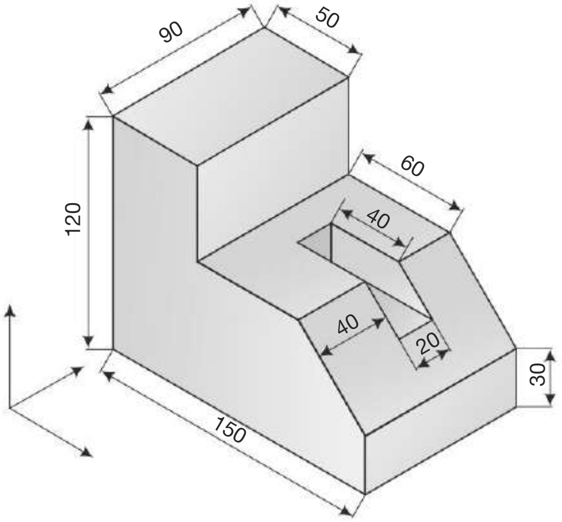


Fig. 3.9 *A typical component to be modelled*

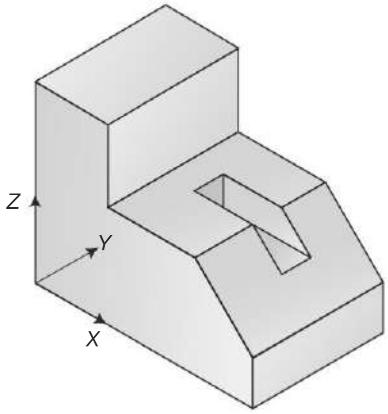


Fig. 3.10 *A typical component with its associated WCS*

system in which the part database is stored. However, the user will also have the flexibility of inputting the data in other coordinate systems as well such as polar coordinates or spherical coordinates. The software before it stores the data will actually convert this information into the Cartesian system.

User Coordinate System The default coordinate system when the user starts the modelling is the WCS. However, sometimes it becomes difficult to define certain geometries if they are to be defined from the WCS. In such cases, alternate coordinate systems can be defined relative to the WCS. These coordinate systems are termed User Coordinate Systems (UCS) or working coordinate systems. For example, in Fig. 3.11, X', Y', Z' is the user coordinate system defined for modelling the slot. Similarly, there can be other UCS that could be defined depending upon the geometry. This reduces the modelling complexity. The UCS can be defined by shifting the origin only as shown in Fig. 3.11 or by combining the origin with the orientation of the axes as well.

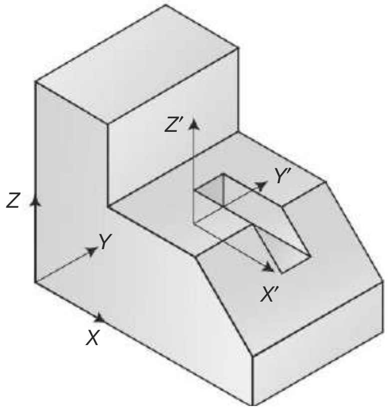


Fig. 3.11 *A typical component with its associated WCS and UCS*

Display Coordinates This refers to the actual coordinates to be used for displaying the image on the screen. It may also be termed the screen coordinate system. The actual screen coordinates relate to the pixels to determine whether the actual values of the screen or the virtual image that can be displayed are needed to help in the image display. The virtual size will be larger than the actual pixels of the screen resolution.

View Generation The display screen is two-dimensional. Sometimes, it is necessary to organise the information when presented on the screen in two dimensions using the orthogonal projection. The screen is therefore divided into a number of view ports wherein the various views are presented. For example, the most common views required for representing fully the component details are the front, top and right-side views as shown in Fig. 3.12. The views generated in the process along with their coordinate systems as referred to the WCS are shown in Fig. 3.13.

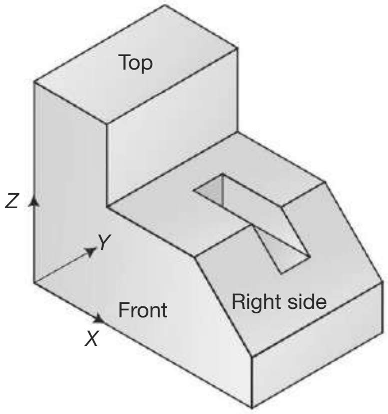


Fig. 3.12 *A typical component with its various view positions*

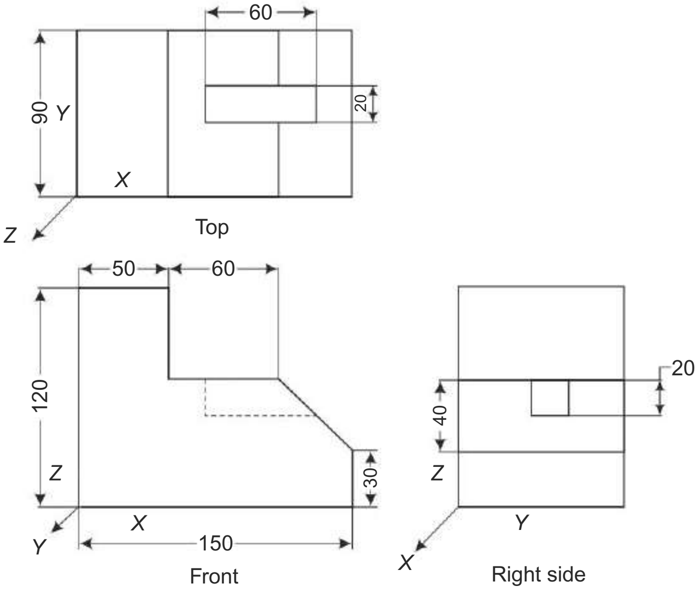


Fig. 3.13 *Various views generated from the model shown in Fig. 3.12*

3.3 DATABASE STRUCTURES FOR GRAPHIC MODELLING

The database of the graphical representation of the model is present in the computer system in a form convenient for use. The major functions of a database are to manipulate the data on screen, such as zooming and panning; to interact with the user, essentially for the purpose of editing functions like trimming, filleting, stretching, etc.; to evaluate the properties like areas, volumes, inertias, etc.; and to provide additional information like manufacturing specifications.

The complexity of the database depends to a very great extent, on the type of modelling and information-retrieving capabilities built into it. Graphic databases may contain graphical information such as point coordinates, alphanumerical information as manufacturing requirements, or some procedural type wherein the concerned data is to be fitted in a certain form, for example, like that of a parametric macro. Most of the information contained in the database is interdependent and often makes use of labels or pointers for the purpose of accessing data in the various interconnected files. In view of this the general-purpose database modelling systems are not used, but special procedures are to be developed for the particular modelling system under development.

Typical data that would normally be contained in a geometric model file is

1. *Organisational data*
    Identification number
    Drawing number
    Design origin and status of changes
    Current status
    Designer name

Date of design
Scale
Type of projections
Company

2. *Technological data*

Geometry
Dimensions
Tolerances
Surface finishes
Material specifications or reference
Manufacturing procedures
Inspection procedures

There is a lot of data pertaining to the geometric model and it is to be read and manipulated during the process of modelling and drawing generation. One may have to fetch the data a number of times from the same database, so it becomes necessary to devise a database structure, which is simple and also easy to amend. The data structure that is envisaged will have to fulfil a number of functions to be most effective for the given application. In the case of geometric modelling, the database structure should provide some of the following functionalities:

- Allow for greater interaction of the user with the modelling system. This should allow the user to add, delete and modify the data in the form of geometric entities as required during the process of modelling and other associated functions.
- Support a large variety of types of data to be represented in the database as explained earlier. The types of data may be in the form of real and integer values, geometric information, textual information such as manufacturing notes or finishes, CNC tool paths, finite element meshes, etc.
- The data structure should be able to allow for associating the data with the modelling process such that the design intent can be effectively captured by the data and can be utilised later for any of the modifications as required during the lifecycle of the product.
- The data should allow for storing information in the most compact form possible such that less data storage and transmission speeds would be required for working in group environments. For example, multiple instances of geometry may only be stored once and then simply the instant information be used for manipulating that to generate the models or drawing as required.
- Data should maintain complete associativity with all the downstream and upstream applications such that modification done at any stage should be able to percolate through all the databases as required.

The above requirements are fairly indicative of the complexities that are associated with the design of data structures for the geometric modelling systems. Unfortunately, the geometric elements involved in the model have different geometric requirements, and the data model will have to accommodate all such variations. For example, a line can be simply represented by means of two vertices (points) in terms of the x, y, and z coordinate values. On the other hand, a spline curve may have to be specified by means of a number of vertices and other parameters depending upon the number of knots utilised. So that variation will have to be efficiently handled by the data structure adopted.

3.3.1 Data Structure Organisation

A database can be defined as a collection of data which is shared throughout a given process for multiple applications. In our particular case, the given process is the modelling and associated functions such as analysis, manufacturing, planning, etc. The main purpose of a database is the ability for sharing the data by a number of processes and users. Before we proceed with developing the database-structure concepts, it is necessary to understand some of the terminologies that are used with databases.

Data Record Data record consists of a series of facts or statements that may have been collected, stored, processed and/or manipulated together to represent the information about a particular item.

Data File A file is a collection of data items put together. For example, all the necessary information required to completely draw a part in two dimensions can be organised into a file such as a drawing file. The information present in a data file normally has some structured relationship to decode it properly. On the other hand, flat files are data files that contain records with no structured relationships. Additional knowledge is required to interpret these files such as the file-format properties.

Data Field A field is a single unit of data stored as part of a database record. Each record is made up of one or more fields that correspond to the columns in a database table.

Data Model Data model is concerned with what the data in the database is represented. The data model is used by the database designers as the basis for their designing work. It generally uses three types of relationships: one-to-many, many-to-many, and one-to-one.

A lot of information needs to be stored in the system for the purpose of generating any meaningful information about a product. For example, the product may consist of a number of sub-assemblies, and each of the sub-assembly consists of a number of parts, some of which are standard components such as fasteners, while others need to be completely defined. Each of the parts will have individual characteristics such as material, form, shape, treatment, structure, etc., all of which needs to be stored inside the database for the product. This is only when we consider a simple part-modelling application. But if the same structure has to be made to use all downstream applications such as drafting, analysis, manufacturing planning and manufacturing information generation then it is necessary to provide all the proper information and handles will have to be taken care of.

3.3.2 Data Models

The more common data models used in database management systems are

- the hierarchical model,
- the network model, and
- the relational model.

Hierarchical Model The data in this model is arranged in the logical hierarchy with each data element containing a pointer to only one element in the level above it. This is often represented as a tree structure in a manner similar to a bill of material (Fig. 3.14), and a complete database may consist of a number of separate structures. Though this was one of the first type of database models implemented, its use is coming down in recent times. The main reason for its declining popularity is that it is rare to find a complete set of data that is purely hierarchical, even though some parts may be. This method can express the one-to-many relationship easily while the many-to-many relationship is extremely difficult to implement. Therefore, imposing a hierarchical structure on the database calls for unnecessary coding complications and could be avoided.

Network Model The network model does not use the tree structure as shown in Fig. 3.15 and hence avoids the problems as faced by the hierarchical data model. In order to describe associations between the records in a database, the network model utilises a ‘connection record’ that links records together to form data sets.

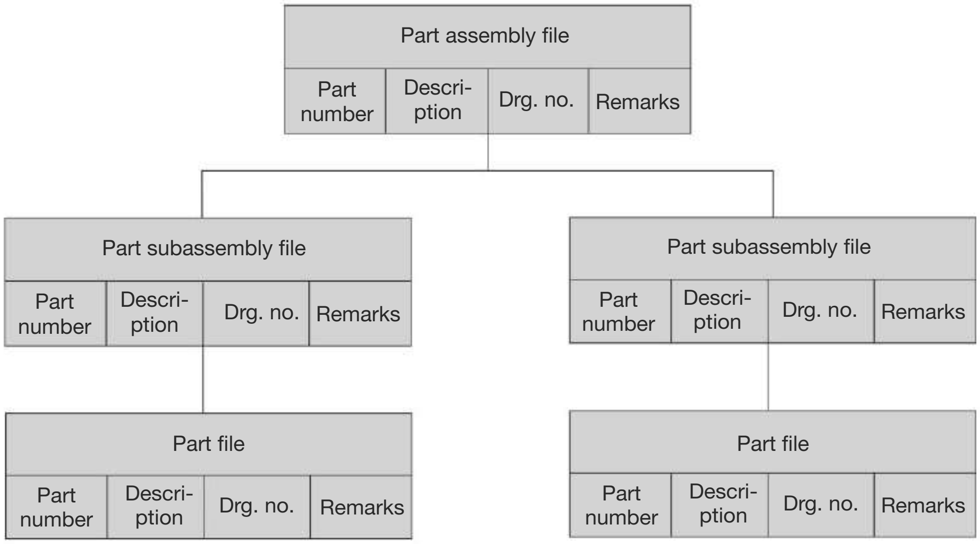


Fig. 3.14 *A hierarchical representation of data for a product model*

Unlike a hierarchical model, the network model does not allow a record type to be both the owner and member of a set. As a result, modelling a hierarchy requires a level of indirection to be introduced through programming, which should not be too difficult to achieve. The network model allows the user to store and retrieve many-to-many relationships in an efficient manner and is more flexible. However, it requires a more complex data dictionary or schema together with detailed data-access procedures. This has a better performance compared to the hierarchical model and is generally used in finance and business institutions and rarely for engineering applications.

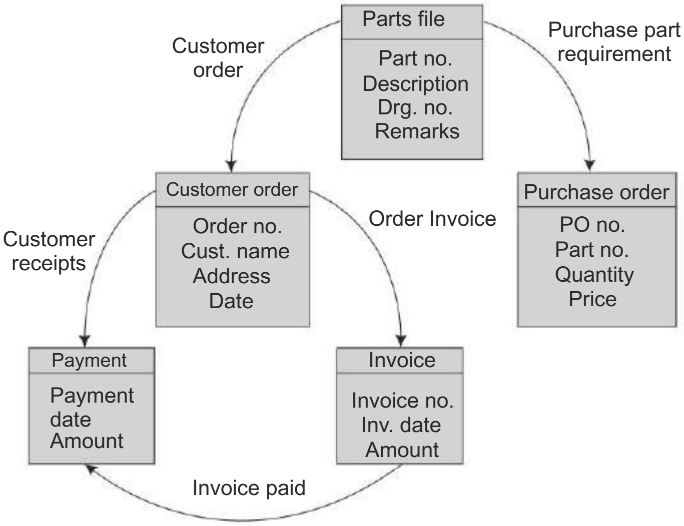


Fig. 3.15 *A network representation of data for a product model*

Relational Model This is more recent compared to the other models, but widely used in engineering applications. In this model, the data is represented in the form of tables each of which is then linked by means of relations. A table is a collection of records. In a database system, the records are considered as rows in a table, and the fields are the columns (Fig. 3.16). In academic terminology, the rows are referred to as *tuples,* and columns within a record as *attributes*.

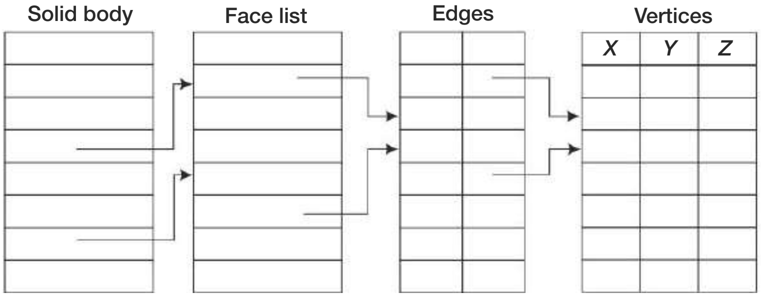


Fig. 3.16 *A relational representation of data for a product model*

In a relational database there are no predefined connections between the data tables. The fields that need to be connected are simply duplicated in all the data tables and the actual connections are done dynamically. The dynamic joining capability means that the structure of any individual table can be modified without affecting the other data tables. This aspect is very important and is useful for developing the applications in modular form.

Structured Query Language (SQL) developed by IBM and standardised by ANSI (American National Standards Institute) is used for the data manipulation in a relational data model. It utilises English-like words such as SELECT, FROM, and WHERE to develop the necessary relations in the database.

Relational Database Management Systems (RDBMS) are widely used in the industry for a variety of applications due to their simplicity and flexibility. The greatest disadvantage of RDBMS is that it is slower compared to the other models described above. However, because of the developments in computer hardware and software developments in the last decade, this limitation is not of much consequence.

3.3.3 Geometric Model Data

One of the possible ways in which graphic data can be stored is in the sequential form. In sequential form, the data as is being generated is stored, and the disadvantage is that whenever one has to access certain data, the retrieval is not simple. The problem is further complicated by the fact that one may not be retrieving the data in sequential form, and as a result, the sequential form of data storage proves inefficient from the graphic manipulation point of view. It is, therefore, necessary in most of the geometric modellers to opt for a combination of random and sequential form so as to get the best out of both the forms.

Hence, records that are contained in a graphical database are maintained as random access files and all the files present are linked by means of pointers. The main record may be termed 'head record' to and from which a number of pointers link the entire data required for the component in a coherent form. Another important advantage that can be derived from this form is that there is a possibility of reducing the data storage by making use of some form of referencing where the same data required at a number of places is stored only once, somewhat similar to subroutines or procedures in the programming languages. In view of the slow access times of the secondary storage devices (floppies and hard disks), some portion of the active drawing database is stored in the RAM itself by a suitable arrangement of paging of the data. This increases the response time of the system for modifications.

To get an idea of how the data for a component may be stored, refer to Fig. 3.17 showing a job modelled. The solid is first broken into edges, which are further broken into surfaces and the vertexes for completely defining the object. A face meets another face to form an edge. The edge meets the points (vertices) at the end. Faces may be multiple connected and one bound by one or more loops of edges as shown in Fig. 3.17. This is one method of representation of the solid data, which is called the *boundary representation* or *B-rep*. A boundary is formed by faces. A face is formed by a combination of edges. Edges may be considered as curves such as lines and arcs. Edges, in turn, are formed by a combination of vertices, with each vertex being indicated by means of its Cartesian coordinates *X*, *Y* and *Z*. Thus, a typical database structure of a product may consist of a number of component parts as shown in Fig. 3.18.

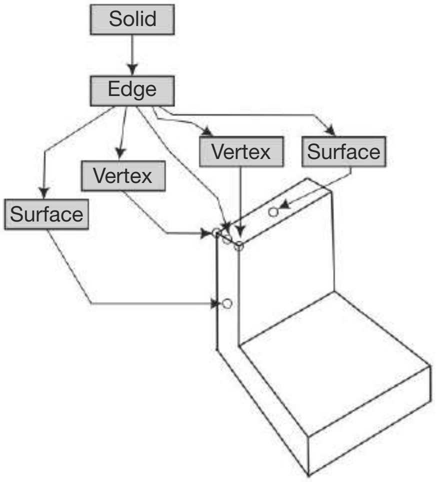


Fig. 3.17 *Data structure for geometric models*

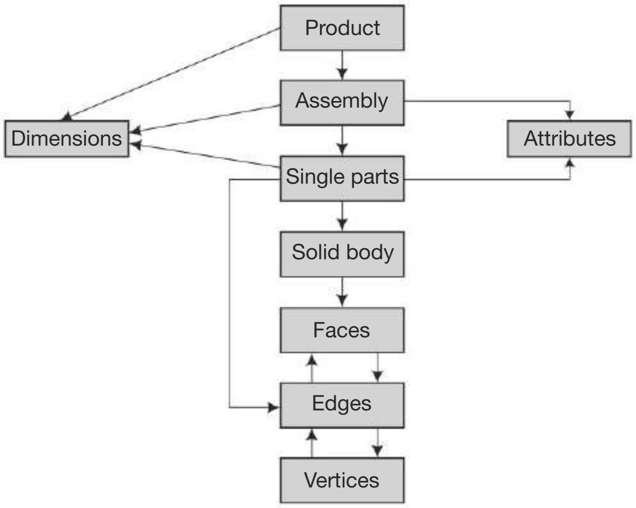


Fig. 3.18 *Complete data structure for geometric models of products*

The data when organised in a database will have to ensure the basic integrity of the data in terms of eliminating redundancy and security problems while maintaining the standards and ease of use. The most common form in which the graphical database is organised is in terms of the number of tables that are interlinked by relations as described earlier.

3.4 ENGINEERING DATA MANAGEMENT (EDM) SYSTEM

It is common knowledge that the products that are now available in the market have much more functionality and performance compared to the earlier models. For example, if we look at the music systems or TVs of the current generation then one can notice a plethora of facilities and features that were not present in the earlier generations. This increased functionality and performance puts tremendous amount of pressure on the designers to produce parts at a faster rate to beat the competition. This is further compounded by the fact that the need to adhere to greater amount of standards, problems associated with the product liability litigation, increasing awareness of quality issues, documentation, etc. All these contribute to a large amount of data that needs to be managed as part of the product-development process. Thus, the number of documents that needs to be managed becomes immense. For example, a typical automobile may have more than 100 000 parts each with several related documents such as design, geometric model, drafting, CNC programs, FEM simulations, etc. It is noted that at any given time, a large number of those documents may not be current depending upon the updating procedures adopted.

The engineering data traditionally is present in various forms such as papers, drawings, charts, etc., along with computer storage media. The variety of documentation methods, the inefficient documentation, change and control methods would not be desirable for modern complex product-development methods. Thus, it is necessary to develop methods for efficient collection, control, dissemination and archival of all data relating to a product or service. Also, the use of computers in all walks of the product development and manufacturing operations has increased. Examples of such applications are computer aided design, drafting, FEM analysis, and CNC part-program generation, product simulations, computer aided testing, rapid prototyping, to name a few.

The management of such a large amount of data over the entire lifecycle of the product is called by a number of names with different understandings such as Product Data Management (PDM), Product Information management (PIM), Engineering Data Management (EDM), Technical Documentation Management (TDM), Engineering Management System (EMS), etc. To further confuse the issues, a number of other derivative processes have been identified such as Drawing Office Management Systems (DOMS), Engineering Records Management (ERM), and Engineering Data Management (EDM). These mean different things to different people depending upon the way they are defined and applied. Thus, the application of EDM has to be carefully thought out for a given application. EDM can be defined as the systematic planning, management and control

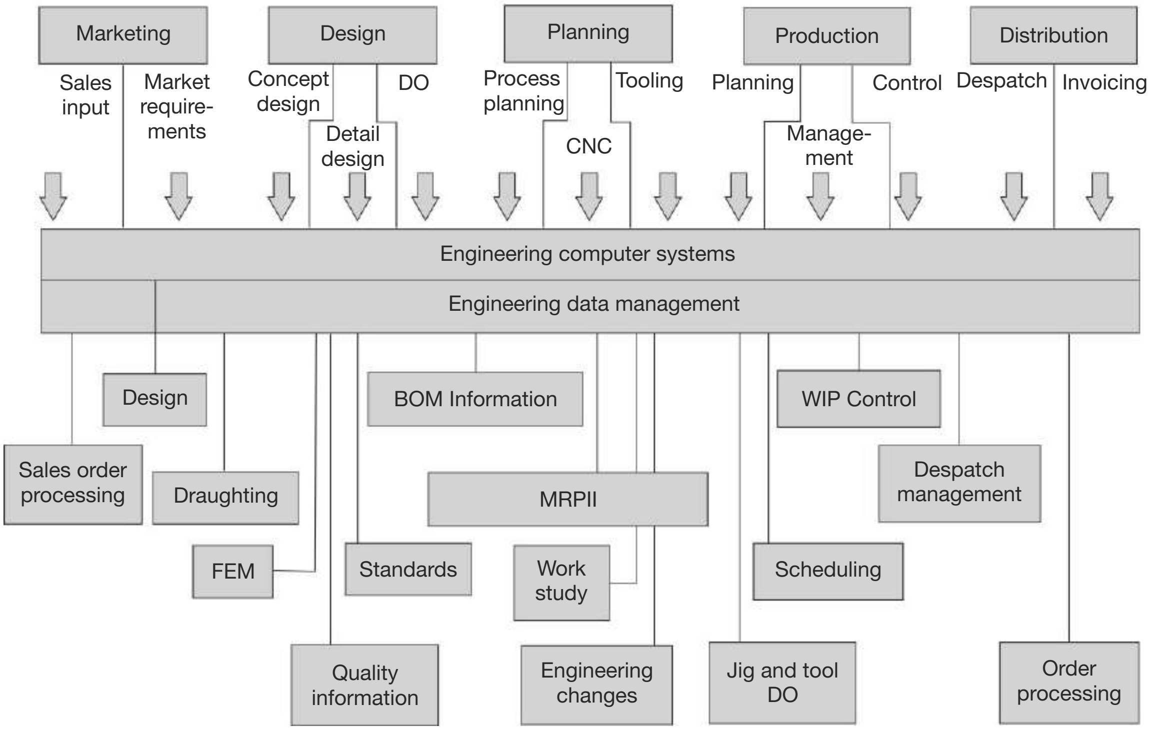


Fig. 3.19 *Engineering data management within a manufacturing enterprise*

of all the engineering data required to adequately document a product from its inception, development, test, and manufacture to its ultimate retirement from service. The process is shown diagrammatically in Fig. 3.19.

As can be seen from Fig. 3.19, not all the data associated with the product is 'engineering-related'. Some of it is related to business and finance such as sales orders, invoices, despatches, shipping, etc. Most of that data may be part of a separate DBMS and the EDM will have to coexist with that. Some of the important features that may be considered as part of an EDM application are the following:

- The EDM will be built upon a comprehensive Database Management System (DBMS) that takes care of storage of data and provides a consistent mechanism for the manipulation and control of data. The DBMS will have to provide extensive facilities for input, storage, updating, retrieval and reporting the data with appropriate facilities for defining the data formats, sizes and relationships. The total structure should take care of the top-level assembly, sub-assemblies as well as each of the parts. Each field or attribute can be customised to suit the particular application requirement, for example, part number structure and format, quantity of information, relevant project, revision level, etc. All the data from various other sources can be referenced here such as file references, CAD data, CAPP, CNC programs, etc. The logical arrangement of the data-reference structure is shown in Fig. 3.20.

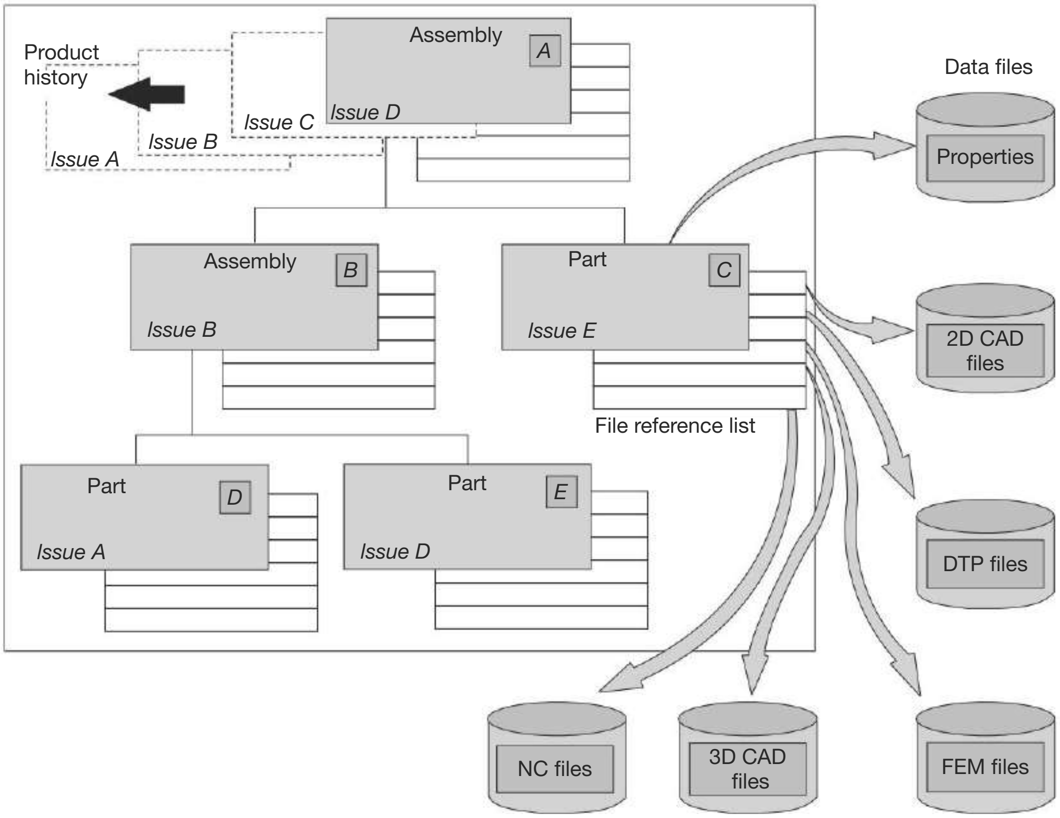


Fig. 3.20 *Engineering data-management references and their associations*

- The data structure employed should have the ability to maintain the product-structure relationship with the historical data for engineering change through the product structure such as the Bill Of Material (BOM) file. The BOM data should be able to link upstream to CAD systems as well as to the downstream MRP II BOM files. If possible, the same files could be utilised at all levels to reduce the errors and redundant data sources. All the historical information should be maintained so that it is possible to maintain traceability as per the standards such as ISO 9000.
- Appropriate access-control procedures will have to be established since a large number of departments will be accessing and controlling the data in the EDM. The access control can be established depending upon the user login. The controls can be No Access, Read Only, Read and Write or Read, Write and Delete. These access controls will depend upon the actual module that is being accessed.
- The EDM system should allow for an efficient search facility whereby the engineers will spend less time searching and accessing the data and spend more useful time in actual engineering tasks. It is generally considered that engineers spend from 15 to 40% of their time in information retrieval. Most of the engineering-design function depends on the past design experiences and as such, the engineers should be able to easily retrieve similar designs, products or sub-assemblies. That way, it may be possible that past designs could be easily recycled, thereby reducing a lot of product-development time.
- It should provide for a method to manage the change control process such that the right information is available at the right place at the right time, thereby controlling the final product costs. Design teams generate the designs based on the various inputs that have been received for the purpose. However, the interpretation at each stage may mean that the final result may not be what was actually wanted. So the need for engineering changes will increase as the design comes towards the final stages. This is more costly and should be avoided. However, it is a reality that there will be a number of changes in the design. The EDM system should be capable of raising and tracking the Engineering Change Orders (ECR) by raising and circulating the Engineering Change Notes (ECN). The engineering change notices should be interfaced effectively with the MRPII system to keep track of the changes and leaving a proper audit trail.

  The ECR should have all the necessary information to explain the details of the change, the associated reasons and likely ramifications. The ECR will be circulated to all the relevant personnel for their comments and approval or rejection electronically. Once the ECR is approved, it will generate the ECN indicating the work involved in the change and its likely effective date or revision. By following a systematic engineering change process, the number of unauthorised and ad-hoc changes to a product can be controlled. This control can then be used to improve the development process and the subsequent cost of the entire process. An example of a detailed change control process is shown in Fig. 3.21.
- The EDM system should be able to provide full traceability over the entire product structure to enable information about the individual parts to be retrieved at any given time.
- The EDM system should have the ability to generate the necessary warnings and messages. These can be used to determine and track the system’s current status.
- Since they handle a large amount of data, they should have the ability to maintain the integrity of the database, avoid data corruption and provide the ability to easily archive and restore the sets of data.

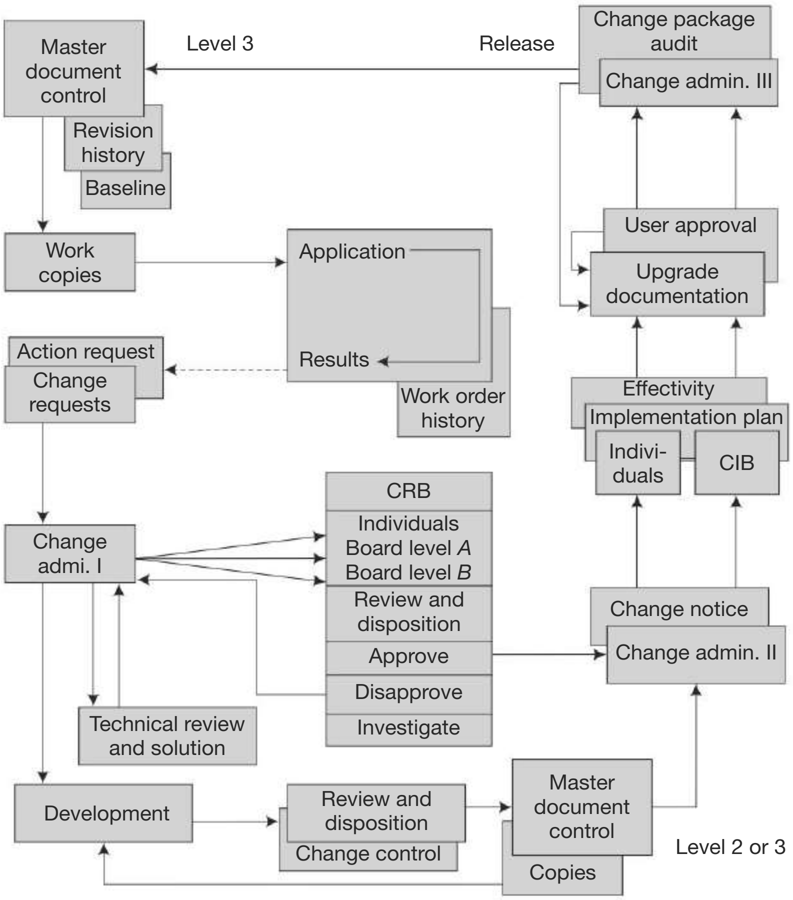


Fig. 3.21 *Engineering data-management references and their associations*

3.5 TRANSFORMATION OF GEOMETRY

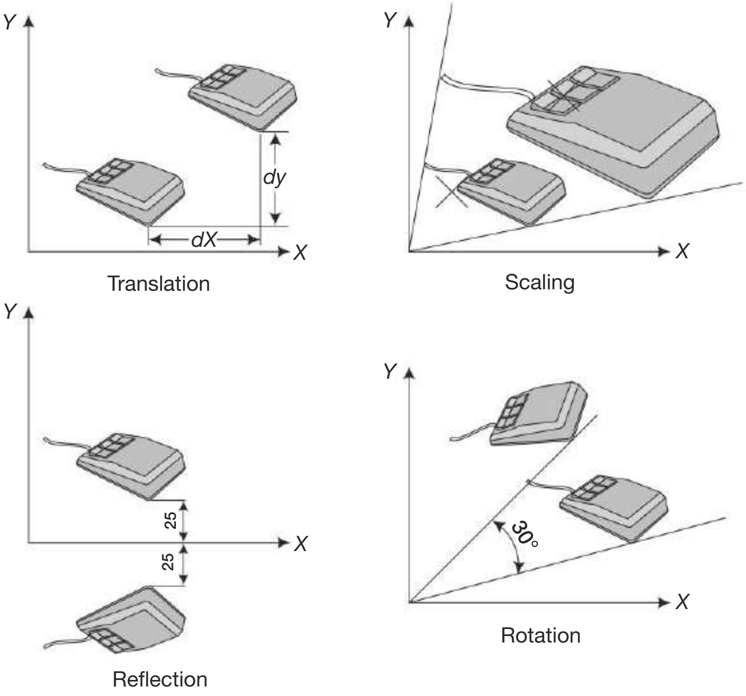
The geometry traditionally followed is the Euclidean geometry. In the traditional sense, we follow the Cartesian coordinate system specified by the X, Y and Z coordinate directions. The three axes are mutually perpendicular and follow the right-hand system.

In the handling of geometrical information, many a times it becomes necessary to transform the geometry. The transformations actually convert the geometry from one coordinate system to another.

The main types of pure transformations with which we are likely to come across are

- Translation
- Scaling
- Reflection or mirror
- Rotation

These transformations are symbolically shown in Fig. 3.22.



Translation

Scaling

Reflection

Rotation

Fig. 3.22 *Some of the possible geometric transformations*

A point in space can be represented by its coordinates (x, y, z) from the datum. As shown in Fig. 3.22 a point in three dimensions can be represented by the coordinates (x, y, z). The same can also be represented by a vector starting from the origin of the coordinate system as shown in Fig. 3.23.

$$P = [x, y, z] \tag{3.22}$$

$$[P] = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.23}$$

In order to understand the system easily, we can look at the transformations in the two-dimensional system for the sake of easy comprehension. The same would then be extended to look at the three-dimensional viewing.

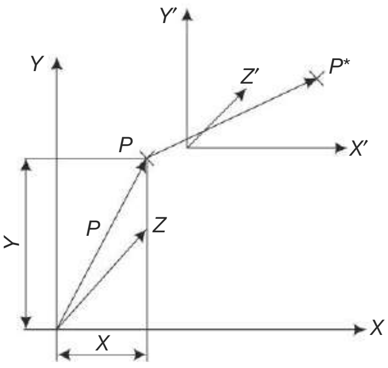


Fig. 3.23 *Translation of the point*

3.5.1 Translation

It is the most common and easily understood transformation in CAD. This moves a geometric entity in space in such a way that the new entity is parallel at all points to the old entity. A representation is shown in Fig. 3.24 for an object. Let us now consider a point on the object, represented by P which is translated along X and Y axes by dX and dY to a new position P^*. The new coordinates after transformation are given by

$$P^* = [x^*, y^*] \tag{3.24}$$

$$x^* = x + dX \tag{3.25}$$

$$y^* = y + dY \tag{3.26}$$

Putting Eqs 3.25 and 3.26 back into Eq. 3.24, we can write

$$[P^*] = \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} x + dX \\ y + dY \end{bmatrix} \tag{3.27}$$

This can also be written in the matrix form as follows:

$$[P^*] = \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} x + dX \\ y + dY \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dX \\ dY \end{bmatrix} \tag{3.28}$$

This is normally the operation used in the CAD systems as the MOVE command.

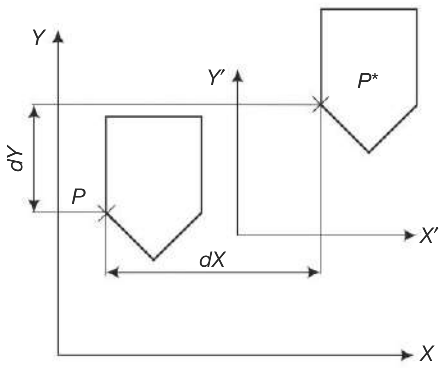


Fig. 3.24 *Translation of a group of points (plane figure)*

3.5.2 Scaling

Scaling is the transformation applied to change the scale of an entity. As shown in Fig. 3.25, this alters the size of the entity by the scaling factor applied. For example, in the figure, to achieve the scaling, the original coordinates are multiplied uniformly by the scaling factor.

$$P^* = [X^*, Y^*] = [S_x \times X, S_y \times dY] \tag{3.29}$$

This equation can also be represented in the matrix form as follows:

$$[P^*] = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.30}$$

$$[P^*] = [T_s] \,.\, [P] \tag{3.31}$$

where

$$[T_s] = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \tag{3.32}$$

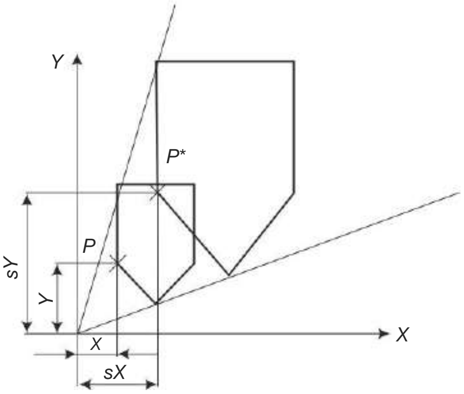


Fig. 3.25 *Scaling of a plane figure*

Since the scaling factors can be individually applied, there is a possibility to have differential scaling when $S_x \neq S_y$. Normally, in the CAD systems, uniform scaling is allowed for object manipulation. In the case of zoom facility in graphic systems, uniform scaling is applied. Zooming is just a display attribute and is only applied to the display and not to the actual geometric database. It can also be noted from Fig. 3.25 that the transformed figure has actually moved away as if a translation has taken place. This apparent translation is because the scaling is with respect to the origin of the coordinate system. If the centroid of the plane figure coincides with the base of the scaling factor then there will be no apparent translation. The student is advised to check this using the CAD software available in the laboratory.

3.5.3 Reflection or Mirror

Reflection or mirror is a transformation, which allows a copy of the object to be displayed while the object is reflected about a line or a plane. Typical examples are shown in Fig. 3.26, wherein (a) shows reflection about the *X*-axis, while the one in (b) is the reflection about the *Y*-axis. The reflection shown in (c) is about the *X* and *Y*-axis or about the origin.

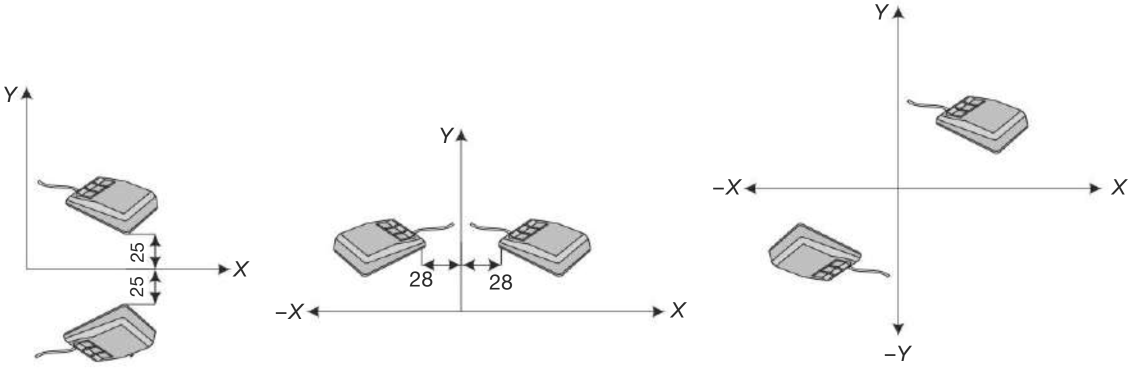


Fig. 3.26 *Possible reflection (mirror) transformations of geometry in 2D*

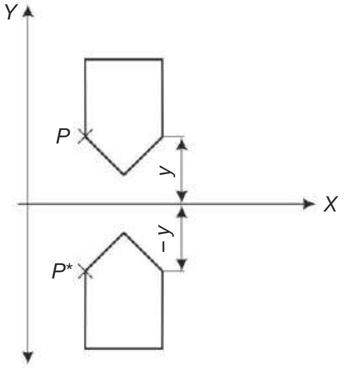


Fig. 3.27 *Example for reflection transformation*

The transformation required in this case is that the axes of coordinates will get negated depending upon the reflection required. For example, from Fig. 3.27, the new

$$P^* = [X^*, Y^*] = [X, -Y] \tag{3.33}$$

This can be given in a matrix form as

$$[P^*] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.34}$$

$$[P^*] = [T_m] . [P]$$

where

$$[T_m] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.35}$$

Thus, the general transformation matrix will be

$$[M] = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix} \tag{3.36}$$

Here, –1 in the first position refers to the reflection about the *Y*-axis where all the *X*-coordinate values get negated. When the second term becomes the reflection, it will be about the *X*-axis with all *Y*-coordinate values getting reversed. Both the values are –1 for reflection about *X* and *Y*-axes.

3.5.4 Rotation

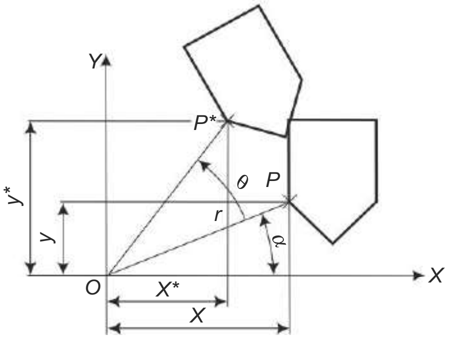


Fig. 3.28 *Rotation transformation*

This is another important geometric transformation. The final position and orientation of a geometric entity is decided by the angle of rotation and the base point about which the rotation θ as shown in Fig. 3.28, is to be done.

To develop the transformation matrix for transformation, consider a point *P*, located in the *XY* plane, being rotated in the counter-clockwise direction to the new position, *P** by an angle θ as shown in Fig. 3.28. The new position *P** is given by

$$P^* = [x^*, y^*] \tag{3.37}$$

From Fig. 3.24, the original position is specified by

$$\begin{aligned} x &= r\cos\alpha \\ y &= r\sin\alpha \end{aligned} \qquad (3.38)$$

The new position P^* is specified by

$$\begin{aligned} x^* &= r\cos(\alpha+\theta) \\ &= r\cos\theta\cos\alpha - r\sin\theta\sin\alpha \\ &= x\cos\theta - y\sin\theta \end{aligned} \qquad (3.39)$$

$$\begin{aligned} y^* &= r\sin(\alpha+\theta) \\ &= r\sin\theta\cos\alpha + r\cos\theta\sin\alpha \\ &= x\sin\theta + y\cos\theta \end{aligned} \qquad (3.40)$$

This can be given in the matrix form as

$$[P^*] = \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (3.41)$$

$$[P^*] = [T_R]\,.\,[P]$$

where $$[T_R] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad (3.42)$$

The above is the transformation matrix for rotation, which can be applied in any plane as follows:

$$\begin{bmatrix} y^* \\ z^* \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \qquad (3.43)$$

$$\begin{bmatrix} z^* \\ x^* \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \qquad (3.44)$$

Example 3.3 A square (Fig 3.29) with an edge length of 10 units is located in the origin with one of the edges at an angle of 30° with the +*X*-axis. Calculate the new position of the square if it is rotated about the *Z*-axis by an angle of 30° in the clockwise direction.

Solution The end points of the edges are

$$dx_1 = 10 \times \cos 30° = 8.66$$

$$dx_2 = 10 \times \cos 30° - 10 \times \sin 30° = 3.66$$

$$dx_3 = 10 \times \cos 30° - dx_2 = 5$$

$$dy_1 = 10 \times \sin 30° = 5$$

$$dy_2 = dy_1 + 10 \times \sin 60° = 13.66$$

$$dy_3 = 10 \times \cos 30° = 8.66$$

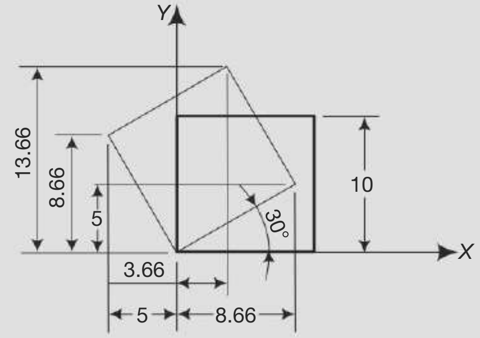


Fig. 3.29 *Example 3.3*

The transformation matrix is

$$[T_R] = \begin{bmatrix} \cos -30 & -\sin -30 \\ \sin -30 & \cos -30 \end{bmatrix} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}$$

The new coordinates are

$$\begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix} \begin{bmatrix} 0 & 0.866 & 3.66 & -5 \\ 0 & 5 & 13.66 & 8.66 \end{bmatrix} = \begin{bmatrix} 0 & 10 & 10 & 0 \\ 0 & 0 & 10 & 10 \end{bmatrix}$$

3.5.5 Concatenation of Transformations

Many a times it becomes necessary to combine the individual transformations as shown above in order to achieve the required results. In such cases, the combined transformation matrix can be obtained by multiplying the respective transformation matrices. However, care should be taken to see that the order of the matrix multiplication be done in the same as that of the transformations as follows:

$$[P^*] = [T_n]\,[T_n - 1]\,[T_n - 2]\,\ldots\,[T_3]\,[T_2]\,[T_1] \tag{3.45}$$

3.5.6 Homogeneous Representation

In order to concatenate the transformations as shown in Eq. 3.41, all the transformation matrices should be of multiplicative type. However, as seen earlier, the translation matrix (Eq. 3.28) is vector additive while all others are matrix multiplications. The following form could be used to convert the translation into a multiplication form:

$$[P^*] = \begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.46}$$

Hence, the translation matrix in multiplication form can be given as

$$[MT] = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix} \tag{3.47}$$

This is termed homogeneous representation. In homogeneous representation, an n-dimensional space is mapped into an $(n + 1)$-dimensional space. Thus a two-dimensional point $[x\ y]$ is represented by three dimensions as $[x\ y\ 1]$. This greatly facilitates computer graphics operations where the concatenation of multiple transformations can be easily carried out. This will be experienced in the following situations:

Rotation about an Arbitrary Point The transformation given earlier for rotation is about the origin of the axes system. It may sometimes be necessary to get the rotation about any arbitrary base point as shown in Fig. 3.30. To derive the necessary transformation matrix, the following complex procedure would be required.

1. Translate the point P to O, the origin of the axes system.
2. Rotate the object by the given angle.
3. Translate the point back to its original position.

The transformation matrices for the above operations in the given sequence are

$$[T_1] = \begin{bmatrix} 1 & 0 & -dX \\ 0 & 1 & -dY \\ 0 & 0 & 1 \end{bmatrix} \tag{3.48}$$

$$[T_2] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.49}$$

$$[T_3] = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix} \tag{3.50}$$

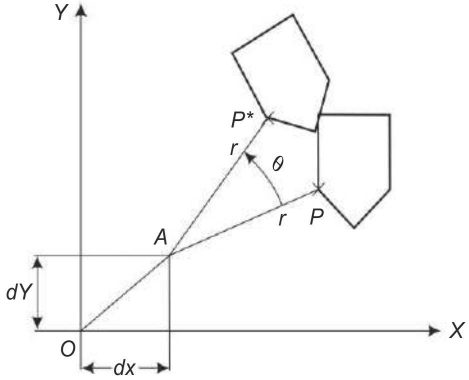


Fig. 3.30 *Rotation about an arbitrary point*

The required transformation matrix is given by

$$[T] = [T_3]\,[T_2]\,[T_1]$$

$$[T] = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -dX \\ 0 & 1 & -dY \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T] = \begin{bmatrix} \cos\theta & -\sin\theta & dX(1-\cos\theta) + dY\sin\theta \\ \sin\theta & \cos\theta & -dX\sin\theta + dY(1-\cos\theta) \\ 0 & 0 & 1 \end{bmatrix} \tag{3.51}$$

Reflection about an Arbitrary Line Similar to the above, there are times when the reflection is to be taken about an arbitrary line as shown in Fig. 3.31.

1. Translate the mirror line along the Y-axis such that the line passes through the origin, O.
2. Rotate the mirror line such that it coincides with the X-axis.
3. Mirror the object through the X-axis.
4. Rotate the mirror line back to the original angle with the X-axis.
5. Translate the mirror line along the Y-axis back to the original position.

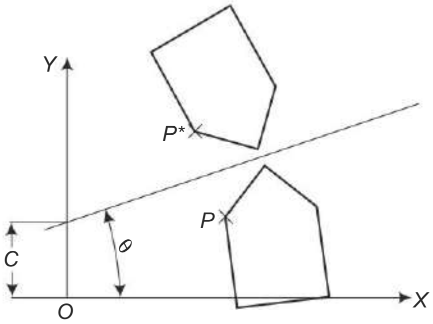


Fig. 3.31 *Example for reflection transformation about an arbitrary line*

The transformation matrices for the above operations in the given sequence are

$$[T_1] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -C \\ 0 & 0 & 1 \end{bmatrix} \tag{3.52}$$

$$[T_2] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.53}$$

$$[T_3] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.54}$$

$$[T_4] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.55}$$

$$[T_5] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & C \\ 0 & 0 & 1 \end{bmatrix} \tag{3.56}$$

The required transformation matrix is given by

$$[T] = [T_5]\,[T_4]\,[T_3]\,[T_2]\,[T_1]$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & C \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -C \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T] = \begin{bmatrix} \cos 2\theta & \sin 2\theta & -C\sin 2\theta \\ \sin 2\theta & -\cos 2\theta & C(\cos 2\theta + 1) \\ 0 & 0 & 1 \end{bmatrix} \quad (3.57)$$

Scaling about an Arbitrary Point The transformation given earlier for scaling is about the origin of the axes system. However, sometimes it may be necessary to get the scaling about any arbitrary base point as shown in Fig. 3.32. To derive the necessary transformation matrix, the following complex procedure would be required.

1. Translate the point P to O, the origin of the axes system.
2. Rotate the object by the given angle.
3. Translate the point back to its original position.

The transformation matrices for the above operations in the given sequence are

$$[T_1] = \begin{bmatrix} 1 & 0 & -dX \\ 0 & 1 & -dY \\ 0 & 0 & 1 \end{bmatrix} \quad (3.58)$$

$$[T_2] = \begin{bmatrix} S_x & 1 & 0 \\ 1 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.59)$$

$$[T_3] = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix} \quad (3.60)$$

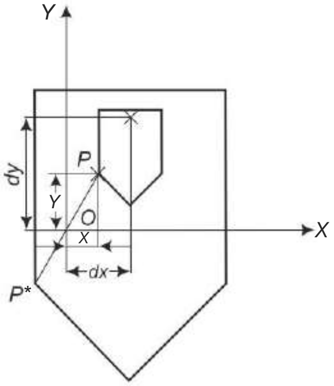


Fig. 3.32 *Example for scaling transformation about an arbitrary line*

The required transformation matrix is given by

$$[T] = [T_3]\,[T_2]\,[T_1]$$

$$[T] = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 1 & 0 \\ 1 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -dX \\ 0 & 1 & -dY \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T] = \begin{bmatrix} S_x & 1 & dX(1 - S_x) - dY \\ 1 & S_y & -dX + dY(1 - S_y) \\ 0 & 0 & 1 \end{bmatrix} \quad (3.61)$$

3.6 3D TRANSFORMATIONS

The 2D transformations as explained above can be extended to 3D by adding the Z-axis parameter. The transformation matrix will now be 4 × 4. The following are the transformation matrices to be used for the purpose.

Translation

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dX \\ 0 & 1 & 0 & dY \\ 0 & 0 & 1 & dZ \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.62}$$

Scaling

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.63}$$

Reflection

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \pm 1 & 0 & 0 & 0 \\ 0 & \pm 1 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.64}$$

Rotation about Z-axis (XY plane)

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.65}$$

Rotation about X-axis (YZ Plane)

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.66}$$

Rotation about Y-axis (ZX Plane)

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.67}$$

3.7 MATHEMATICS OF PROJECTION

In the realm of drawing, there are a number of methods available for depicting the details of a given object. Conceptually looking, if we imagine the plane in which the drawing is being made, termed the projecting plane, it implies that the outlines of the object are actually forming some kind of a shadow on the projecting plane. As shown in Fig. 3.33, it is possible to have a variety of representations to be obtained from the same object depending upon the nature of the projectors, the projecting plane and their inter-relationship.

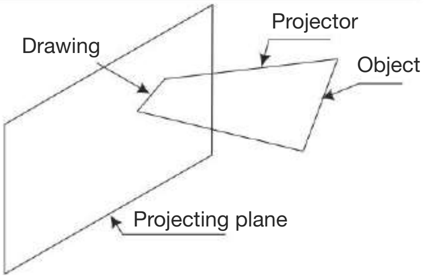


Fig. 3.33 *The principle of projection*

3.7.1 Orthographic Projection

The most common form of projection used in engineering drawings is the 'orthographic projection'. This means that the projecting lines or projectors are all perpendicular (orthogonal) to the projecting plane. As a result, if the feature of the object happens to be parallel to the projecting plane then the true picture and true dimensions would be visible in the orthographic projection.

The orthographic projection system includes a total of six projecting planes in any direction required for complete description. A typical example is shown in Fig. 3.34 where the object is enclosed in a box such that there are 6 mutually perpendicular projecting planes on which all possible 6 views of the object can be projected. This helps in obtaining all the details of the object as shown in Fig. 3.35. The visible lines are shown with the help of continuous lines while those that are not visible, by means of broken lines to differentiate them.

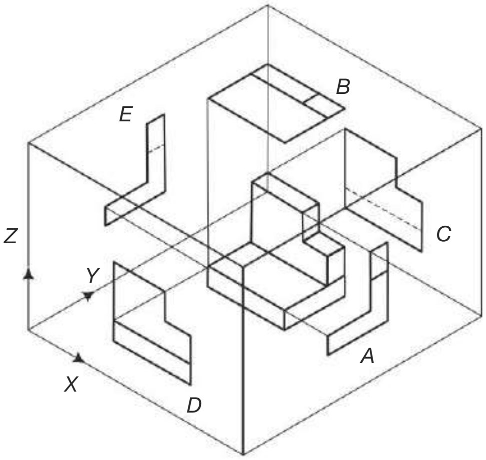


Fig. 3.34 *An object enclosed in a cube to obtain various parallel projections*

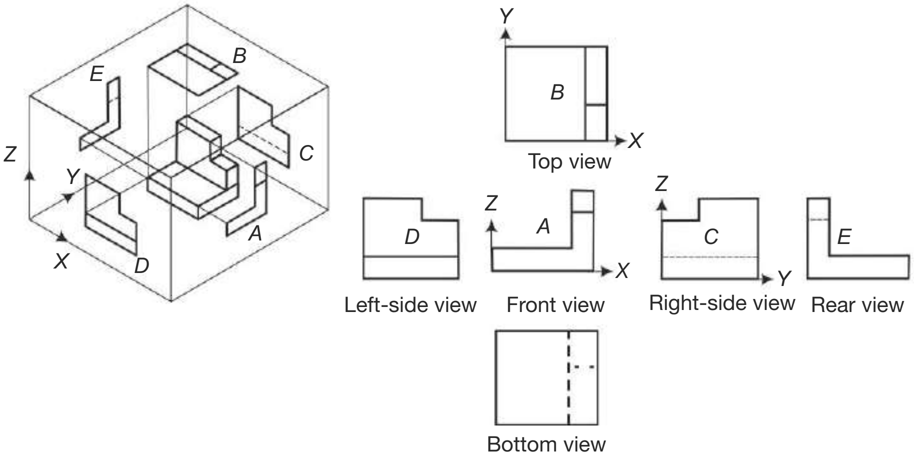


Fig. 3.35 *Orthographic projection of an object*

Obtaining the orthographic projection is relatively straightforward because of the parallel projections involved. The top view can be obtained by setting $z = 0$. The transformation matrix will then be

$$[M_{\text{TOP}}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.68}$$

For obtaining the front view, we put $y = 0$ and then the resulting coordinates (x, z) are rotated by 90° such that the Z-axis coincides with the Y-axis. The transformation matrix will then be

$$[M_{\text{FRONT}}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.69}$$

Similarly, for obtaining the right-side view, we put $x = 0$ and then the coordinate system is rotated such that Y-axis coincides with the X-axis and the Z-axis coincides with the Y-axis. The transformation matrix will then be

$$[M_{\text{RIGHT}}] = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.70}$$

3.7.2 Isometric Projection

An isometric projection is obtained by aligning the projection plane so that it intersects each coordinate axis in which the object is defined at the same distance from the origin. All the three principal axes are foreshortened equally in an isometric projection so that relative proportions are maintained while showing the pictorial view.

The transformation matrix will then be

$$[M_{\text{ISO}}] = \begin{bmatrix} 0.7071 & 0 & 0.7071 & 0 \\ 0.4082 & 0.8165 & -0.4082 & 0 \\ -0.5774 & 0.5774 & 0.5774 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.71}$$

3.8 CLIPPING

Clipping is a very important element in the displaying of graphical images. This helps in discarding the part of the geometry outside the viewing window, such that all the transformations that are to be carried out for zooming and panning of the image on the screen are applied only on the necessary geometry. This improves the response of the system. For example, in Fig. 3.36 the image shown inside the window with dark lines is the only part that will be visible. All the geometry outside this window will be clipped (only for display purpose).

Clipping is used in addition to extracting the part of a scene, for identifying visible surfaces in three-dimensional views; displaying multi-window environments, and selecting objects that can be applied with the necessary geometric transformations such as rotation and scaling.

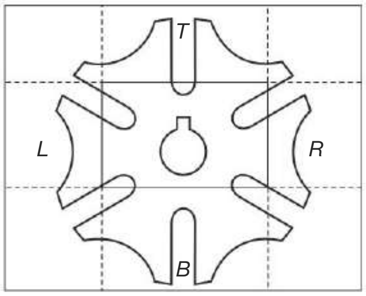


Fig. 3.36 *Clipping of geometric objects outside the display window*

3.8.1 Clipping Lines

In order to carry out the clipping operation, it is necessary to know whether the lines are completely inside the clipping rectangle, completely outside the rectangle or partially inside the rectangle as shown in Fig. 3.37. To know whether a line is completely inside or outside the clipping rectangle, the end points of the line can be compared with the clipping boundaries. For example, the line P_1P_2 is completely inside the clipping rectangle. Similarly line P_3P_4 and P_9P_{10} are completely outside the clipping rectangle. When a line, such as P_5P_6, is crossing the clipping boundary, it is necessary to evaluate the intersection point of the line (P'_6) with the clipping boundary to determine which part of the line is inside the clipping rectangle. The resultant display after clipping is shown in Fig. 3.38.

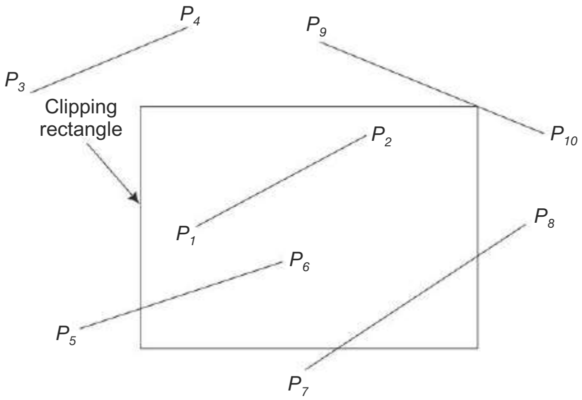


Fig. 3.37 *Clipping of lines – before clipping*

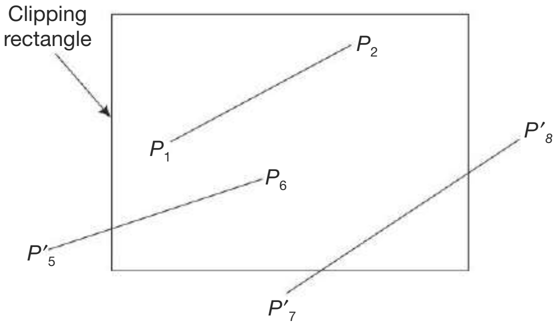


Fig. 3.38 *Clipping of lines – after clipping*

The parametric representation of a line with end points (x_1, y_1) and (x_2, y_2) which is given below could be used to find out the intersection of the line with the clipping boundaries.

$$x = x_1 + u\,(x_2 - x_1)$$
$$y = y_1 + u\,(y_2 - y_1) \tag{3.72}$$

where $0 \le u \le 1$.

If the value of u for an intersection with the clipping boundaries is outside the range 0 to 1 then the line is not inside the clipping rectangle. If the value is between 0 and 1 then the line is inside the clipping rectangle. This method needs to be applied to each of the edges of the clipping rectangle to identify the position of lines. This requires a large amount of computation and hence a number of efficient line-clipping algorithms have been developed.

Cohen–Sutherland Clipping Algorithm in 2D

In this method all the lines are classified as to whether they are in, out or partially in the window by doing an edge test. The end points of the line are classified as to where they are with reference to the window by means of a 4-digit binary code as shown in Fig. 3.39. The code is given as TBRL. The code is identified as follows:

T = 1 if the point is above the top of the window
= 0 otherwise
B = 1 if the point is above the bottom of the window
= 0 otherwise
R = 1 if the point is above the right of the window
= 0 otherwise
L = 1 if the point is above the left of the window
= 0 otherwise

1001	1000	1010
0001	0000	0010
0101	0100	0110

Fig. 3.39 *The 4-digit coding of the line end points for clipping*

The full 4-digit codes of the line end points with reference to the window are shown in Fig. 3.38.

Having assigned the 4-digit code, the system first examines if the line is fully in or out of the window by the following conditions:

The line is completely inside the window if both the end points are equal to '0000'.

The line is completely outside the window if both the end points are not equal to '0000' and a 1 in the same bit position for both ends.

For those lines which are partly inside the window, they are split at the window edge and the line segment is discarded outside the window. There are possibilities when the line may be crossing two regions as shown in Fig. 3.40. For the line P_1P_2, starting from the lower edge, the intersection point P_1' is found and the line P_1P_1' is discarded. Since P_2 is outside the boundary, the intersection point with the boundary P_2' is found out. Since this intersection point is above the window, the second intersection point P_2'' is found out and the line $P_1'P_2''$ is the final line segment inside the clipping boundary.

There are other line-clipping algorithms that are faster and can be found in computer graphics books such as Foley et al [1996].

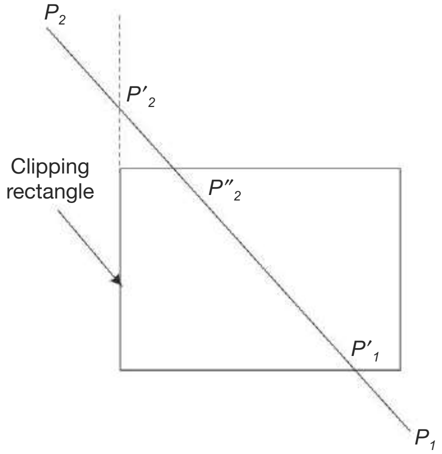


Fig. 3.40 *Lines extending from one coordinate region to the other*

3.8.2 Clipping Polygons

The line-clipping algorithm discussed earlier can be modified to obtain polygon clipping. However, as can be seen from Fig. 3.41, extending the line-clipping procedure described above produces a result which can mean that there exists more than one geometry. This ambiguity is removed by the use of the polygon-clipping algorithm developed by Sutherland and Hodgman.

Sutherland–Hodgman Polygon-Clipping Algorithm in 2D The basic idea used in polygon clipping is that an *n*-sided polygon is represented by *n* vertices. On each of the polygons, two tests are conducted. If the line (edge of the polygon) intersects the window edge, the precursor point is added to the output list. If the next vertex is outside the window, discard it otherwise and add to the output list. This process is repeated for all

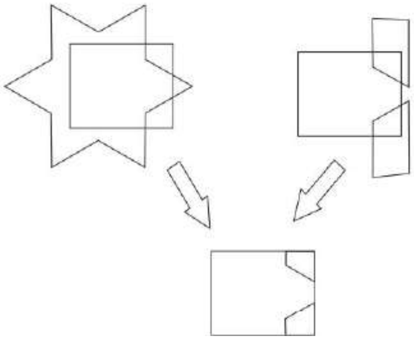


Fig. 3.41 *Identical line clipping of two different geometries*

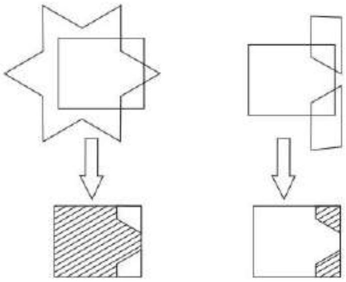


Fig. 3.42 *Clipping produced for different geometries by polygon clipping*

the edges of the polygon. The resulting output is an *m*-sided polygon, which can be displayed as shown in Fig. 3.42. The main advantage of this algorithm is that it can be used for a clipping window that need not be a rectangle. Further, this can be easily extended to 3D.

3.9 HIDDEN LINE/SURFACE REMOVAL

Removing hidden lines and surfaces greatly improves the visualisation of the objects. Looking at Fig. 3.43, the left side shows the butterfly valve body in a wire-frame model. The appearance of the object is greatly complicated by the appearance of the hidden details. However, by removing the hidden details the object geometry is clearly identifiable in the right-hand side.

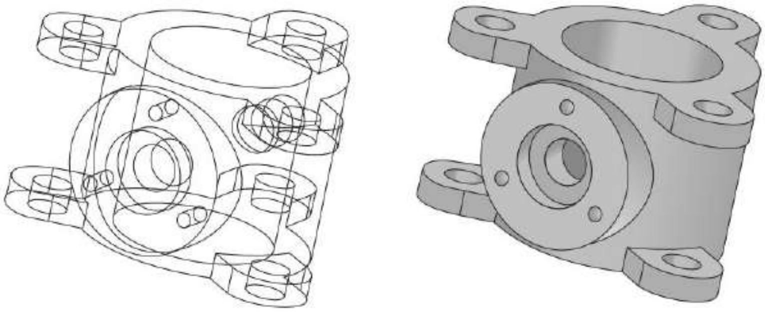


Fig. 3.43 *Need for eliminating the hidden surfaces for clarity*

A wide variety of hidden-surface removal algorithms are in existence. They have been developed historically for the different systems they were supporting. These algorithms may be classified into *object-space methods* and *image-space methods*. The image-space methods can be further divided into *vector* and *raster* methods depending upon the type of displays that are used. However, in view of the type of displays being mostly raster, we will be looking at only those algorithms.

Hidden-surface removal algorithms require considerably large processing power. There is no single best solution for the hidden-surface removal problem. Because of the complexities involved, a large number of algorithms have been developed, some of which are useful for specific applications, while others are useful for general situations. There are many approaches to hidden-surface removal and it is difficult to cover all of them here. Hence, some basic approaches are highlighted here while the reader is advised to refer to more specialised literature to get the details.

The following hidden-surface algorithms are discussed below:

- Back-face removal
- Z-buffer (depth buffer)
- Depth-sort algorithm

3.9.1 Back-face Removal

The basic concept used in this algorithm is for only those faces that are facing the camera (centre of projection). The normal form of a polygon face indicates the direction in which it is facing. Thus, a face can be seen if some component of the normal N (Fig. 3.44) is along the direction of the projector ray P.

If an object can be approximated to a solid polyhedron then its polygonal faces completely enclose its volume. It is then possible that all the polygons can be defined such that their surface normals point out of their

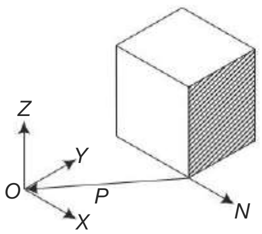


Fig. 3.44 *Back-face removal using the surface normal and projecting ray*

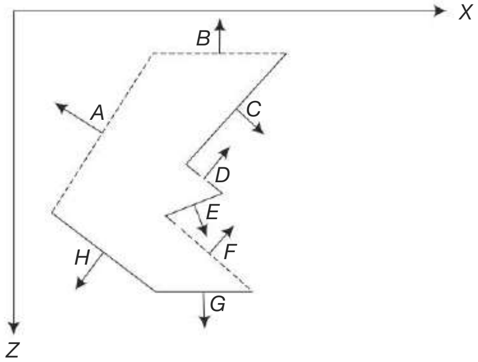


Fig. 3.45 *Back-face removal using the surface normal – A, B, D, F are back faces and C, E, G, H are front faces*

polyhedron faces as shown in Fig. 3.45 for a polygon slice. If the interior of the polyhedron is not exposed by the front clipping plane then those polygons whose surface normals point away from the camera (observer) lie on a part of the polyhedron which is completely blocked by other polygons that are closer (Fig. 3.45). Such invisible faces called *back faces* can be eliminated from processing leaving all the front faces.

As can be seen, not all the front faces are completely visible. Some may be completely obscured by other faces (such as *E*) or partially visible (such as *C*). This method allows identifying the invisible faces for individual objects only. However, in majority of the cases this removes almost 50% of the surfaces from the database, which can then be processed faster by the other algorithms.

3.9.2 *Z*-Buffer Algorithm (Depth Buffer)

The *Z*-buffer is a separate depth buffer used to store the *z*-coordinate (depth) of each pixel in the geometric model. This method utilises the principle that for each of the pixel locations, only that point with the smallest *z*-depth is displayed. Figure 3.46 shows two surfaces S_1 and S_2 with varying distances along the position (x, y) in a view plane. Surface S_1 is closest at this position, so its surface depth value is saved at the (x, y) position.

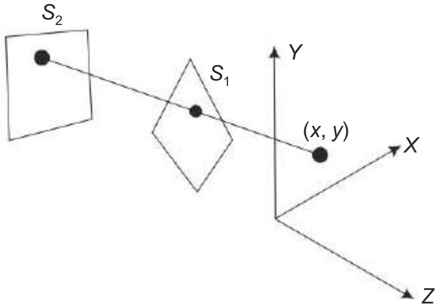


Fig. 3.46 *z-buffer method – S_1 is closer to the observer (x, y) than S_2*

For this purpose, it constructs two arrays:

- $Z(x, y)$ the dynamic nearest *z*-depth of any polygon face currently examined corresponding to the (x, y) pixel coordinates.
- $I(x, y)$, the final output colour intensity for each pixel, which gets modified as the algorithm scans through all the faces that have been retained after the back-face removal algorithm.

The first face is projected on to the viewing plane and the $Z(x, y)$ and $I(x, y)$ arrays are filled with the *z*-depth and colour of the face. The next polygon is projected and its *z*-depth for each pixel location is compared with the corresponding one that is stored in $Z(x, y)$. If the new *z*-depth is smaller then it replaces the existing value, while its colour is stored in the corresponding position in $I(x, y)$. This process is repeated for all the faces. Thus, the image stored in $I(x, y)$ is the correct image, accurate to the nearest pixel with all the hidden surfaces removed.

The main advantage of the algorithm is its simplicity and the amount of storage required. The disadvantage of the method is the difficulty in implementing anti-aliasing, transparency and translucency effects. The reason for this is that the algorithm writes the pixels to the frame buffer in an arbitrary order, and the necessary information for pre-filtering anti-aliasing techniques is not easily available. Similarly, for transparency and translucency effects, pixels may be written to the frame buffer in incorrect order, leading to local errors.

3.9.3 Depth-Sort Algorithm

The depth-sort algorithms utilise the principle of painting the polygons into the frame buffer in the order of decreasing distance (depth) from the view point. This can be done by following the three steps as shown below:

- Sort all the polygons according to the smallest z-coordinate of each.
- Resolve any ambiguities (Fig. 3.47) when the z-coordinates of polygons overlap, by splitting the polygons.
- Scan and convert each polygon in ascending order from back to front (in terms of z-coordinate).

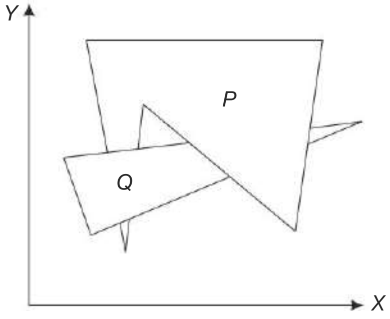


Fig. 3.47 *Depth-sort algorithm–overlapping of polygons in the depth direction*

This method is often called the *painter's algorithm*, since it utilises the procedures followed by artists in oil painting. The artist first paints the background colours and then adds the distant objects first. Later, he adds the nearer objects in the order of decreasing depth. Finally, the foreground objects are added to the canvas over the background and other objects that have already been painted. Each new layer of paint added covers the paint already present on the canvas.

This process is carried out in a number of steps. All the surfaces (polygons) are ordered in the first pass according to the smallest z-value on each surface. The surface with the largest depth (z-value) is compared with all other surfaces in the list to compare if there is any overlap in the z-direction. If there is no overlap then it is scan converted to fill the frame buffer. The same procedure is completed for all other surfaces in the list if there is no overlap. If an overlap is detected then further tests need to be done on that surface to examine the visibility.

The following tests are conducted to see if re-ordering of surfaces is necessary. If any of these tests are true then we proceed to the next surface.

- The bounding rectangles in the xy plane for the two surfaces do not overlap.
- One surface is completely behind the overlapping surface relative to the viewing position.
- The overlapping surface is completely in front of the surface relative to the viewing position.
- The projections of the two surfaces on to the viewing plane do not overlap.

Some examples are shown in Fig. 3.48. There is an overlap in the z-direction between the surfaces S_1 and S_2. However, there is no overlap in the x-direction. Then we also check in the y-direction. If there is no overlap, then S_2 cannot overlap S_1. The surface S_4 is completely in front of the surface S_3 but the surface S_3 is not completely inside S_4. When all the first three tests fail then the intersection between the bounding edges of the two surfaces is checked as shown in Fig. 3.49. The two surfaces may or may not interfere, but still the test fails because of the intersection of the bounding edges.

When all the tests fail, the order of the surfaces in the list is interchanged and the procedure repeated. There is no guarantee that even after interchanging we may not come across situations where the surfaces may get into an infinite loop where the same surfaces may need to be continuously reordered in the processing. In

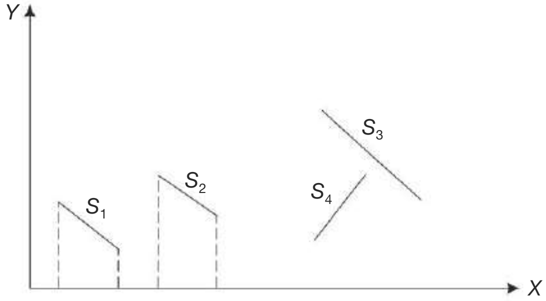


Fig. 3.48 *Depth-sort algorithm–surfaces that are overlapping and non-overlapping*

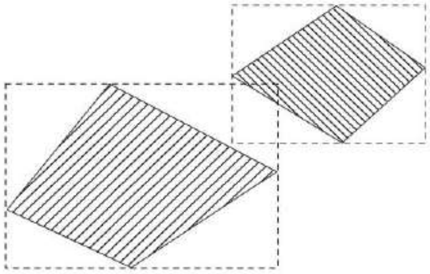


Fig. 3.49 *Depth-sort algorithm–overlapping bounding rectangles in the xy plane*

such cases, some of these surfaces need to be flagged and re-ordered to a further depth position so that it cannot be moved again. Alternatively, the surfaces that are being moved more than twice may be divided into two surfaces and the processing continued.

3.10 COLOUR

The human visual system can distinguish only a few shades of grey, while it has much more discriminating power with respect to colour shades. Thus, colour provides a lot of information about the object displayed. Our perception of colour is determined by the colour of the light source and the reflectance properties of the object, since only those rays that are reflected are seen, while others are absorbed. The use of colours enhances the presentation of information in CAD/CAM in a number of ways. Using different colours for different types of geometric entities during the construction stage helps the engineer to follow the modelling processes with more clarity. For example, in a wireframe, surface, or a solid modelling process, the entities can be assigned different colours to distinguish them. In addition, to get a realistic appearance of the object, colour becomes very important for the shaded images produced by shading algorithms as seen later. In finite element analysis, colours can be used effectively to display contour images such as stress or heat-flux contours.

3.10.1 Colour Models

There are several different colour models that are used:

- RGB Model
- CMY Model
- HSI Model
- YIQ Model

RGB Model In the RGB model, an image consists of three independent image planes: red, green and blue. This is an additive model, i.e., the colours present in the light add to form new colours, as shown in Fig 3.50. The other colours obtained are yellow (red + green), cyan (blue + green), magenta (red + blue) and white (red + green + blue). A particular colour is specified by the amount of each primary colour to get all the other shades of colours. This model is appropriate for the mixing of coloured light and is used for colour monitors and most video cameras.

CMY Model Unlike the RGB model, the CMY (cyan-magenta-yellow) model is a subtractive model (Fig 3.51). The three primary colours are cyan (C), magenta (M) and yellow (Y). The other colours obtained are red (yellow + magenta), green (cyan + yellow), blue (cyan + magenta) and black (cyan + magenta +

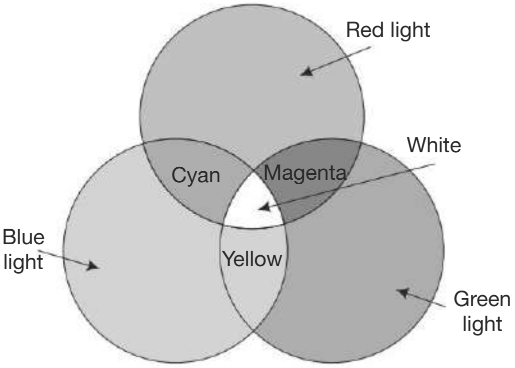


Fig. 3.50 *Colour model—RGB*

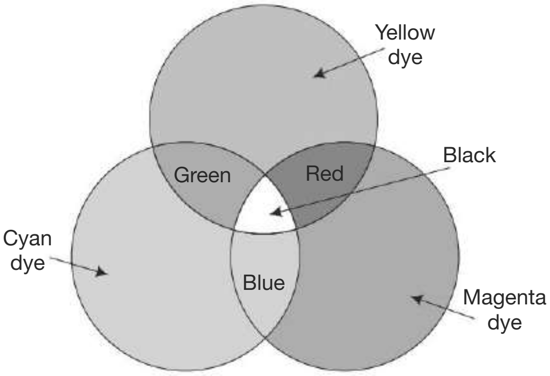


Fig. 3.51 *Colour model—CMY*

yellow). Hence, it is used in cases where absorption of colours is valid. In contrast to the RGB model, where colours are added to black to get a particular colour, the CMY model defines what is subtracted from white. It is used with colour printers and photocopiers.

In the printing industry, this model is frequently called the CMYK model instead of CMY. Here, *K* stands for *black*. Though the mixture of cyan, magenta, and yellow ink should absorb all the primary colours and print as black, it normally comes as muddy brown. Hence, all the colour printers (inkjet as well as laser) or photocopiers have a separate cartridge for black ink or toner in addition to the cyan, magenta, and yellow.

3.11 SHADING

Having studied the principles that are required for displaying realistic graphical images without the hidden surfaces, the next logical step is to render them by incorporating the correct shading of different surfaces. In order to get a realistic appearance, it is necessary to render the object with the actual colours as closely as possible. Further improvement will be to add colour. In order to do this, it is necessary to understand the mechanics of light as reflected from the object based on its reflectivity.

3.11.1 Object Lighting

All objects emit light, whose origins could be many and varied. The object itself may be emitting light. A more common phenomenon is the reflection of the light that is falling on the object. The light that is falling on the object can be coming from a light source such as the sun or a light bulb, or a reflected light, for example, the reflected light from the light bulb from the surrounding walls. As a result, the light emanating from any object depends upon many sources and the entire scene, therefore, needs to be taken into account to render the object correctly. It is the reflected light that allows the object to be seen as shown in Fig. 3.52.

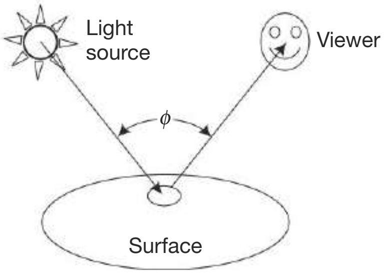


Fig. 3.52 *Basic illumination of an object by a light source*

Reflection When light falls on a surface, it can be reflected, absorbed or transmitted through the surface. Reflections can be classified into two types: diffuse reflection and specular reflection. In *diffuse reflection*, the incident light is randomly reflected in all directions. If the incoming light rays are diffused then the reflection from the object will also be diffused. Diffuse reflection alone does not give any visual realism since the light from the object has no relation to the incident light. The observer will not be able to identify if it is a flat surface or a convex surface.

Part of the light incident on the surface reflects back to the observer. With diffuse reflection, the proportion of light reflected back is dependent simply on the surface properties and not dependent on the angle of the viewer. The observed intensity (E) of light on a surface can be written as

$$E = R\,L \tag{3.73}$$

where R is the reflection coefficient of the surface and is $0 \leq R \leq 1$, and

L is the strength of the incident illumination.

The intensity of the reflected light is proportional to the cosine of the angle between the surface and the incoming light direction, which is called *Lambert's cosine law*.

Specular reflection is the property of a surface that reflects the incident light in a nearly fixed direction and without affecting its quality. It is this reflection that is responsible for the highlights seen on shiny objects. It is normally assumed that all incoming wavelengths of the light are reflected equally. Another aspect that should be noted is the Mach band effect. The *Mach band effect* makes smooth intensity changes on surfaces look sharper than they really are. It occurs whenever the light intensity on the object surfaces changes sharply. A very nice Java illumination model can be seen on the Internet. You may play with the parameters to get an idea about the illumination effects. (http://www.siggraph.org/education/materials/HyperGraph/illumin/vrml/pellucid.html)

3.11.2 Shading Methods

In order to present an object, shading is an important form of display that enhances the realism. All these methods are computationally intensive. A few of the techniques used for shading in CAD are discussed here.

Lambert Shading In the Lambert shading method, also called *faceted shading*, the surface of the object is approximated by polygons (flat surface) even though it is smooth. This gives rise to the appearance of the smooth solid as a faceted solid, shown in Fig. 3.53. The surface normal for each polygon is computed and the illumination model is applied to get the intensity of the surface. The polygon is then filled with this intensity. The result is seen in Fig. 3.53 as a faceted solid. By increasing the number of polygons, the surface can become smooth. However, for complex objects the number of polygons may be too large to get a smooth appearance. Thus, the Lambert shading method will rarely provide a smooth-looking surface.

Gouraud Shading Gouraud shading, invented by Henri Gouraud in 1971, is a method to display each surface polygon with an intensity that varies smoothly. The smoothness is obtained by linearly interpolating a colour or shade across the polygon unlike the Lambert shading which has a single colour across the entire polygon. In order to do this, the intensity is calculated for each pixel rather than one intensity for the entire polygon. It is also called *intensity interpolation* and is a very simple and effective method of adding a curved feel to a polygon that would otherwise appear flat. An example of a Gouraud shaded cylinder is shown in Fig. 3.53.

When two polygons meet, the colour results for the neighbouring pixels across the border are interpolated so that they end up with approximately the same colours. Thus, the adjoining polygons avoid the sudden discontinuity at the border by having the same colour.

First, the colour at each vertex of the polygon is calculated similar to the Lambert shading procedure as explained earlier. Then the colour for all the remaining pixels in the polygon can be calculated by following the appropriate interpolating from the vertices. For example, for a pixel that is in the middle of one of the edges, the colour value should be halfway between the colour values at the two ends. In fact, the colours are

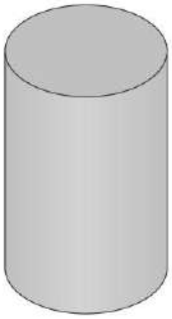


Fig. 3.53 *Lambert shading of a faceted solid*

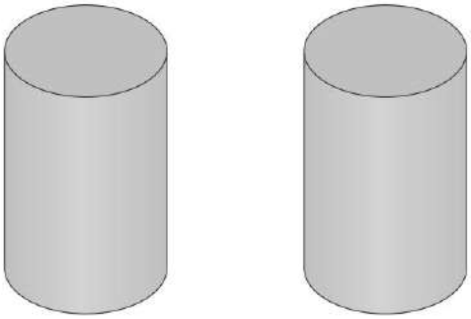


Fig. 3.54 *Gouraud shading of a faceted solid compared to Lambert shading*

normally calculated for the pixels along the scan line through the polygon. This procedure is explained below with reference to Fig. 3.55.

Imagine that a polygon *XYZ,* as shown in Fig. 3.55, is to be shaded. Each square in the figure refers to a pixel on the screen. The hatched squares represent the actual pixels being occupied by the polygon. Since the rendering will be done along the scan lines, currently the scan line *AB* is being rendered. It may be noticed that *A* and *B* are not falling at the centre of the pixels. The two ends of the line are actually passing through pixels *C* and *D*. Hence, it is necessary to make a slight adjustment for increased accuracy. The gradient of the shade, G_{AB} along the line is given as

$$G_{AB} = \frac{S_B - S_A}{X_B - X_A} \tag{3.74}$$

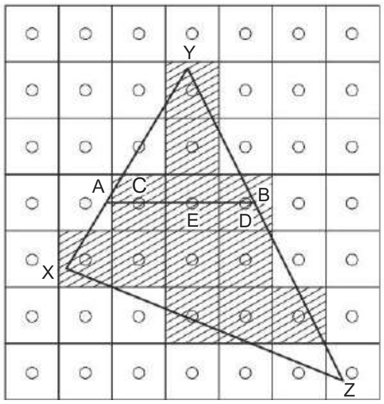


Fig. 3.55 *Gouraud shading principle of a polygon*

where S_A = shade at *A*,

S_B = shade at *B,*

X_A is the *X*-coordinate value of *A*, and

X_B is the *X*-coordinate value of *B*.

From the gradient, the exact value of the shade at *C* can be calculated as

$$S_C = S_A + (X_C - X_A) * G_{AB} \tag{3.75}$$

Since the shade changes linearly across the scan line, the shade along all the pixels can be calculated in a similar manner using the following formulae:

$$S_E = S_A + (X_E - X_A) * G_{AB}$$
$$S_D = S_A + (X_D - X_A) * G_{AB} \tag{3.76}$$

By performing separate calculations for red, green and blue, a complete colour value can be obtained for each of the pixel.

Though Gouraud shading is an improvement over flat shaded polygons, problems occur when mixing light source calculations with big polygons. It tends to miss certain highlighting, particularly with specular reflections. Also, Gouraud shading can introduce anomalies known as Mach bands**.**

Phong Shading Bui-Truong Phong developed the Phong shading method in 1975. It is similar to Gouraud shading in using the vertex normals. However, Phong shading interpolates the vertex normals themselves rather than the intensities at the vertices. It is also called the *normal-vector interpolation shading method*. The procedure to be followed for Phong shading is as follows:

- Determine a normal vector at each vertex of a polygon similar to the procedure used in Gouraud shading.
- Interpolate normal vectors along the edges of the polygon.
- Interpolate normal vectors across each scan line, so that there is one normal vector for each pixel in the polygon.
- Apply an illumination model at positions along scan lines to calculate pixel intensities using the interpolated normal vectors.

Similar to the Gouraud shading for each scan line in the polygon, evaluate the normal vectors at each pixel by the linear interpolation of the normal vectors at the end of each line (N_A and N_B) as shown in Fig. 3.56. The normal vector thus calculated for each pixel on the polygon is an approximation to the real normal on the curved surface approximated by the polygon.

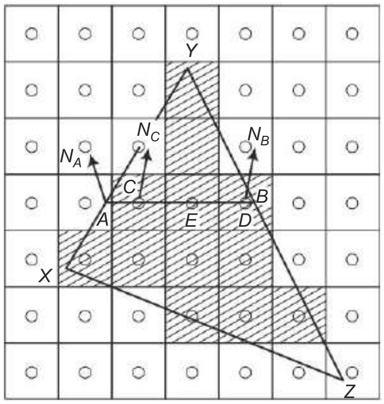


Fig. 3.56 *Phong shading principle of a polygon*

$$N_C = \frac{N_A(X_B - X_C) + N_B(X_C - X_A)}{X_B - X_A} \tag{3.77}$$

The interpolated normal vector N_C, is then used in the intensity calculation. Since this is a vector equation it needs to be solved for each of the coordinates (X, Y and Z) direction. This makes the Phong shading three times more expensive compared to Gouraud shading.

Summary

- We have studied various principles that are required for displaying the graphical images on the output device.
- In order to display the graphical information, which is vectorial in nature, it is necessary to convert it into raster format.
- For converting lines into raster format, the DDA algorithm is simplest while Bresenham's algorithm reduces the computations into integer format, thereby making it a faster alternative.
- It is necessary to modify the pixel information for display to get a more realistic visual experience.
- Depending upon the type of graphic display used, it is necessary to be familiar with a number of different coordinate systems to facilitate the graphic construction as well as display.
- In addition to the actual graphic information, a large amount of additional data such as organisational and technological data is stored with the product data.
- Three different types of data models are used for storing modelling data. They are the hierarchical model, network model and the relational model. Of these, the relational model is quite extensively used because of its flexibility in usage. It maintains data tables and pointers to store the required data in a very compact manner.

- Engineering Data Management (EDM) takes care of all the data that is required for a given product throughout its lifecycle.
- Geometric transformations can be handled conveniently using matrix algebra. For this purpose, it is necessary to use homogenous representation of vertex data.
- Various transformations that are quite useful are translation, rotation, scale and reflection. It is possible to extend these basic transformations for more complex transformations.
- The 2D transformation methods can be easily extended into 3D.
- The 3D geometry data needs to be converted into 2D by adopting a suitable projection system such as orthographic, isometric or perspective projection.
- Since only part of the geometric model will be displayed most of the time, it is necessary to clip the information outside the display window. A number of methods are available for straight line as well as polygon clipping.
- Line clipping is simple, but polygon clipping is more comprehensive and can be easily extended to 3D.
- Also, it is necessary sometimes to remove the hidden lines to make the display easier to understand. For this purpose, back-face removal, depth buffer (Z) and depth-sort algorithm are used.
- There are a number of colour models used with the graphic displays and hard copies. Adding colour and shading coupled with the removal of hidden surfaces allows for realistic visualisation of objects modelled.
- Gourad shading and Phong shading models are frequently used to get smoothly shaded 3D profiles.

Questions

1. Explain the basic principle of (i) the DDA, and (ii) Bresenham's algorithms for the linear interpolation for graphics terminals (no derivation of the relations). Explain the relative advantages of the methods.
2. Derive the relationship for rasterisation of vectors using the principle of (i) the DDA, and (ii) Bresenham's algorithms for the linear interpolation for graphics terminals.
3. Explain the concept of anti-aliasing of lines. Give examples of their implementations in graphic terminals.
4. Briefly explain the concept of various coordinate systems required for geometric display systems. Give examples.
5. Briefly explain the requirements for a graphic database.
6. What are the functionalities expected of a graphic database structure?
7. What are the different types of data models used in a graphic database structure?
8. Explain briefly the hierarchical model of a graphic database.
9. Explain briefly the network model of a graphic database.
10. Explain briefly the relational model of a graphic database.
11. Explain briefly what you know about EDM (Engineering Data Management) systems.
12. Describe some important features of an EDM (Engineering Data Management) system.
13. Explain the type of database structure that is generally employed in solid modelling.
14. What are the various types of information normally stored in a geometric database for products in a CIM environment?

15. Briefly explain the various graphic transformations required for manipulating the geometric information.
16. What is the need for concatenation of transformations? Explain the care to be taken in such cases.
17. Explain why the homogeneous coordinate system is generally used in graphics in place of a normal coordinate system, in particular for software implementation. Give an example to illustrate the advantage.
18. Derive the relationship for geometric rotation in the *YZ* plane.
19. Explain the concept of obtaining a rotation about an arbitrary point in the *XY* plane.
20. Explain the concept of obtaining a reflection about an arbitrary line starting from the plain reflection about an axis.
21. How do you obtain the orthographic projections of a 3D geometric database?
22. Explain the importance of clipping. Give details of methods used for line clipping.
23. Explain the Cohen–Sutherland clipping algorithm.
24. Explain the Sutherland–Hodgman clipping algorithm.
25. Explain the details of polygon clipping. Give its advantages compared to line clipping.
26. Explain the method of back-face removal. Give its advantages and limitations with reference to hidden-line removal.
27. Give the details of *z*-buffer method for hidden-surface removal.
28. Explain the importance of colour in CAD/CAM applications.
29. Give the two most common colour models used in CAD.
30. Give the details of depth-sort algorithm for hidden-surface removal.
31. What is the effect of specular reflection of objects on illumination noticed?
32. Explain the Lambert shading model for faceted solids.
33. What are the improvements brought by Gouraud shading compared to Lambert shading?
34. Explain the Phong shading procedure.

Problems

1. An object is to be rotated about an axis parallel to the *X* axis, but its origin passes through a point (X_c, Y_c). Obtain the necessary transformation matrix in two dimensions.
2. A scaling factor of 2 is applied in the *Y* direction while no scaling is applied in the *X* direction to the line whose two end points are at coordinates (1, 3) and (3, 6). The line is to be rotated subsequently through 30° in the counter-clockwise direction. Determine the necessary transformation matrix for the operation and the new coordinates of the end points.
3. The vertices of a triangle are situated at points (15, 30), (25, 35) and (5, 45). Find the coordinates of the vertices if the triangle is first rotated by 10° in a counter-clockwise direction about the origin and then scaled to twice its size. Draw the triangles on a graph paper or in Auto CAD.
4. Prove that the multiplication of transformation matrices for the following sequence of operations is commutative:
   (a) Two successive rotations
   (b) Two successive translations
   (c) Two successive scalings
5. Prove that a uniform scaling and a rotation form a commutative pair, but that, in general, scaling and rotation are not commutative.
6. Show that transformation matrix for a reflection about the line $Y = +X$ is equivalent to a reflection relative to the *X*-axis, followed by a counter-clockwise rotation of 90°.
7. Show that the transformation matrix for a reflection about the line $Y = -X$ is equivalent to a reflection relative to the *Y*-axis, followed by a counter-clockwise rotation of 90°.

8. Determine the form of transformation matrix for a reflection about an arbitrary line with the equation $Y = mX + c$.
9. Calculate the transformed (rotation of 30°) coordinates of the following:
   (a) a rectangle of size 25 mm × 40 mm
   (b) a hexagonal prism of 15 mm edge and 120 mm height located at any convenient point
   (c) a frustum of a cone of 60 mm base diameter and 30 mm top diameter with a height of 40 mm
10. If a line is represented by the following in two dimensions, $2Y = 3X - 5$, find out the final position of the midpoint of the line. The line before the transformation starts at (1, –1) and ends at (5, 5). The transformation being carried is rotation about the origin in the *XY* plane (about *Z*-axis) of 30° in the counter-clockwise direction.
11. The two ends of a straight line have coordinates *A* (0.5, 1.5) and *B* (1, 2.5). The line must be rotated through 40° in the counter-clockwise direction about the origin in the *XY* plane and then translated 4 units in the +*X* direction. Write the necessary transformation matrix and determine the new coordinates of the two end points.
12. Show that the midpoint of a line transforms to the midpoint of the transformed line.
13. Show that
   (a) Translation is commutative
   (b) Mirror and 2D rotation about the *Z*-axis are not commutative
   (c) Scaling and 2D rotation about the *Z*-axis are commutative
14. A cube of 10 unit length has one of its corners at the origin (0, 0, 0) and the three edges along the three principal axes. If the cube is to be rotated about the *Z*-axis by an angle of 30° in the counter-clockwise direction, calculate the new position of the cube.
15. A square with an edge length of 15 units is located in the origin with one of the edges at an angle of 30° with the + *X*-axis. Calculate the new position of the square if it is rotated about the *Z*-axis by an angle of 30° in the clockwise direction.
16. The two ends of a straight line have coordinates (1, 3) and (2, 5). It is to be reflected about a straight line that passes through the points (0, 0.5) and (4, 6). Write the necessary transformation matrix for the above operation, and determine the new coordinates of the end points of the line.
17. The two end points of a line segment have coordinates (1, 3) and (3, 6). If this is to be scaled to twice its present size, write the transformation matrix and the coordinates of the new end points.
18. A point *P* lies originally at the position ($\sqrt{2}$, 0). Find its coordinates if it is rotated 45° clockwise about the origin. If it is given a subsequent rotation of 45°, what will its coordinates be?

Part – II

DESIGN OF
INDUSTRIAL PRODUCTS

4

GEOMETRIC MODELLING

Objectives

Geometric modelling constitutes the most important and complex part in most of the present-day software packages. After completing the study of this chapter, the reader should be able to

- Understand the various requirements for the information that is generated during the geometric modelling stage
- Study various types of geometric models possible and their applications
- Develop various methodologies used for geometric construction such as sweep, surface models, solid models, etc.
- Recognise the various types of surfaces and their applications as used in geometric modelling
- Appreciate the concept of parametric modelling which is the current mainstay of most of the 3D modelling systems
- Develop the various mathematical representations of the curves and surfaces used in geometric construction
- Understand the parametric representation of curves and surfaces
- Understand the solid construction methods including b-rep and CSG methods
- Discuss the various CAD system requirements that need to be considered while selecting a system for a given application
- Understand the concept of rapid prototyping and the various methods available for the purpose

4.1 REQUIREMENTS OF GEOMETRIC MODELLING

What does geometric modelling mean? What is it expected to provide? The total product cycle in a manufacturing environment involves a large number of interconnected functions. The concept of a product originates in the designer's mind. If it is to be translated into reality, he needs to present it in a relevant form for the manufacturing engineer to understand and carry out the necessary operations on it for its production (Fig. 4.1). The total process will have to be carefully integrated such that a large amount of duplication of work is avoided. This may consist of some kind of a problem identification based on market research, product feedback or some innovative new idea. Based on this input, some preliminary ideas could be developed. These preliminary ideas can then be expanded into some preliminary designs, which are basically gross designs, without going really into the various principles involved. Based on all this, one or more candidate designs could be generated which needs further exploration. For all these activities, geometric modelling becomes the central part that is manipulated at all these stages as shown in Fig. 4.1.

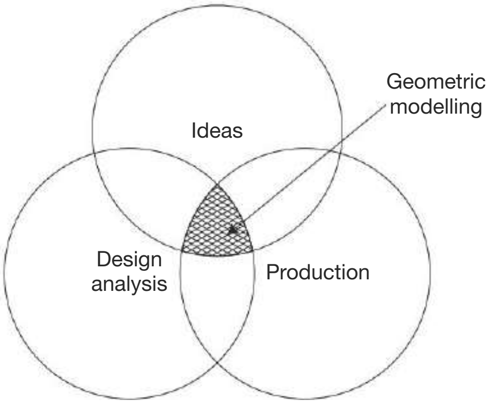


Fig. 4.1 *Total product cycle in a manufacturing environment*

Traditionally, product drawings were made together with the prototypes for passing across the information. However, in a computerised environment, the information a designer generates can form the basic unit which is accessed by a number of other elements of a CAM system, as explained in the first chapter. Hence, it is important that the geometric model generated should be as clear and comprehensive as possible so that the other modules of the modelling and manufacturing system are able to use this information in the most optimal way.

The functions that are expected of geometric modelling are

Design Analysis

- Evaluation of areas and volumes
- Evaluation of mass and inertia properties
- Interference checking in assemblies
- Analysis of tolerance build-up in assemblies
- Analysis of kinematics—mechanics, robotics
- Automatic mesh generation for finite element analysis

Drafting

- Automatic planar cross-sectioning
- Automatic hidden line and surface removal
- Automatic production of shaded images
- Automatic dimensioning
- Automatic creation of exploded views for technical illustrations

Manufacturing

- Parts classification
- Process planning
- Numerical-control data generation and verification
- Robot program generation

Production Engineering

- Bill of materials
- Material requirement
- Manufacturing resource requirement
- Scheduling

Inspection and Quality Control

- Program generation for inspection machines
- Comparison of produced part with design

In view of such varied applications, the geometric-modelling technique used has to provide all such facilities for interaction. The modelling system should be able to describe the parts, assemblies, raw material used, and the manufacturing requirements. From geometric models (of parts, assemblies, stock and tools), it is possible to obtain manufacturing, assembly and inspection plans and command data for numerically controlled machine tools.

Another important aspect to be considered with geometric modelling is the fact that the ways of traditional designers are followed as far as possible. This is easier said than done. But what brings out this interaction is the 'interactive graphic environments' provided by most of the operating systems. The user (designer) is able to see his designing process instantly, and is thereby in a position to take any corrective action as required. However, in the batch mode of operation, which was prevalent in most of the earlier modelling systems, this aspect was missing.

Therefore, it becomes necessary for the geometric modelling system to provide complete information on all aspects related with the further use of the system and at the same time be simple and in tune with the designer's methods. Requicha and Voelker [1981] specified the following properties to be desired of in any geometric modelling (solids) system.

1. The configuration of the solid (geometric model) must stay invariant with regard to its location and orientation.
2. The solid must have an interior and must not have isolated parts.
3. The solid must be finite and occupy only a finite shape.
4. The application of a transformation or other operation that adds or removes parts must produce another solid.
5. The model of the solid in E^3 (Euler space) may contain infinite number of points. However, it must have a finite number of surfaces which can be described.
6. The boundary of the solid must uniquely identify which part of the solid is exterior and which is interior.

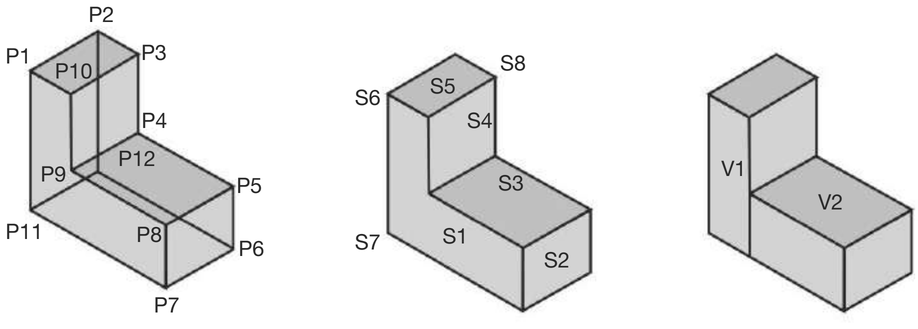
4.2 GEOMETRIC MODELS

There are a large number of geometric modelling methods that have been represented in the literature referenced in the end. All these models satisfy the requirements presented in the previous article.

The geometric models can be broadly categorised into two types:

1. two-dimensional, and
2. three-dimensional.

The two-dimensional models were the first ones to be developed in view of their relatively lesser complexity. However, their utility is limited because of their inherent difficulty in representing complex objects. Their utility lies in many of the low-end drafting packages, or in representing essentially two-dimensional manufacturing applications such as simple turning jobs (axi-symmetric), sheet-metal punching or flame or laser cutting. Serious CAM applications are extremely difficult in operation, if they start from the two-dimensional geometric modelling. Hence, hardly any application exists with only a two-dimensional geometric modelling.



(a) Line model (b) Surface model (c) Volume model

Fig. 4.2 *3D geometric representation techniques*

In contrast, three-dimensional geometric modelling has the ability to provide all the information required for manufacturing applications. There are a number of ways in which the three-dimensional representation can be arrived at. The three principal classifications can be

1. the line model (wireframe modelling),
2. the surface model, and
3. the solid or volume model.

These are represented in Fig. 4.2.

Of these, the line model is the simplest and is used in low-cost designing systems. The complete object is represented by a number of lines with their endpoint coordinates (x, y, z), and their connectivity relationships. This is also called wireframe representation. Though it appears simple, as seen in Fig. 4.3, there is an ambiguity possible as it is difficult to understand the outside of the solid represented by the wireframe model in Fig. 4.4. Thus, the line model or wireframe model is inadequate for representing the more complex solids. It is also possible to draw some impossible solid objects using wireframe modelling as shown in Fig. 4.5. However, in view of the simpler manipulation methods used in organising the wireframe models, these are used in low-end designing and manufacturing systems. Examples could be AutoCAD, Versa CAD, Personal designer of ComputerVision, Micro Station, CADKEY, etc. To fully describe the nature of the solid, it is essential to store further information in addition to the vertex data. Most of these systems have now progressed to provide comprehensive three-dimensional facilities in addition to the two-dimensional methods described above.

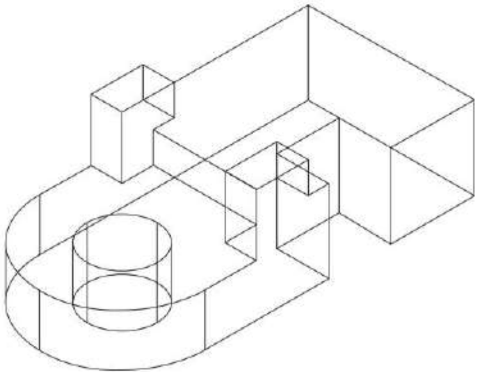


Fig. 4.3 *A geometric model represented in wireframe model*

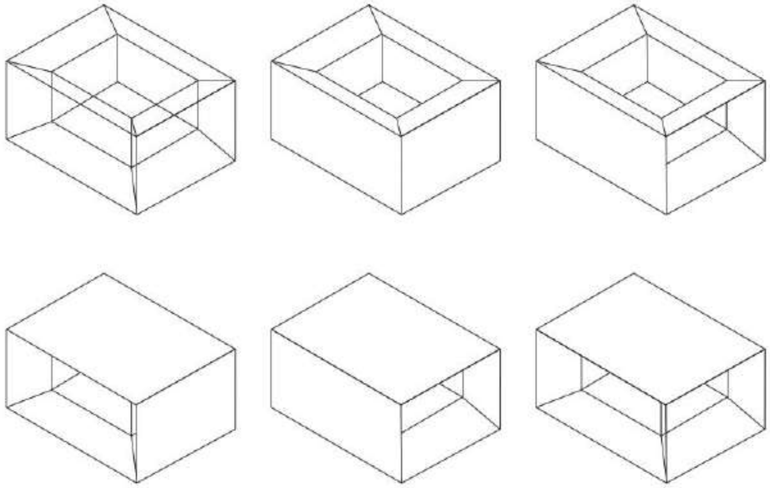


Fig. 4.4 *Ambiguities present in the wireframe model*

The surface model is constructed essentially from surfaces such as planes (as shown in Fig. 4.6), rotated curved surfaces (ruled surfaces) and even very complex surfaces. These are often capable of clearly representing the solid from the manufacturing point of view. However, no information regarding the interior of the solid model is available which can be relevant for generating the NC cutter data. Further, the calculation of properties such as mass and inertia is difficult. Thus, this model, as a complete technique for constructing the solid, is extremely tedious and is not generally attempted. But these facilities are available as part of the modelling technique, and are used when such a surface is present in the product for design. For example, this method is used more for specific non-analytical surfaces, called *sculptured surfaces*, such as those used for modelling car bodies and ship hulls. There are a number of mathematical techniques available for handling these surfaces such as Bézier and B-splines, details of which are presented later in this chapter.

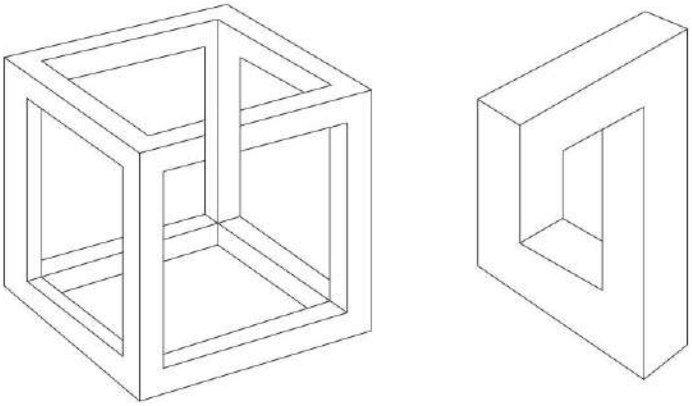


Fig. 4.5 *Impossible objects that can be modelled using a wireframe model*

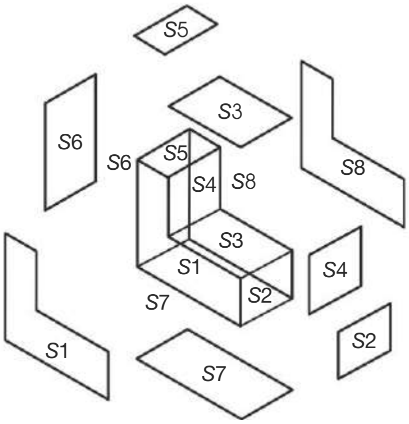


Fig. 4.6 *Generation of 3D geometry using planar surfaces*

The solid or volume model consisting of the complete description of the solid in a certain form is the most ideal representation, as all the information required for manufacturing can be obtained with this technique. This is the most widely used method and a number of different techniques are employed in representing the data related to the solid, the details being presented below. A comparison of the capabilities and applications of the various modelling techniques is presented in Table 4.1.

Table 4.1 *Comparison of the different modelling methods*

	Line model	*Surface model*	*Volume model*
Automatic view generation (perspective and orthographic)	Impossible	Impossible	Possible
Cross-sectioning	Manually guided	Manually guided	Possible, even automated cross-hatching is possible
Elimination of hidden details	Manually guided	May be possible	Possible
Analysis functions (geometric calculations)	Difficult or impossible	Difficult or impossible	Possible
Numerical control application	Difficult or impossible	Automatic possible	Automatic possible

4.3 GEOMETRIC CONSTRUCTION METHODS

A technical (production) drawing consists of a lot of information besides the simple view or the geometric representation. Examples are the dimensions, tolerances, material specifications, the processing requirements, assembly requirements, etc. Many alternative views of the details in the object should be presented for clarity. A sample illustration of the technical drawing produced using a drafting software (AutoCAD) is shown in Fig. 4.7.

It is incumbent on the geometric construction method employed to make use of the normal information available at the product-design stage and also be as simple as possible in construction. In addition, the current-day interactive interfaces provided between the software and the user minimise the hassles associated with the geometry input.

The three-dimensional geometric construction methods, which extend from 2D, that are normally used are

- linear extrusion or translational sweep, and
- rotational sweep.

In addition, the 3D solid-generation method includes the primitive instancing or constructive solid geometry that involves the direct use of 3D solids for modelling.

4.3.1 Sweep or Extrusion

In linear extrusion, initially a two-dimensional surface is generated (Fig. 4.8) and then swept along a straight line, thus generating three dimensions. It is possible to repeat the same technique for generating reasonably complex geometry. The sweep direction can be any three-dimensional space curve and need not be a straight line.

One of the reasons why sweep is the most used geometric construction in spite of its limitations is that it is the natural extension in which designers or draftsmen work. As a result, further variations in sweep can be made available for generating more complex geometry. For example, it is possible to sweep in a linear

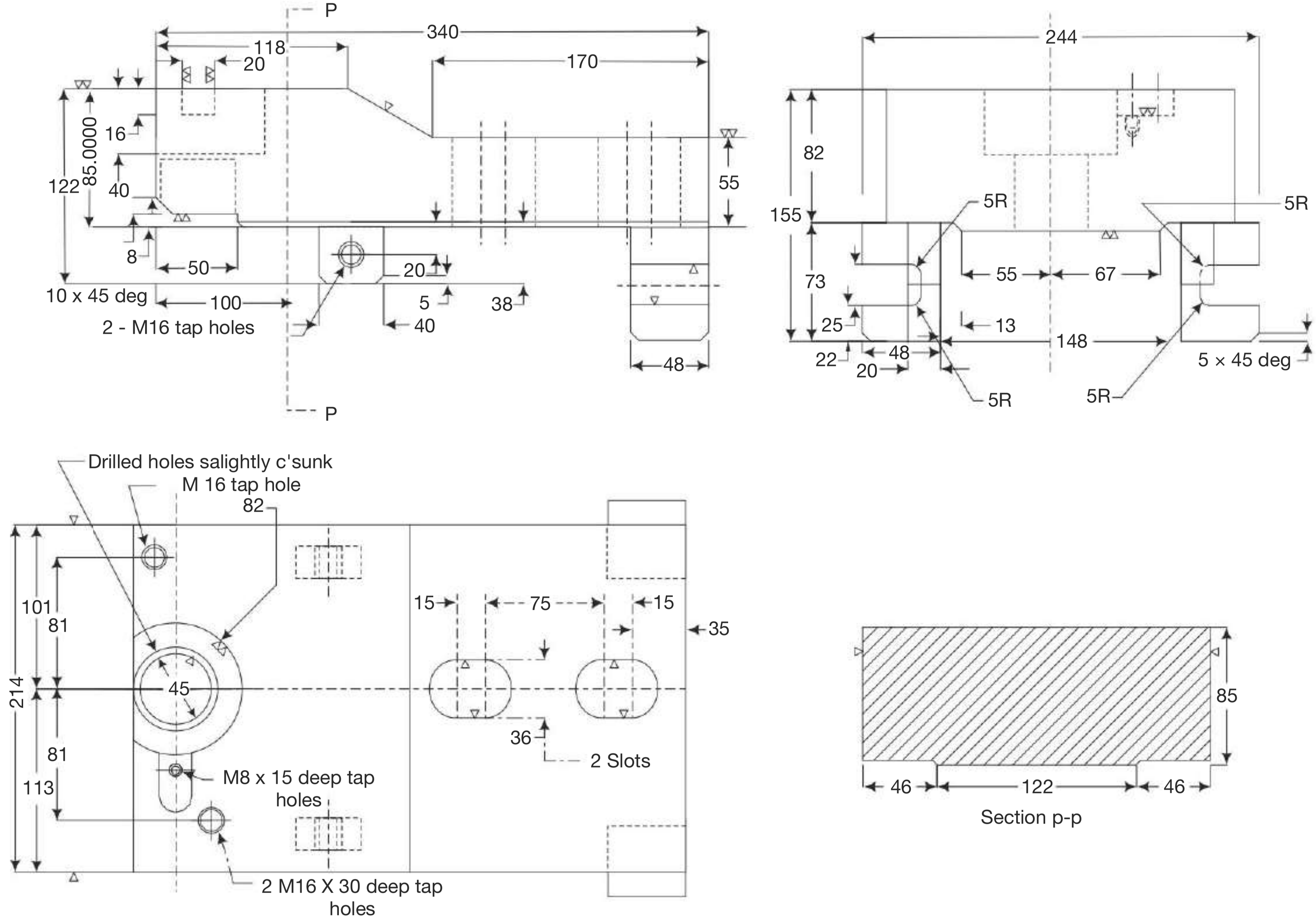


Fig. 4.7 *Information present in a 2D drawing (drawing produced using AutoCAD)*

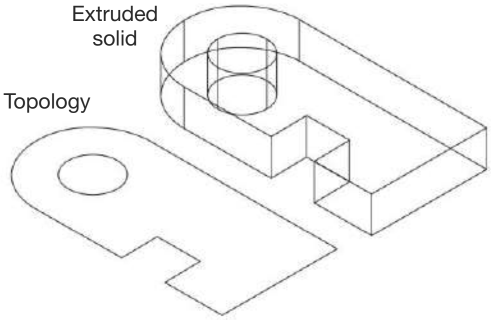


Fig. 4.8 *Component model produced using translational (linear) sweep (extrusion)*

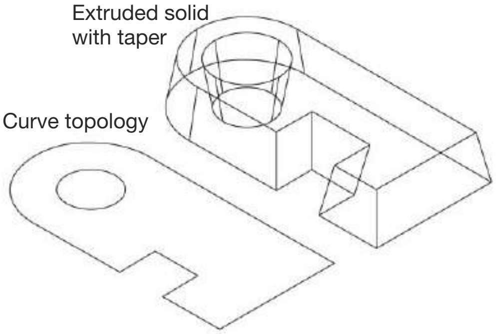


Fig. 4.9 *Component model produced using translational (linear) sweep with the taper in sweep direction*

direction with a taper along the sweep direction. This is quite useful for generating the die and mould surfaces for providing the draft angle.

Another possibility is to sweep a profile through a three-dimensional direction instead of the simple straight line to get a more complex surface as shown in Fig. 4.10.

A further advantage of sweep is that in view of the varied facilities available normally in the two-dimensional modellers, they can also be utilised for modelling three-dimensional solids. There are some places where sweep may fail to generate a solid. For example, as shown in Fig. 4.11, the extending line in the topology when swept will generate only a hanging edge and not a solid. Hence, the sweep operator will have to inhibit such cases by imposing some restrictions on the topology chosen for extrusion. Generally, the topology should be closed with no overhanging edges to produce a solid by linear extrusion.

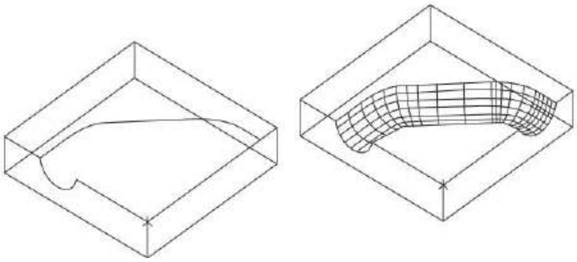


Fig. 4.10 *Component model produced using linear sweep with the sweep direction along a 3D curve*

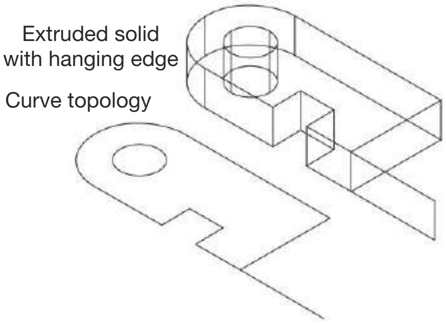


Fig. 4.11 *Component model produced using translational (linear) sweep with an overhanging edge*

Also, not all types of objects can be modelled by this technique in addition to the problems associated with it as explained above. It is not possible to add details in planes parallel to the sweeping direction by the sweeping technique alone. This technique is, therefore, used along with other modelling methods.

Another type of construction technique is the *rotational sweep*, which can be utilised only for axi-symmetric jobs as shown in Fig. 4.12. This type is used for all axi-symmetric components such as bottles used for various applications.

It is also possible to add twist to the sweeping in the third dimension. Similarly, the rotational sweep can be enhanced by the addition of axial and/or radial offset while sweeping to get helical or spiral objects. In CAM, sweep can be used in the material-removal operations to calculate the tool paths. The volume swept by the tool when subtracted from the blank will generate the final shape required.

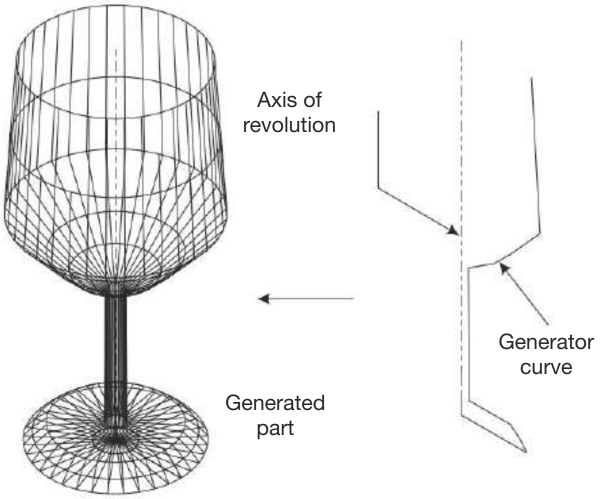


Fig. 4.12 *Component produced by the rotational sweep technique*

4.3.2 Solid Modelling

Though the above techniques are capable of generating surfaces that are reasonably complex, they are not suited for the purpose of inputting geometry. The best method for three-dimensional solid construction is the *solid modelling* technique, often called *primitive instancing* or *constructive solid geometry* (CSG). In this, a number of 3-dimensional solids are provided as primitives.

Some typical primitives utilised in the solid modellers are shown in Fig. 4.13. Though these are the analytical solid primitives generally used, the modelling is restricted only to these. It is possible to use any solid obtained by the other modelling tools in the system as a primitive. From these solid primitives, the complex objects are created by adding or subtracting the primitives.

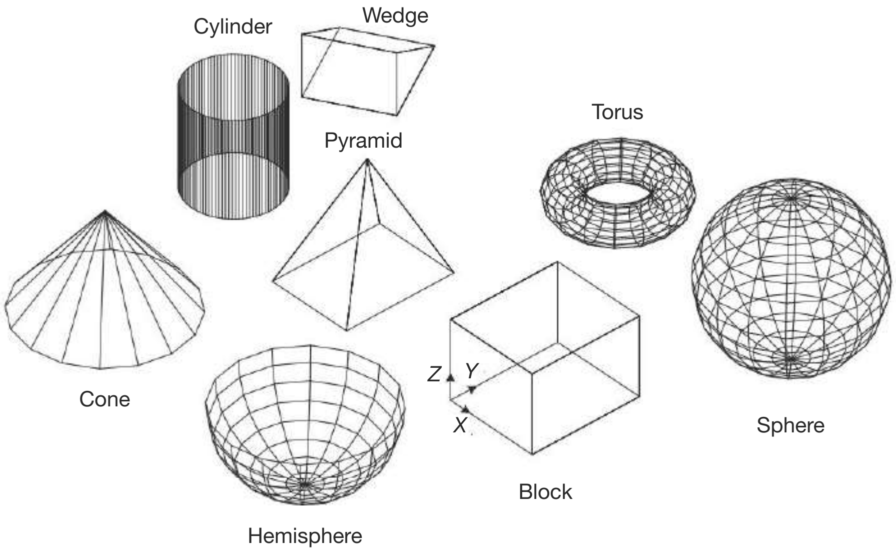


Fig. 4.13 *Various solid-modelling primitives*

For combining the primitives to form the complex solid, basic set operators, also called *Boolean operators*, are used. They are of the following types—union, intersection and difference. The effect of these operators on the primitives is shown in Fig. 4.14 for the simple case of a block and cylinder shown in their 2-dimensional relationship. However, the same is true for 3-dimensional as well as for any other complex orientation of the primitives and for any size as shown in Fig. 4.15. It may be seen that this is a very powerful technique for representing fairly complex objects with relative ease. A job, as generated with these two primitives, is shown in Fig. 4.16 wherein the entire operators and their effect is shown.

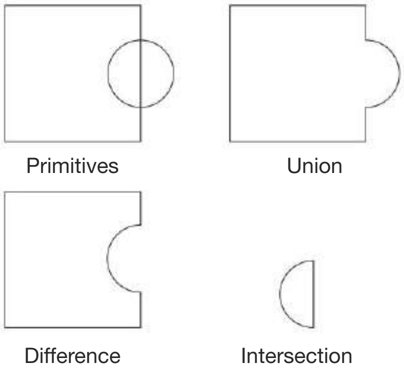


Fig. 4.14 *Boolean operators and their effect on model construction*

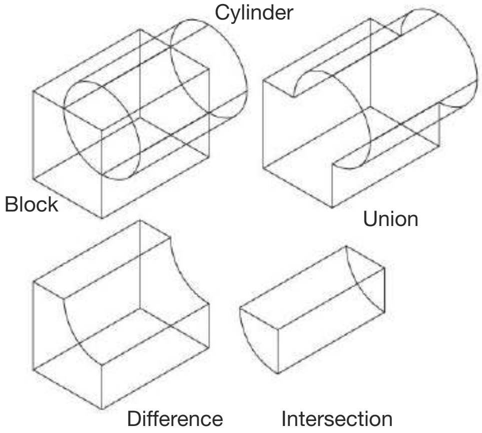


Fig. 4.15 *Boolean operators and their effect on model construction*

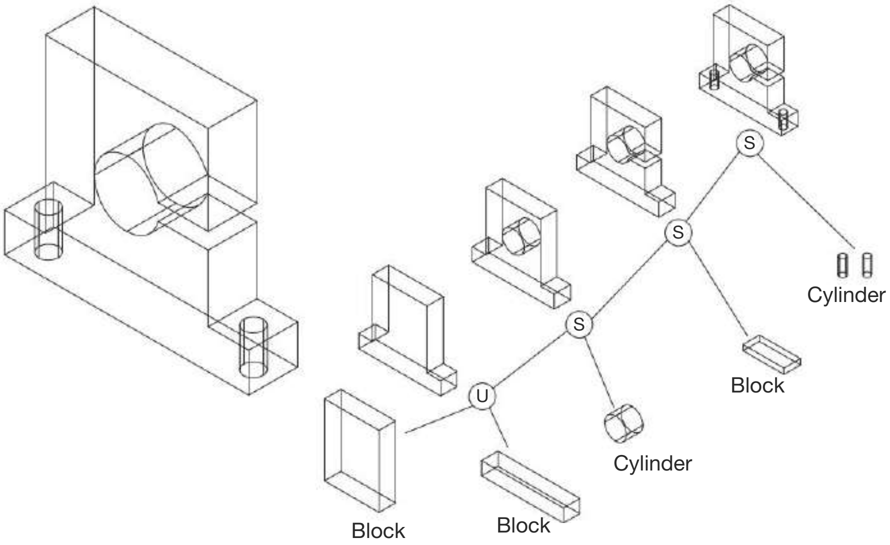


Fig. 4.16 *Creating a solid with 3D primitives in solid modelling and the model shown in the form of Constructive Solid Geometry (CSG)*

From Fig. 4.14, also to be noted is the way the operators affect the geometry generated. The storage of data required for the complex job is only the construction tree of the operators and the relevant dimensions of the primitives. This facilitates the reduction of the storage requirement. Also, by making modifications to the CSG tree, a new object can be obtained at any time. The Boolean operators always guarantee that the object formed by these rules is physically realisable. However, if a CSG tree is evaluated for functions other than modelling, enormous amount of computational overheads are caused. Therefore, most of the modelling systems store, along with the CSG representation, the boundary representation (B-rep), which is essentially the evaluated boundaries of the solid model stored in any parametric form. The most common method of storage used is the Non-Uniform Rational B-splines (NURBS), and the method of modelling is implemented in hardware form by many of the high-end graphic adapter cards.

The primitives are normally stored internally using analytical representation. However, non-analytical surfaces such as Bézier surfaces cannot be modelled using CSG representation. Also, special facilities have to be developed for joining the sculptured surfaces with the other solids. Hence, separate representations for these surfaces are to be maintained in the design system.

4.3.3 Free-Form Surfaces

There may be surfaces which cannot be defined by any of the analytical techniques available. A few examples are car bodies, ship hulls, some die-cavity surfaces, and decorative surfaces styled for aesthetics. An example of sculptured surface is shown in Fig. 4.17. The only way these surfaces can be modelled is through a series of control points and other boundary conditions, which specify the nature of the surface desired. There are a large number of numerical techniques available in such cases. Some of the types of surfaces that are normally employed in CAD systems are ruled surfaces, Bézier surfaces, B-spline surfaces, and NURBS.

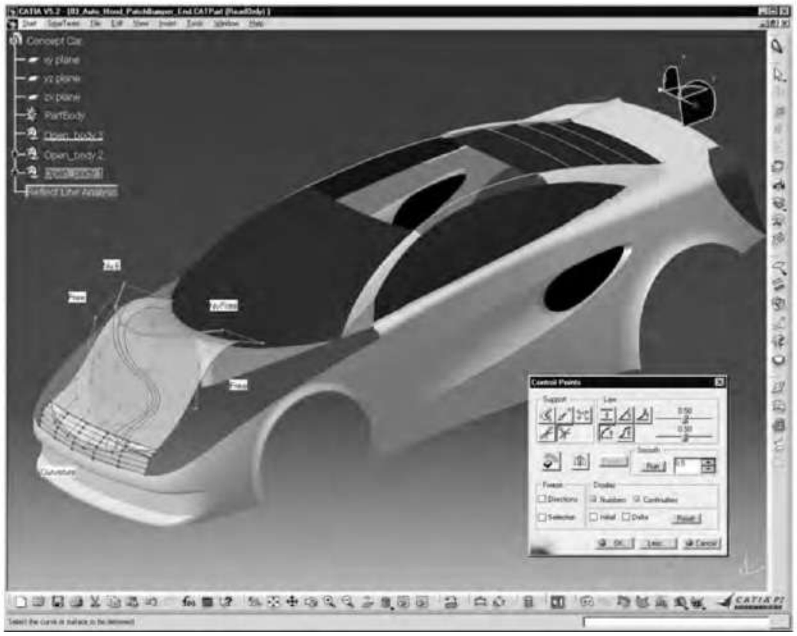


Fig. 4.17 *Model generated using the sculptured surfaces (image appears with the permission of IBM World Trade Corporation/Dassault Systems—Model generated using CATIA)*

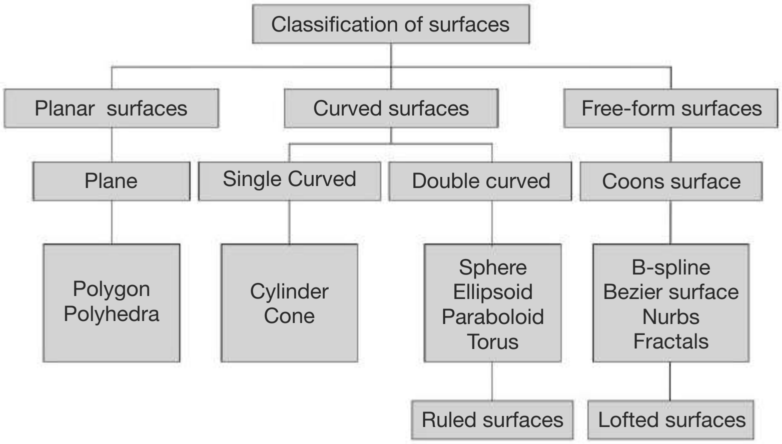


Fig. 4.18 *The various types of surfaces used in geometric modelling*

The various surface types are shown in Fig. 4.18. Some of these surfaces can be modelled using analytical methods. The free-form surfaces need to be defined using mathematical techniques that require a large amount of computation to evaluate and extrapolate the surfaces.

1. Planar Surface A planar surface is a flat 2D surface.

2. Curved Surface The two types of curved surfaces are the following:

(a) Single Curved Surface It is a simple curved surface. For example, it can be generated by a straight line revolved around an axis, such as cylindrical surface. Some other examples are conical surfaces, and surfaces of pyramids, prisms and conics. This surface can be developed using geometric techniques.

(b) Double Curved Surface It is a complex surface generated not by a straight line but by a curved surface. This surface is not developable. Some examples of these surfaces are spherical, torus, ellipsoid and hyperboloid. These surfaces are used for airplane fuselages, spherical fuel tanks, automobile bodies and ship hulls.

Some of the construction methods employed for these surfaces are explained below:

Ruled Surface A ruled surface is a surface constructed by transitioning between two or more curves by using linear blending between each section of the surface. An example of a ruled surface is shown in Fig. 4.19. The selection of the starting point on the curve for the surface determines the actual surface obtained. This is shown in Fig 4.20 where the two different surfaces were obtained from the same data points based on the sequence in which the points were selected. A planar surface can be obtained by forming a ruled surface

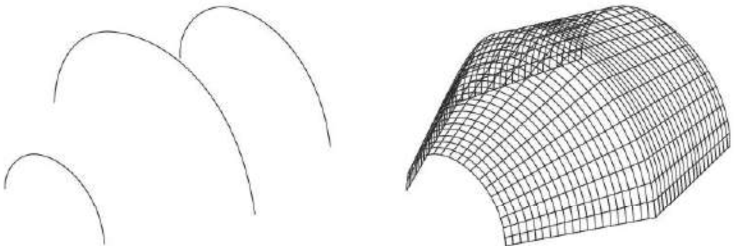


Fig. 4.19 *Ruled surface on the left shows the curves from which the ruled surface on the right is formed*

between two straight lines as shown in Fig. 4.21a. Similarly, cylindrical and conical surfaces can be obtained by taking appropriate closed curves as shown in Fig. 4.21a and b.

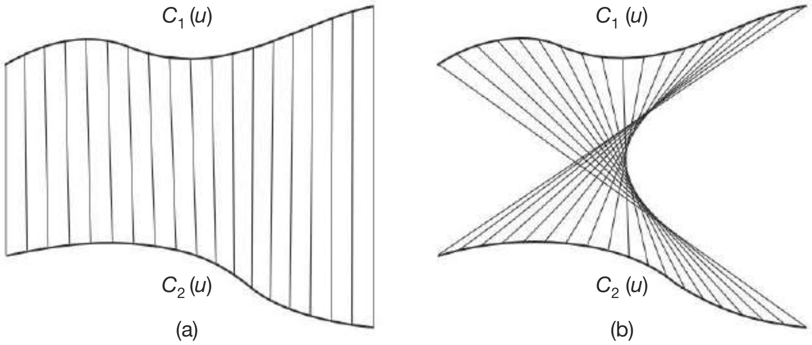


Fig. 4.20 *Two different ruled surfaces formed between two space curves $C_1(u)$ and $C_2(u)$.*

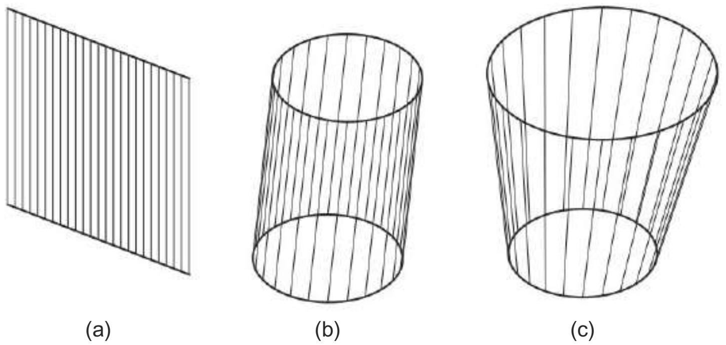


Fig. 4.21 *Three different ruled surfaces: (a) Planar surfaces between two straight lines (b) Cylindrical surface between two identical circles (c) Conical surface between two circles of different diameters*

Fig. 4.22 *Different ruled surfaces obtained from the same closed curves. The different surfaces resulted because of the differences in the selection of the starting point on each of the curves*

Coons Surface A Coons surface or patch is obtained by blending four boundary curves. The single patch can be extended in both the directions by adding further patches. The blending of these patches can be done either by means of linear or cubic blending functions, thereby giving rise to a smooth surface linking all the patches. An example is shown in Fig. 4.24 where the number of curves in both the directions are shown in (a) which are then used to generate the Coons surface shown in (b). The main advantage of a Coons surface is its ability to fit a smooth surface through digitised points in space such as those used in reverse engineering.

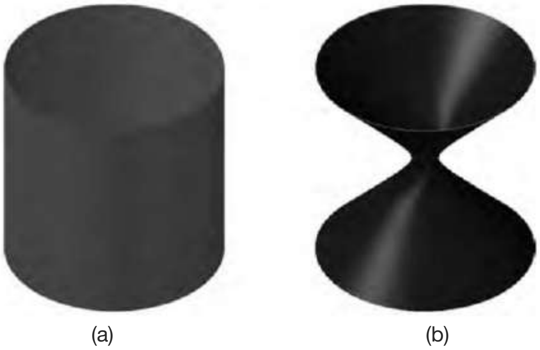


Fig. 4.23 *Different ruled surfaces obtained from the same closed curves (circles): (a) Cylindrical surface (b) Hyperboloid*

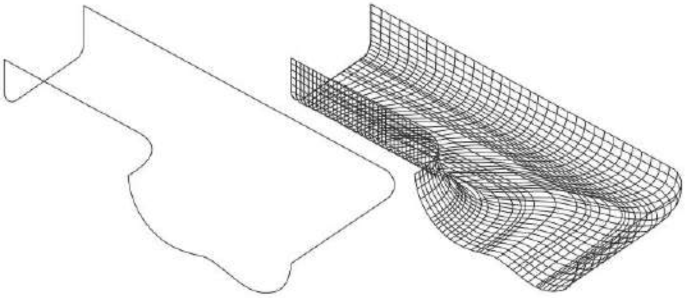


Fig. 4.24 *Coons surface generation*

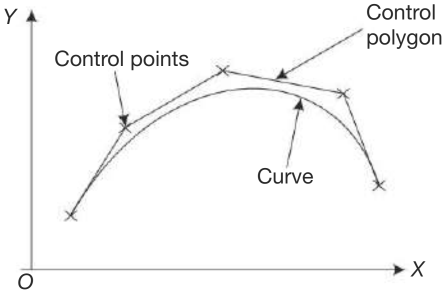


Fig. 4.25 *Bézier curve and the associated control polygon*

Bézier Curves To obtain a more free-form design for aesthetic surfaces that satisfy some requirements, the modelling techniques need to provide more flexibility for changing the shape. This can be achieved by the use of Bézier curves named after P Bézier, the designer of the French car company Renault, who invented the procedure in the 1960's.

A Bézier curve uses the vertices as control points for approximating the generated curve. The curve will pass through the first and last point with all other points acting as control points. An example is shown in Fig. 4.25 showing the control polygon and the generated smooth curve. Some examples of different Bézier curves are shown in Fig. 4.26, which will depend upon the nature of the control polygon vertices.

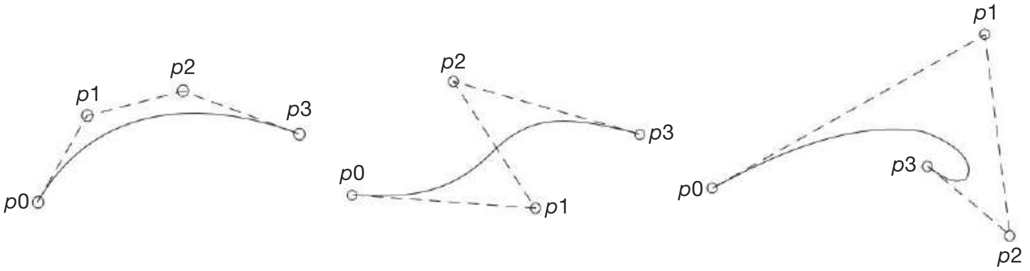


Fig. 4.26 *The various examples of Bézier curves depending on the associated control polygons*

The flexibility of the process can be seen by changing the position of the individual control points in space, thereby altering the control polygon. Changing the second point in space, alters the curve as shown in Fig. 4.27. This is a very flexible process and is widely used for the design of aesthetic surfaces. The flexibility of the curve increases with more control points. The process can be extended for surfaces as well.

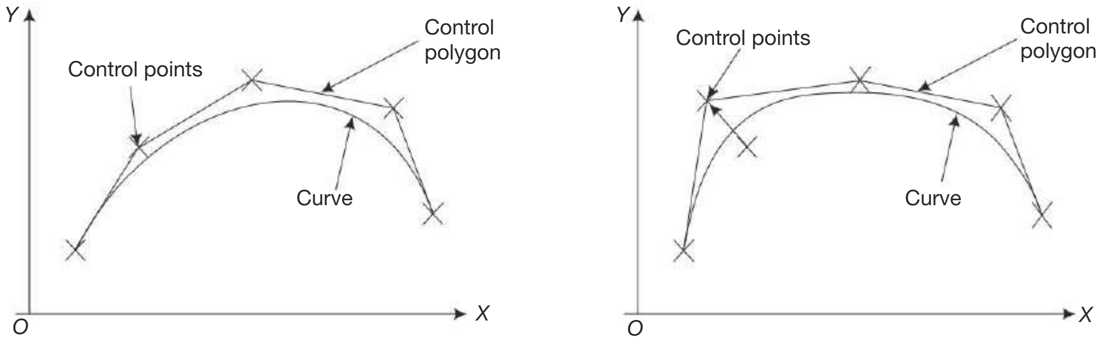


Fig. 4.27 *Modification of* Bézier *curve by tweaking the control points*

B-splines In the case of Bézier curves, it is considered as a single curve controlled by all the control points. As a result, with an increase in the number of control points, the order of the polynomial representing the curve increases. To reduce this complexity, the curve is broken down into more segments with better control exercised with individual segments, while maintaining a simple continuity between the segments. An alternative is to use a B-spline to generate a single piecewise parametric polynomial curve through any number of control points with the degree of the polynomial selected by the designer (Fig. 4.28). B-splines exhibit a local control in that whenever a single vertex is moved, only those vertices around that will be affected while the rest remains the same.

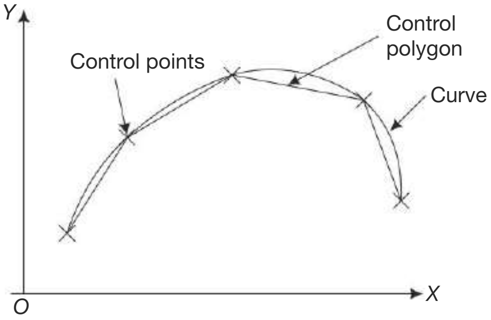


Fig. 4.28 *Spline curve*

The B-spline surface thus combines the strengths of Coons and Bézier surface definitions by forming the surface from a number of controlling curves similar to Coons while having the control points to alter the surface curvature.

Uniform cubic B-splines are curves with parametric intervals defined at equal lengths. The most common scheme used in all the CAD systems is the non-uniform rational B-spline (commonly known as NURB), which includes both the Bézier and B-spline curves. It also has the capability to represent a wide range of shapes including conics. NURBS is a rational surface with each point on the surface having a weight associated with it. NURBS is similar to the B-spline surface with the addition of the weighing factors, which can have any value, thus allowing for the necessary surface-design alteration possibilities. When all the weighing factors become 1, then it becomes a B-spline surface.

Lofted Surface A lofted surface is a surface constructed by transitioning between two or more curves similar to a ruled surface by using a smooth (higher order) blending between each section of the surface, as shown in Fig. 4.29. This method is extremely useful for modelling engine manifolds, turbine blades, airframes, volute chambers and the like. Such solids are otherwise difficult to produce by any of the previously described modelling methods.

Surface Fillet This is the ability to automatically generate the fillet radius between two urfaces, either analytical or sculptured. The radius could be uniform or vary linearly, depending upon the meshing surfaces. If the same fillet is to be generated through the solid-modelling approach, a large amount of input is needed in terms of the difference of a number of solids that would form a regular fillet surface. An example is shown in Fig. 4.30—the left is without a blend and the right is the same object with a uniform blend radius. Modern CAD systems have the capability to blend even complex surfaces, thereby relieving the designer of many unwanted design operations.

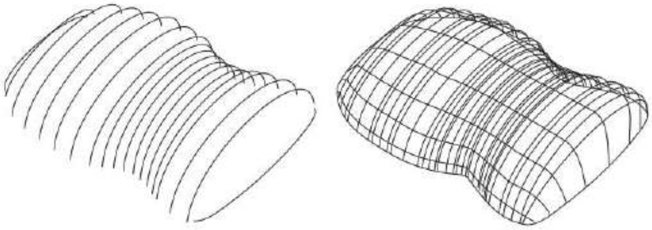


Fig. 4.29 *Lofted surface*

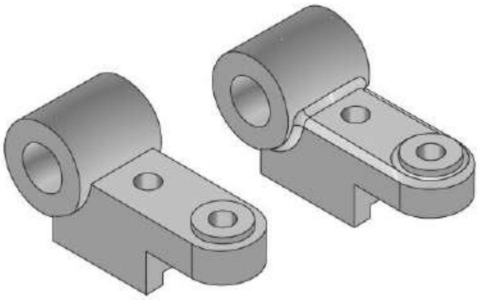


Fig. 4.30 *Example of filleting or blend method for model generation*

Tweaking This is the ability to alter a model already created using any of the earlier approaches described. In this, a face or a vertex in the model is interactively moved to see the effect in the modification of the geometry. For example, in the case of sculptured surfaces, a point on the surface can be changed, and the

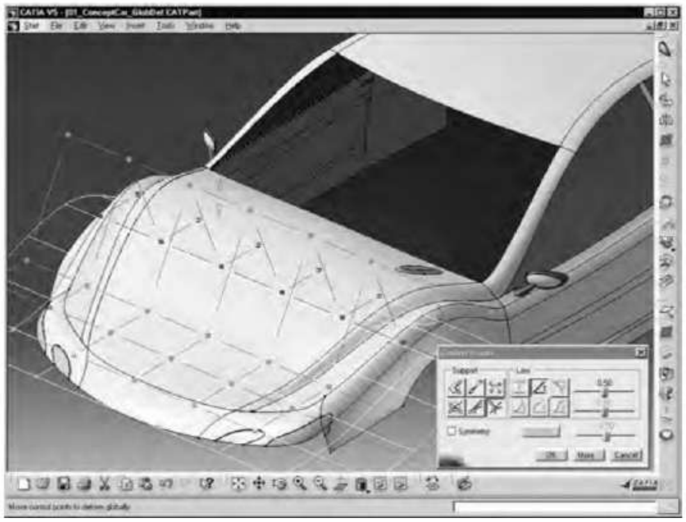


Fig. 4.31 *Example of tweaking method for surface modification (image appears with the permission of IBM World Trade Corporation/Dassault Systems—Model generated using CATIA)*

surface modification observed for the purpose of styling (Fig. 4.31). However, when tweaking is applied to CSG models, certain problems are involved. Since the CSG representations retain the way the job was modelled using the primitives, any tweaking done may have unpredictable results because the effect obtained depends on how the original model was developed.

4.4 CONSTRAINT-BASED MODELLING

Many of the modelling methods outlined earlier rely to a great extent on the geometry of the part. However, when engineers design parts, geometry is only one aspect of it, while the physical principles governing the operation of the part are the main reason for getting the required geometric features. As a result, the modelling process is driven by the design process. The constraint-based modelling, parametric modelling or variational modelling was developed in the 80's by Pro Engineer and later continued by other modellers (for example, Solidworks and Autodesk's Inventor) using the sweep or extrusion technique that we have discussed earlier. They combine all the modelling techniques described thus far in such a way that the modelling process embeds the design intent to the maximum extent possible.

In this type of modelling, often called parametric modelling or constraint-based modelling, most of the time the modelling starts with a sketch in 2D plane and then is swept along a specified direction, thereby producing the desired component. We will discuss this method a little more in detail since this is the mainstay of most of the current modellers. The other facilities that we have described are also retained so as to provide a complete modelling methodology.

The starting point for constraint-based modelling is a 2-dimensional sketch, where the approximate shape of the object is created in a given two-dimensional plane such as the *XY*, *YZ* or *ZX* planes. All the facilities that are required are readily available along with a lot of intelligence in the sketching process. Initially, the user tries to develop the approximate shape without worrying about the dimensions (Fig. 4.32). Later when the shape is completed, the user starts inputting the dimensions and gives some constraints such that the sketch is uniquely defined (Fig. 4.33). Some of the constraints that are used are parallel, perpendicular, coincident, horizontal, concentric, tangential, etc. The dimensions can be given by simple values or can be given in terms of some equations that represent some design requirement. The available equation capability depends on the individual modeller. Complete rules for solving the sketch are embedded in the modeller such that it guides through the sketch constraining process.

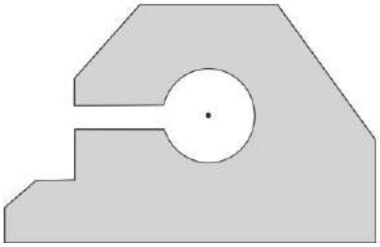


Fig. 4.32 *Example of initial sketch without any dimensions*

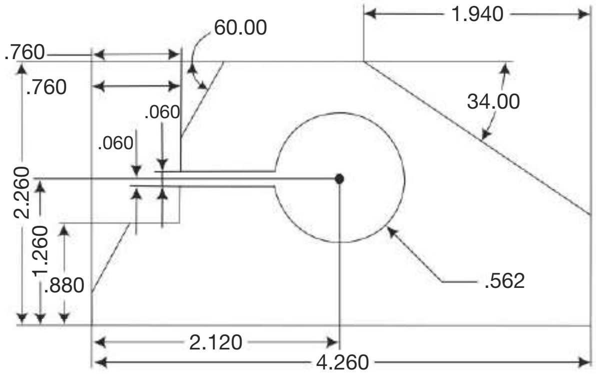


Fig. 4.33 *Sketch which is fully constrained and dimensioned*

The dimensions shown in Fig. 4.33 can be direct numerical values, or they can be driven by other dimensions in the form of equations. Facilities are also available whereby these dimensions can be in the form of equations with variables. The variables need not be dimensions but may be other physical properties of the material or phenomenon that is taking place. That makes the modelling process to be directly linked to the design process.

The constrained sketch shown in Fig. 4.33 is then swept along a linear direction that is perpendicular to the sketch plane producing a solid, as shown in Fig. 4.34. This is a simple operation and easy to understand as a starting point for the modelling process. Though the example shown involved a two-dimensional sketch and a fixed sweeping direction perpendicular to the sketch plane, it is possible to do the same process using a 3-dimensional sketch and a 3-dimensional sweeping curve which is used for specific modelling applications.

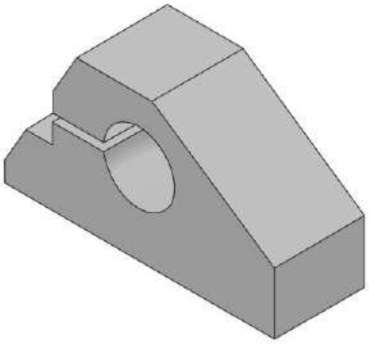


Fig. 4.34 *The sketch in Fig. 4.33 when swept along a linear path produces the solid*

Having obtained the base solid as shown in Fig. 4.34, the next step is to identify the sequence of features to be modelled and carry out similar operations. For this purpose, it is necessary to identify individual planes for each of the sketching operations. These can be identified as any of the existing planar faces of the solid already created, or some other plane, which has a certain relation to any of these planes, for example, a plane that is passing through the centre of any given face and parallel to any one of the principal planes. Similarly, it could be a plane that is tangential to any cylindrical surface in the solid, or a plane offset from a given plane. The bottom side of the part is therefore chosen as the plane and the sketch geometry is made as shown in Fig. 4.35. This geometry is then extruded and cut from the solid to produce the required shape as shown in Fig. 4.36.

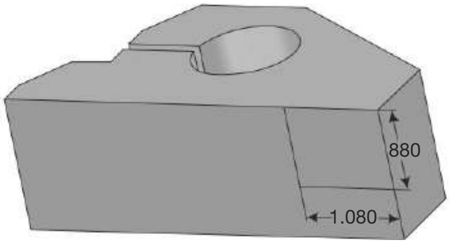


Fig. 4.35 *The sketch for the new feature (a cut)*

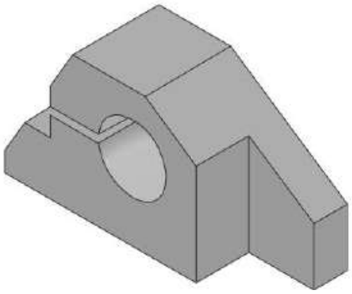


Fig. 4.36 *The solid after executing an extruded cut of the geometry in Fig. 4.35*

Similar processes are continued to make all the necessary features—two holes in different planes and a counter-bored hole are made resulting in the final solid as shown is obtained in Fig. 4.37. The modelling process that resulted in the solid is maintained in the form of a model tree as shown in Fig. 4.38. It is possible to go back and edit any of the stages in the modelling process to carry out the necessary design changes. The model automatically gets updated with all the details as they get affected by the prior editing changes.

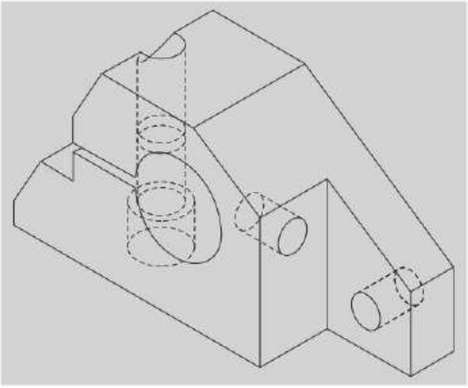
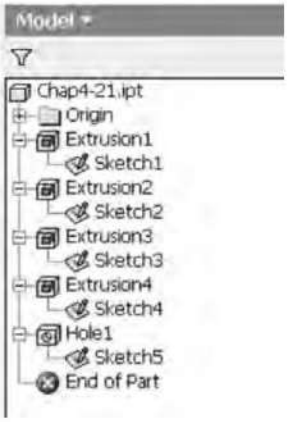


Fig. 4.37 *The final solid*



Model ▾

Chap4-21.ipt
- Origin
- Extrusion1
  - Sketch1
- Extrusion2
  - Sketch2
- Extrusion3
  - Sketch3
- Extrusion4
  - Sketch4
- Hole1
  - Sketch5
- End of Part

Fig. 4.38 *The model tree of the part showing the modelling process*

As explained earlier, feature-based modelling tries to capture the design intent such that any person other than the original designer is able to modify the model and incorporate into it other designs as required. This requires a careful analysis of the part to be done in order to evaluate the individual features and the sequence in which the features are added to the base model to achieve the final model. It is possible that the change in the sequence of design alters the final result. This can be explained with an example as follows. In Fig. 4.39, a box is first created, followed by the two holes as shown. The final operation done on them is the shell, which removes the bulk of the material and maintains a uniform thickness of material all along the surfaces. Figure 4.40 shows the result of interchanging the shell and hole operators in the sequence.

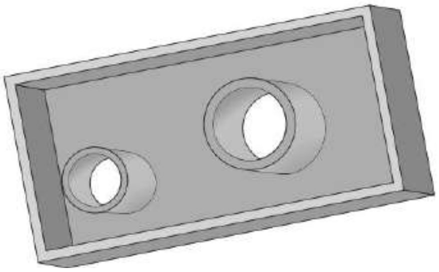


Fig. 4.39 *A geometric model created following the sequence of features as Box → Hole → Shell*

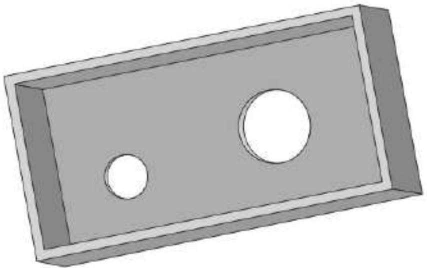
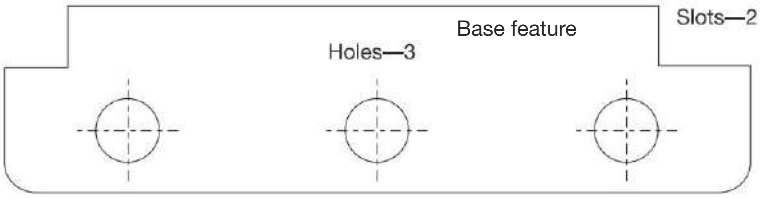


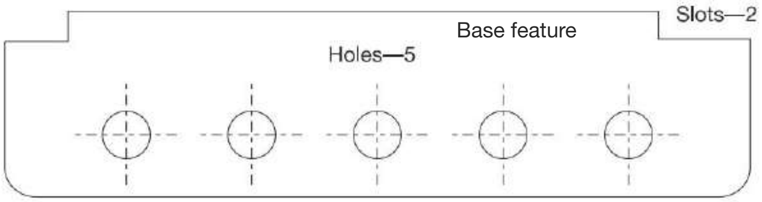
Fig. 4.40 *A geometric model created following the sequence of features as Box → Shell → Hole*

The main advantage of the use of features is that it is possible to modify the geometry by picking up the individual feature and changing the attributes. The system will automatically generate the complete new geometry with the modified attributes. Figure 4.41a shows the original model which consists of three features, the base feature which is a flat plate, two slots at two top corners, and five holes with a given diameter and position with respect to the centre of the plate. The same is modified simply in (b) by modifying the dimensions of the slots, and the diameter and number of holes.

Actual details of the modelling methods are given in Chapter 7 using Autodesk Inventor.



(a) Original model



(b) Modified model

Fig. 4.41 *Feature-based model and its modified form*

4.5 OTHER MODELLING METHODS

In addition to the geometric construction methods mentioned above, a number of other schemes are also used for specific applications depending on the requirements of the particular problem.

4.5.1 Cell Decomposition

Another form of solid modelling is that in which the objects are represented as a collection of arbitrarily shaped 3D primitives. The individual cells can be defined as a set of parameterised cell types that can even have curved boundary surfaces. It consists of a single operator, ‘glue’. This operator restricts the cells to be nonintersecting, which means adjoining cells may touch each other but must not share any interior points. Complex solids are then easily modelled by joining the simple cells using the glue operator. Because of the single operator, the modelling process is easy but sometimes tedious. Also, cell decomposition is not as versatile as CSG. It is also possible to generate a solid through the subdivision process instead of the bottom-up fashion by using the parameterised primitives. This process is generally useful for finite-element mesh generation.

4.5.2 Variant Method

In this form of representation, the complete part is located in the memory in the form of a sample drawing without the dimensions. The job is identified by means of a part code similar to the group technological code described later. The skeleton part drawing would be displayed on the CRT screen as shown in Fig. 4.42. The user then has to simply fill in the blank dimensions as directed by the program. This approach is very convenient when one is interested in a group of similar components to be made or for making very small corrections in only their size and not shape. After the input of the dimensions on the menu drawing, the actual proportionate drawing is generated. Thus, it has a very limited application if a complete modelling system were so written. However, the general modelling systems could be enhanced for special applications of the

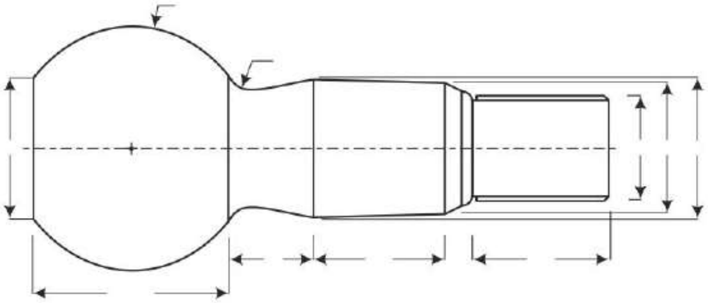
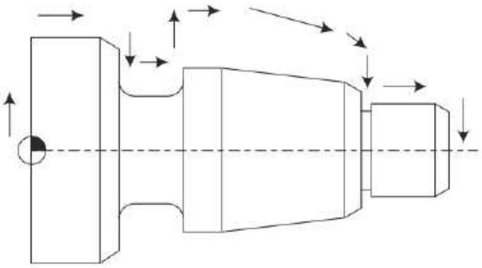


Fig. 4.42 *Typical drawing for the variant method of modelling*

company by making special macros within the modelling system using the programming languages available in the system.

4.5.3 Symbolic Programming

Another very interesting approach adopted by FANUC is the input of part drawings through a series of symbols, which show the direction of movement in the SYMBOLIC FAPT TURN system. As shown in Fig. 4.43, the keyboard consists of a number of direction keys for drawing lines, circles and special features like chamfers, corners, and threads. The operator starts inputting the drawing from one end showing the direction in which the part contour moves, as explained in Fig. 4.44, for a sample drawing. When the operator presses any of the special keys, a series of questions relevant to that key in terms of lengths and diameters of the component are asked and their answers can be input by the operator directly from the part print. This method is suited only for CNC part programming of turned parts of the 2C type.



Composed part

T			
C	↖	↑	↗
R	←		→
G	↙	↓	↘
N	⤷		⤶

Symbol keys

↑	→	C	↓	⤷	→	⤷	↑	→	↘	↘	↓	G	→	C	↓

Key sequence

Fig. 4.43 *Part model produced using symbolic programming*

4.5.4 Form Features

All the modelling methods described so far are based on geometry. A feature is a geometric shape specified by a parameter set which has special meaning for design or manufacturing engineers. Features represent a collection of entities in an intelligent form (like ‘hole’, ‘slot’, ‘thread’, ‘groove’, etc.) and hence provide information at

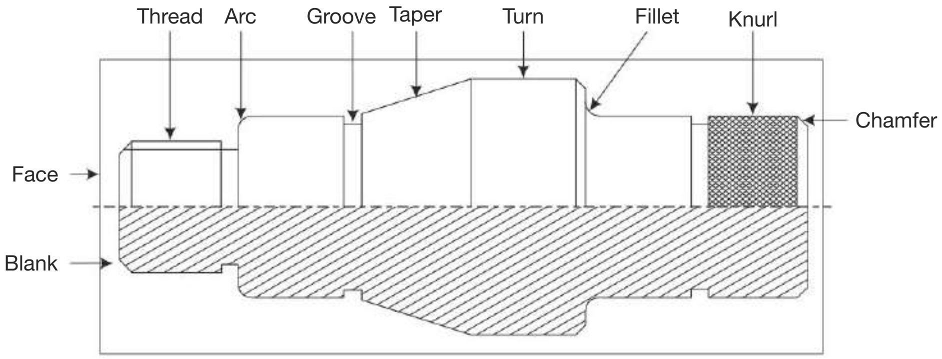


Fig. 4.44 *Examples of form elements used for model generation in the case of axi-symmetric components*

a higher conceptual level. The use of such groups of geometry coupled with the necessary information needed for other applications such as process-planning functions is seen as a practical means of linking the design and manufacturing.

In this type of modelling system, all features that are available for a particular type of job are catalogued into what are called ‘features’ or ‘form elements’. The actual job is therefore treated as an assembly of these form elements properly dimensioned in the correct order. The complete system is highly interactive with proper graphic menus made available. A typical axi-symmetric component is shown in Fig. 4.44 with the features from which it is modelled, being labelled.

Typical form features useful for modelling a majority of axi-symmetric components with all the additional milling features such as holes, slots, etc., are shown in Fig. 4.45. Here, the user starts from one end to assemble the features as required in the final component as shown in Figs 4.46 and 4.47.

It is possible to incorporate any type of features, both external and internal. Though it is most convenient for turned parts, a number of such systems have been designed and are in continuous use. It is possible to incorporate into the modelling system, such features as are not essentially axi-symmetric but are found on these components. Examples are keyways, flats, pitch circle drilling on the flange, gears, etc. A majority of the high-end CAD/CAM systems provide support for a variety of features, but mostly from the geometric viewpoint only.

The major advantage of this kind of modelling is that the feature elements are more convenient for the people involved in the manufacturing process than the geometric elements, and the modelling process is generally faster. Hence, macro languages available in the geometric modelling systems are sometimes used to generate a library of these ‘features’ and provided to the user as menu for easier modelling.

A further advantage of this method of modelling is that it can be easily used to extend to the technological processing functions such as identifying the manufacturing requirements, process planning and CNC part-program generation.

4.6 WIREFRAME MODELLING

As explained earlier, wireframe modelling is the oldest form of modelling objects in 2D and 3D. These modelling methods rely on the ability of the user to define the basic elements such as points, lines and circles to be defined in a number of ways.

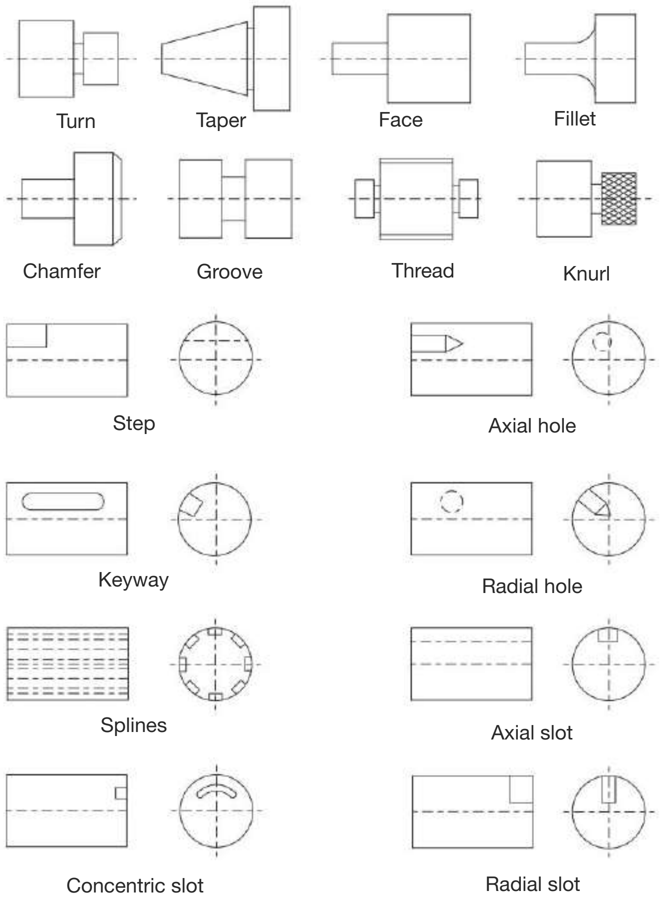


Fig. 4.45 *Examples of form features for modelling axi-symmetric components with milled features*

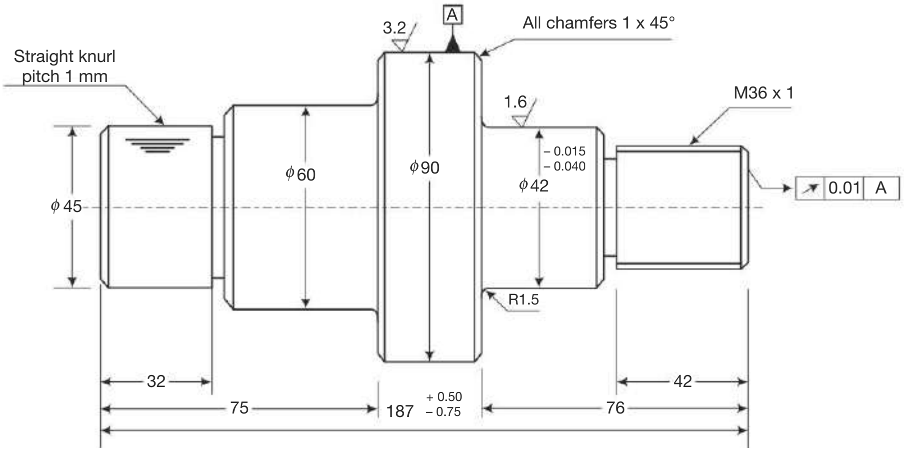
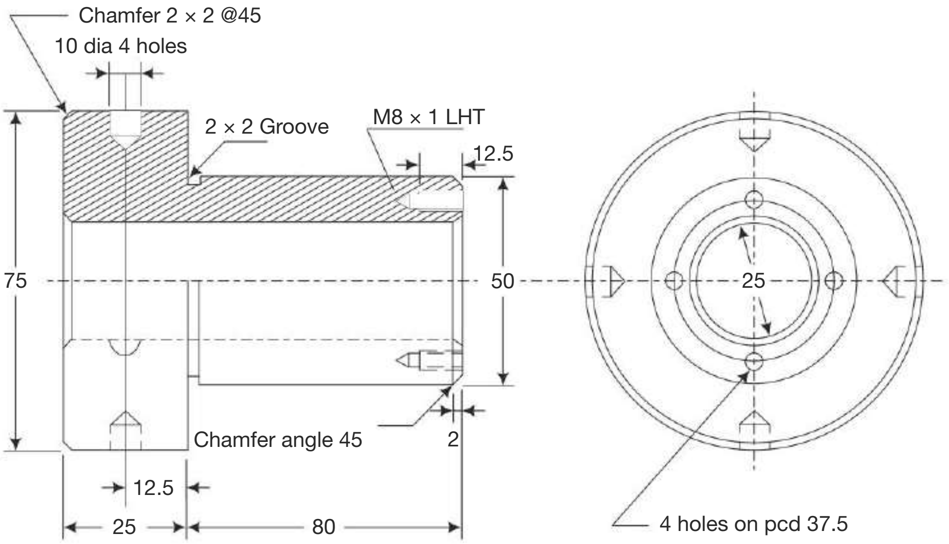


Fig. 4.46 *Example of a modelled component using the features shown in Fig. 4.45*



Sectional elevation

End view

Fig. 4.47 *Example of a modelled component using the features shown in Fig. 4.45*

4.6.1 Point Definitions

A few methods of defining point objects are shown in Fig. 4.48. All the points are defined using rectangular Cartesian coordinates. As shown in Fig. 4.48a, the simplest method of defining a point is by specifying the Cartesian coordinates of the point. Alternatively, these points could be from the existing points or in some relation to the existing objects, such as the centre of a circle or a midpoint of a line. Figure 4.48b shows

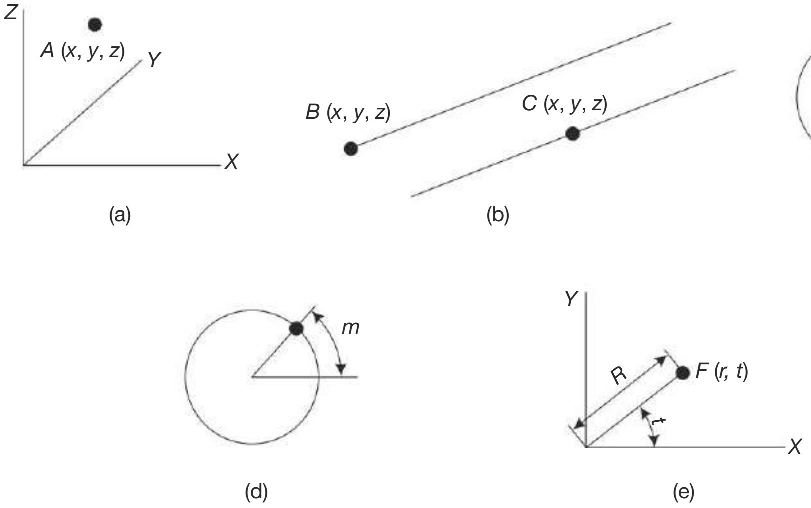


Fig. 4.48 *Point definitions*

the point as part of an existing line. It can be defined as an endpoint of a line as *B* or as a midpoint as *C*. Figure 4.48c shows a point as the centre of an existing circle. Figure 4.48d shows a point on a circle at an angle. Figure 4.48e shows a point using the polar coordinates. It is possible to have similar definitions with variations depending upon the type of existing objects.

4.6.2 Line Definitions

A few methods of defining line objects are shown in Fig. 4.49. As shown in Fig. 4.49a, the simplest method of defining a line is by means of its two endpoints. The endpoints can be explicitly specified by their Cartesian coordinates as shown. Alternatively, these points could be from the existing points or in some relation to the existing objects, such as the centre of a circle or a midpoint of a line. Figure 4.49b shows a line parallel to an existing line and passing through a point *C*. Figure 4.49c shows a line perpendicular to an existing line and passing through a point *D*. Figure 4.49d shows a line parallel to an existing line at a distance. Figure 4.49e shows a line from a point *E* and tangential to an existing circle. Figure 4.49f shows a line tangential to two existing circles. It is possible to have similar definitions with variations depending upon the type of existing objects.

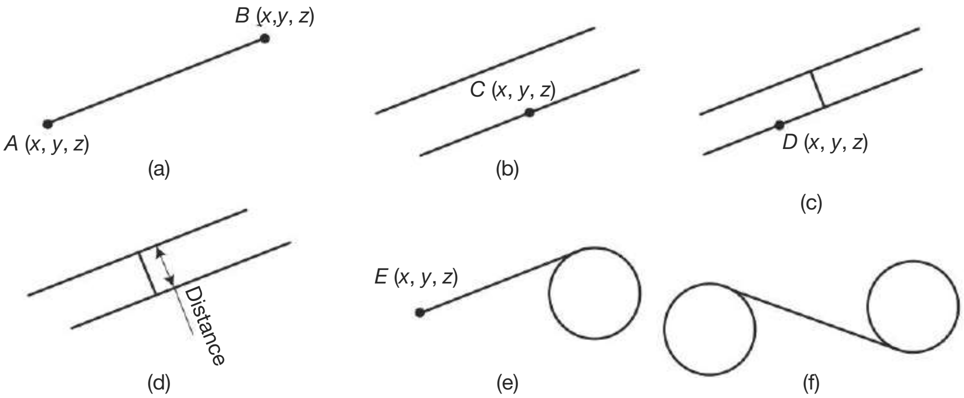


Fig. 4.49 *Line definitions*

4.6.3 Circle Definitions

A few methods of defining the circle objects are shown in Fig. 4.50. As shown in Fig. 4.50a, the simplest method of defining a circle is by means of its centre *A* and radius *R*. Figure 4.50b shows a circle with a centre point *C* and a point *B* on the circumference of the circle. Figure 4.50c shows a circle passing through three collinear points *D*, *E* and *F*. Figure 4.50d shows a circle with a centre *G* and tangential to an existing line. Figure 4.50e shows a circle tangential to two existing lines and a given radius *R*. Figure 4.50f shows a circle tangential to two existing circles and a given radius *R*. It is possible to have similar definitions with variations depending upon the type of existing objects.

4.6.4 Arc Definitions

An arc is a part of a circle. Hence, the method of defining it is very close to that of the circle. The following are a few of the methods that are used for defining an arc:

- three points on the arc (Fig. 4.51a)
- start point, centre, endpoint (Fig. 4.51b)

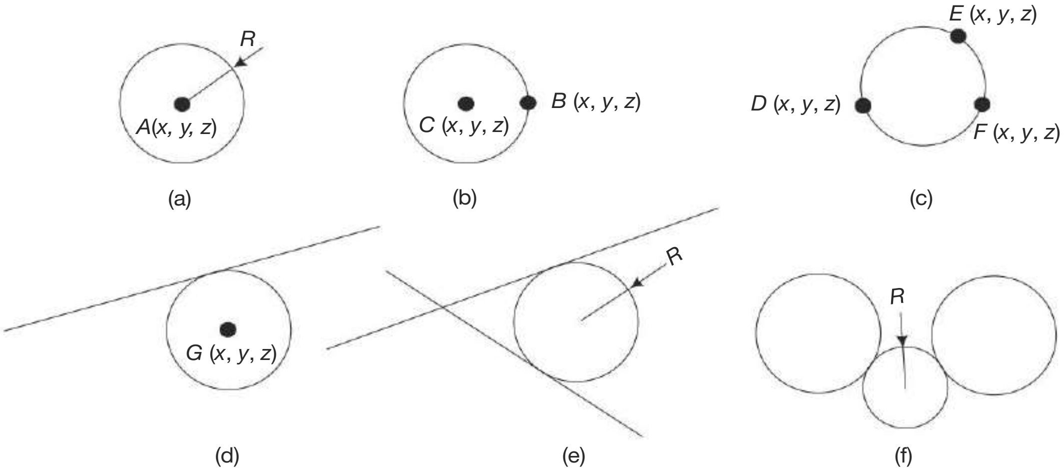


Fig. 4.50 *Circle definitions*

- start point, endpoint, included angle (Fig. 4.51c)
- start point, centre, included angle (Fig. 4.51d)
- start point, endpoint, radius (Fig. 4.51e)

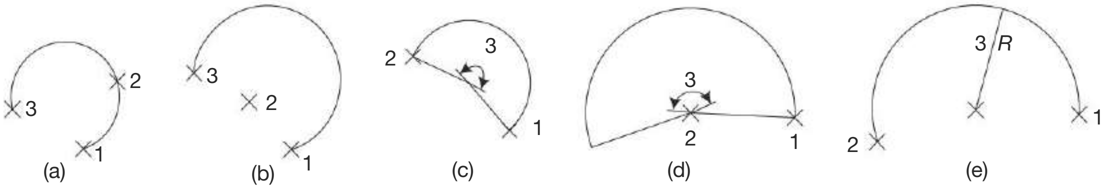


Fig. 4.51 *Arc definitions*

4.6.5 Other Objects

In addition to points, lines, circles and arcs as detailed above, it is possible to define any type of geometric curve like conic sections such as ellipse, parabola and hyperbola, and others.

4.6.6 Wireframe Data Representation

The database in the case of wireframe is represented as two tables (Table 4.2 and 4.3). One table corresponds to the vertices data. In this table (refer to Fig. 4.52), all the vertices are identified by means of their Cartesian coordinates. The second table represents the descriptions of the various edges present in the cube with endpoints identified by the pointers to the first table. There is no face information present in the wireframe model. This example assumed that the object consisted of all planar faces. If the object has cylindrical surfaces as shown in Fig. 4.53, then the data has to be modified to include the curve element data.

4.7 CURVE REPRESENTATION

Representation of the curve geometry can be carried out in two forms:

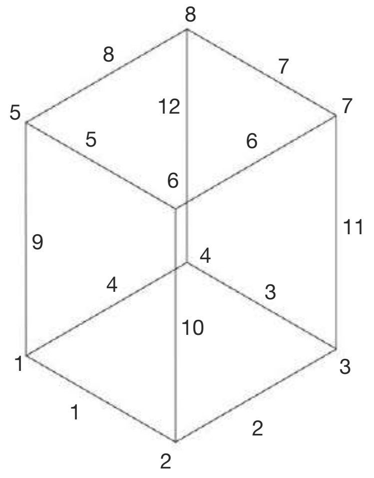


Fig. 4.52 *Wireframe data representation for a cube*

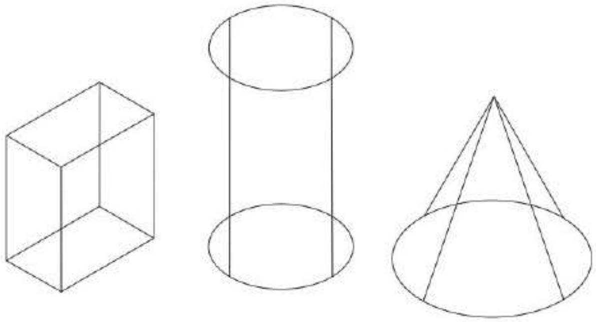


Fig. 4.53 *Wireframe representation of a block, cylinder and cone*

Table 4.2 *Vertex data*

Vertex number	*X coordinate*	*Y coordinate*	*Z coordinate*
1	0	0	0
2	10	0	0
3	10	10	0
4	0	10	0
5	0	0	15
6	10	0	15
7	10	10	15
8	0	10	15

Table 4.3 *Edge details*

Edge number	*Start point*	*Endpoint*
1	1	2
2	2	3
3	3	4
4	4	1
5	5	6
6	6	7
7	7	8
8	8	5
9	1	5
10	1	6
11	1	7
12	1	8

- Implicit form, and
- Parametric form.

In the implicit form, the coordinates (*x, y, z*) are related by two functions such as

$$f(x, y, z) = 0$$
$$g(x, y, z) = 0 \tag{4.1}$$

For a given value of *x*, these equations when solved will give the other coordinates on the curve. The implicit form is convenient for two-dimensional curves of first and second order. For higher orders, the solution is lengthy and inconvenient. It is difficult to handle during computer programming because of its lengthy nature, and requires a lot of computational time for display purpose. Typical curves that can be covered are lines, arcs and circles.

In parametric form, the curve is represented as

$$X = x(u)$$
$$Y = y(u)$$
$$Z = z(u) \tag{4.2}$$

where *X, Y, Z* are the coordinate values on the space curve, and the corresponding functions *x*, *y*, *z* are the polynomials in a parameter, *u*. This representation permits to quickly compute the individual *X, Y, Z* coordinates of all the points on the curve. The parametric form is most convenient for CAD and higher-order curves. The position vector of a point on the curve is fixed by the value of the parameter. The general parametric representation of a curve (Fig. 4.54) is given as

$$p = p(u) \tag{4.3}$$

where *u* is the parameter with which the Cartesian coordinates will be represented as

$$X = x(u); \quad Y = y(u); \quad Z = z(u) \tag{4.4}$$

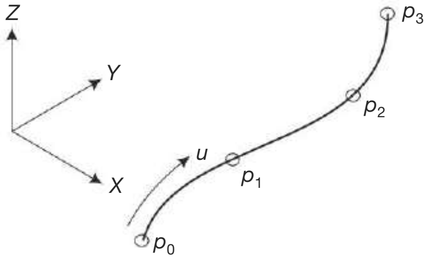


Fig. 4.54 *Parametric curve representation in Cartesian space*

It is also possible to write the above equation in the form of a vector as

$$p = p(u) = [x(u)\ y(u)\ z(u)] \tag{4.5}$$

where *u* is the parameter that can have any value from − ∞ to + ∞. However, generally it may be varied between − 1 to + 1. This form is generally useful for the way the curves can be drawn on the display screen or plotter. Thus, a curve segment can be defined as the locus of all the points whose coordinates are given by continuous-parameter, single-valued polynomials. The parameter variable, *u*, is constrained to some closed interval such as [0, 1].

The tangent vector on a parametric curve is

$$P'(u) = [x'(u)\ y'(u)\ z'(u)] \tag{4.6}$$

where ′ denotes differentiation with respect to the parameter *u* of the position vector components. The slopes of the curve are given by

$$\frac{dy}{dx} = \frac{dy/du}{dx/du} = \frac{y'}{x'}$$
$$\frac{dz}{dy} = \frac{z'}{y'}$$
$$\frac{dx}{dz} = \frac{x'}{z'} \tag{4.7}$$

These form the components of the tangent vector. The magnitude of the tangent vector is given by

$$|P'(u)| = \sqrt{x'^2 + y'^2 + z'^2} \tag{4.8}$$

The directional cosines of the vector are given by

$$\frac{P'(u)}{|P'(u)|} = n_x i + n_y j + n_z k \tag{4.9}$$

where, i, j, k are unit vectors, and n_x, n_y, and n_z are Cartesian space components in the x, y, and z directions, respectively.

The parametric form can be used for analytic curves as well as synthetic curves. Analytical curves are compact, while synthetic curves and provide more flexibility for designers.

4.7.1 Line

Most of the users are familiar with the implicit form of a straight line as

$$Y = mX + c \tag{4.10}$$

where Y and X represent the coordinates of the Cartesian two-dimensional system.

The parametric form for the line can be given as

$$\begin{aligned} X &= a_x u + b_x \quad \text{for } 0 \le u \le 1 \\ Y &= a_y u + b_y \\ Z &= a_z u + b_z \end{aligned} \tag{4.11}$$

Example 4.1 Write down the (a) implicit, and (b) parametric forms of a horizontal line passing through its midpoint (50, 50).

Solution (a) *Implicit form* $Y = 50$

(b) *Parametric form* Defining the interval of the parameter to vary from 0 to 1, and that the line is defined between (0, 50) and (100, 50),

$$X = 100\,u$$
$$Y = 50$$

Example 4.2 Determine the parametric representation of the line segment between the position vectors P_1 [1 1] and P_2 [4 5]. What are the slope and tangent vectors for this line?

Solution A parametric representation is

$$P(u) = P_1 + (P_2 - P_1)u = [1\ 1] + ([4\ 5] - [1\ 1])u \quad 0 \le u \le 1$$
$$P(u) = [1\ 1] + [3\ 4]u \quad 0 \le u \le 1$$

Parametric representations of the x and y components are

$$x(u) = x_1 + (x_2 - x_1)\,u = 1 + (4 - 1)\,u = 1 + 3u$$
$$y(u) = y_1 + (y_2 - y_1)\,u = 1 + (5 - 1)\,u = 1 + 4u$$

The tangent vector is obtained by differentiating $P(u)$.

$$P'(u) = [x'(u), y'(u)] = [3\ 4]$$

or the tangent vector is $\overline{V}_t = 3i + 4j$

where, i, j are unit vectors in the x, y directions, respectively.

The slope of the line segment is, $\dfrac{dy}{dx} = \dfrac{y'}{x'} = \dfrac{4}{3}$

4.7.2 Circle

The implicit form of the circle whose centre lies at the origin of the coordinate system is given by

$$x^2 + y^2 = r^2 \tag{4.12}$$

The parametric form of a circle (Fig. 4.55) is given by

$$X = r \cos \theta$$
$$Y = r \sin \theta \tag{4.13}$$

where $0 \le \theta \le 2\pi$

r is the radius of the circle.

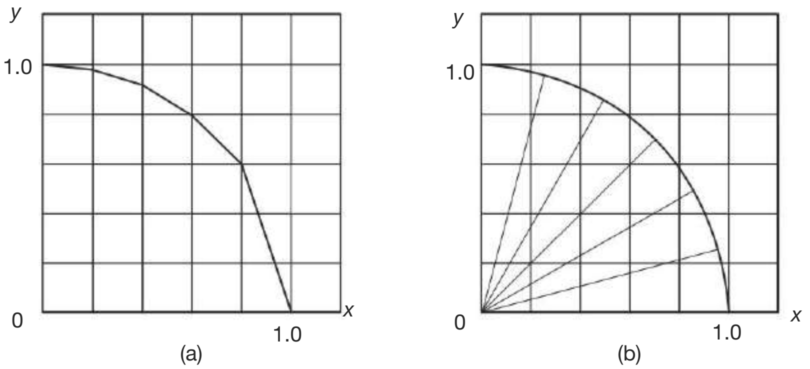


Fig. 4.55 *Circle*

Notice that in Fig. 4.55, the circle is drawn for one quarter in both implicit form and parametric form. In parametric form (b), the circle is more uniform with equal increments of the angle, while in the implicit form in (a), the circle has unequal line segments along it for equal increments of the Y-coordinate.

The alternative form of parametric form for the circle is given by

$$X = x(u) = r\frac{1 - u^2}{1 + u^2}$$
$$Y = y(u) = r\frac{2u}{1 + u^2} \tag{4.14}$$

For $-1 \le u \le 1$

The equation of a circle whose centre is located at (a, b) is given by

$$(x - a)^2 + (y - b)^2 = r^2 \tag{4.15}$$

The parametric form of the circle whose centre is located at (a, b) is given by

$$X = a + r \cos \theta$$
$$Y = b + r \sin \theta \tag{4.16}$$

Example 4.3 Represent a circle with centre (0, 0) and a radius of 50 mm through the (a) implicit form, as well as (b) parametric form.

Solution (a) *Implicit form* $x^2 + y^2 = 2500$

(b) *Parametric form* Defining the interval of the parameter u to vary from –1 to 1 for a quarter of the circle,

$$X = 50\,\frac{1 - u^2}{1 + u^2}$$

$$Y = 50\,\frac{2u}{1 + u^2}$$

Example 4.4 Find the equation for a line passing through (75, 60) and (25, 25). Find the equation of a line that is perpendicular to the above line and passing through a point (45, 25).

Solution To find the equation of the line

The slope, $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{60 - 25}{75 - 25} = 0.7$

The intercept, $C = 60 - 0.7 \times 75 = 7.5$

The equation of the line is $Y = 0.7\,X + 7.5$

The slope of the perpendicular line $= -\frac{1}{0.7} = -\,0.143$

The intercept of the required line is, $C = 25 + 0.143 \times 45 = 6.69$

The required equation is $Y = -\,0.143\,X + 6.69$

Example 4.5 Find the equation of a line that is tangent to a circle whose equation is $x^2 + y^2 = 36$ and passing through the point (12, 4).

Solution The intercept of the line is given by [Chasen 1978]

$$(X^2 - r^2)\,C^2 + 2\,Y\,r^2\,C - (X^2 + Y^2)\,r^2 = 0$$

and the slope is given by $m = \frac{Y - C}{X}$

The first equation gives two solutions for the slope corresponding to the two possible lines.

$$(12^2 - 6^2)\,C^2 + 2 \times 4 \times 36\,C - (12^2 + 4^2)\,36 = 0$$

$$108\,C^2 + 288\,C - 5760 = 0$$

The solution for $C = 6.09,\ -\,8.7569$

The slope of the equation 1, $m_1 = \frac{4 - 6.09}{12} = -\,0.1742$

The slope of the equation 2, $m_2 = \frac{4 + 8.7569}{12} = 1.063$

The two equations of the lines are

$$Y = -\,0.1742\,X + 6.09$$

$$Y = 1.063\,X - 8.7569$$

A number of such cases for calculating the lines and circles are given in Chasen, 1978.

4.7.3 Ellipse

The parametric form of an ellipse whose centre lies at the origin of the coordinate system (Fig. 4.56) is given by

$$x = a\cos\theta$$
$$y = b\sin\theta \qquad (4.17)$$

where a and b are the semi-major and minor axes of the ellipse. The implicit form of the ellipse is given by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad (4.18)$$

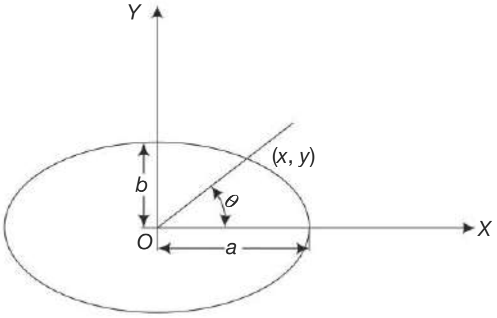


Fig. 4.56 *Ellipse*

4.7.4 Parabola

A parabola is the intersection curve between a right circular cone and a plane such that the plane is parallel to the side of the cone. The cutting plane will actually be intersecting the base, and thus a parabola is an open curve. A parabola has a *focus* and a fixed line, called *directrix*. Any point on the parabola is equidistant from the focus and the directrix. Parabolas have a unique reflective property in that rays that are parallel to the axis of the parabola and fall on a parabolic surface will all converge at the focus of the parabola. This property is utilised in designing solar collectors which concentrate the sun's energy to a fixed location to get very high temperatures. Reflective mirrors used for automobile lights and for lighthouses are all parabolic. The implicit form of a parabola is given by

$$y^2 = 4ax \qquad (4.19)$$

One of the parametric forms of a parabola (Fig. 4.57) is given by

$$x = au^2$$
$$y = 2au \qquad (4.20)$$

where $0 \le u \le \infty$

Since a parabola is not a closed curve like an ellipse, the value of u needs to be limited for display purposes.

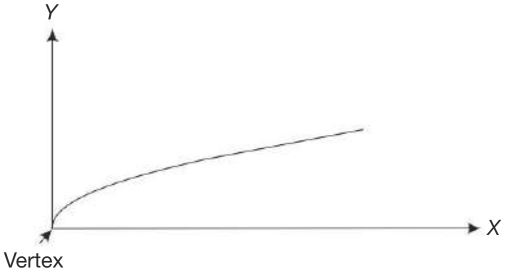


Fig. 4.57 *Parabola*

4.7.5 Hyperbola

A hyperbola is the intersection curve between a right circular cone and a plane such that the plane is parallel to the axis of the cone as shown in Fig. 4.58. The cutting plane will actually be intersecting the base, and thus a hyperbola is an open curve. Mathematically, a hyperbola has two foci. The distance of any point on the hyperbola from its foci has a constant difference. A hyperbola is used in the construction of reflective telescopes used in astronomy. The orbits of comets follow hyperbolas.

The implicit form of a hyperbola is given by

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \qquad (4.21)$$

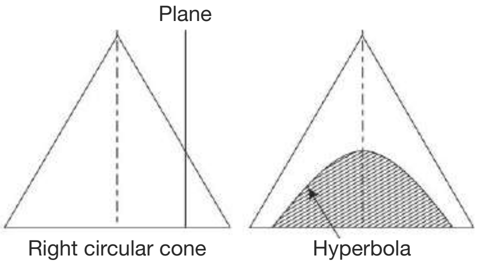


Fig. 4.58 *Hyperbola and its relation to the right circular cone*

One of the parametric forms of a hyperbola (Fig. 4.50) is given by

$$x = a \cosh \theta$$
$$y = b \sinh \theta \tag{4.22}$$

4.7.6 Curve Fitting

Often designers have to deal with information for a given object in the form of coordinate data rather than any geometric equation. In such cases, it becomes necessary for designers to use mathematical techniques of curve fitting to generate the necessary smooth curve that satisfies the requirements. An example of an airfoil section is shown in Fig. 4.59.

Fig. 4.59 *Airfoil section curve fitted with data points*

Since many data points are to be used for representing the curve, often curve fitting methods are utilised, whereby the point set is replaced by an equation. A brief review is presented here of the curve-fitting methods that are commonly used in modelling applications. Curve-fitting methods can be broadly classified into *interpolation techniques*, where the curve will pass through all the points, and *approximation methods*, where the best fit will be obtained instead of the curve passing through all the points. These methods are utilised in differing situations depending upon the requirement. A comparison of these two methods is given later.

Lagrange Polynomial In this method, a Lagrange polynomial is used for fitting the data points. This technique is quite useful for programming in a digital computer. A second-order Lagrange polynomial is given below for fitting the three data points, (x_0, y_0), (x_1, y_1), and (x_2, y_2).

$$L_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} y_2 \tag{4.23}$$

The generalised Lagrange interpolation of the nth order is given as

$$L_n(x) = \sum_{i=0}^{n} \frac{\prod_{n+1}(x)}{(x - x_i)\prod_{n+1}(x_i)} y_i \tag{4.24}$$

B-Splines When large amounts of data are available, it is possible to use B-splines for fitting all the data available in a piecewise manner. B-splines can be of any degree but in computer graphics, the degrees 2 or 3 are generally found to be sufficient. The 3rd order polynomial equation requires 4 sets of data. The polynomial between x_i and x_{i+1} can then be defined as

$$f_i(x) = a_i + a_{i-1}x + a_{i-2}x^2 + a_{i-3}x^3 \tag{4.25}$$

Substituting the values of x_i and x_{i+1} in the above equation, we get

$$f_i(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 \tag{4.26}$$

$$f_i(x_{i+1}) = a_0 + a_1 x_{i+1} + a_2 x_{i+1}^2 + a_3 x_{i+1}^3 \tag{4.27}$$

In order to obtain the four constants a_0 to a_3, two more equations are required. This is obtained as the condition for smooth transition between the curve segments as the first- and second-order derivative of the above equation.

$$f_i(x) = \frac{1}{h_i}\left[\frac{(x_i - x)^3}{6} M_{i-1} + \frac{(x - x_{i-1})^3}{6} M_i\right] + c_i\,(x_i - x) + d_i\,(x - x_{i-1}) \tag{4.28}$$

where

$$h_i = x_i - x_{i-1}$$

$$M_i = f''(x_i), \quad \text{and} \quad M_{i-1} = f''(x_{i-1})$$

$$c_i = \frac{1}{h_i}\left[y_{i-1} - \frac{h_i^2}{6} M_{i-1}\right], \quad \text{and} \quad d_i = \frac{1}{h_i}\left[y_i - \frac{h_i^2}{6} M_i\right] \tag{4.29}$$

Approximate Methods Approximate methods allow for fitting a curve that provides the best fit to the given data. The best-fit curve represents the general trend of the data (without passing through every point). Several criteria can be used to derive the best-fit curve.

(a) Method of Least Squares Since $f(x)$ is an approximate curve and not passing through all the points of data, the error at any given point (x_i, y_i) is given by

$$e_i = y_i - f(x_i) \tag{4.30}$$

The summation of all errors gives an estimate of the total deviation of the curve from the points. However, since the errors tend to cancel out each other, the sum of squares of the error, S is minimised. Hence, the condition for n data points is

$$S = \sum_{i=1}^{n} (e_i)^2 = \sum_{i=1}^{n} [y_i - f(x_i)]^2 \tag{4.31}$$

(b) Polynomial Curve Fitting In this the coordinate relationship is given in the form of a polynomial of any degree depending upon the requirements of the modelling. The polynomial can be of any order and to solve it requires at least $(n + 1)$ data points for an n degree polynomial. Depending upon the degree of the polynomial, the method of solving will be different.

A first-degree polynomial can be specified as

$$y = a_0 + a_1 x \tag{4.32}$$

This is similar to the line equation that was discussed earlier. The constants a_0 and a_1 need to be calculated substituting the two data points (x_1 , y_1) and (x_2 , y_2) in the above equation. Then the constants are given by

$$a_0 = \frac{x_1 y_2 + y_1 x_2}{x_2 - x_1}; \quad \text{and} \quad a_1 = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.33}$$

The general form of a higher degree (n^{th}) polynomial can be specified as

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \tag{4.34}$$

Similar to the above, $n + 1$ or more data points are substituted, and the resultant equations are to be solved to get the constants a_0 to a_n. Using the least-square approach, the best-fit polynomial can be evaluated.

Substitute the data points $(x_{\text{i}}, y_{\text{i}})$ $i = 1, 2, \ldots, m$. Taking the least-square approach, we have

$$S = [y_1 - (a_0 + a_1 x_1 + a_2 x_1^{\,2} + \ldots + a_n x_1^{\,n})]^2 + [y_2 - (a_0 + a_1 x_2 + a_2 x_2^{\,2} + \ldots + a_n x_2^{\,n})]^2 + \ldots$$
$$+ [y_m - (a_0 + a_1 x_m + a_2 x_m^{\,2} + \ldots + a_n x_m^{\,n})]^2 \tag{4.35}$$

Taking the first partial derivative and equating it to zero and simplifying gives the following normal equations shown in matrix form as

$$[X] = [A]^{-1}\,[B] \tag{4.36}$$

$$\text{where } [X] = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, [A] = \begin{bmatrix} m & \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 & \cdots & \sum_{i=1}^{m} x_i^n \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i^3 & \cdots & \sum_{i=1}^{m} x_i^{n+1} \\ \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i^3 & \sum_{i=1}^{m} x_i^4 & \cdots & \sum_{i=1}^{m} x_i^{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{m} x_i^n & \sum_{i=1}^{m} x_i^{n+1} & \sum_{i=1}^{m} x_i^{n+2} & \cdots & \sum_{i=1}^{m} x_i^{2n} \end{bmatrix}, \text{ and } [B] = \begin{bmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} (x_i y_i) \\ \sum_{i=1}^{m} (x_i^2 y_i) \\ \vdots \\ \sum_{i=1}^{m} (x_i^n y_i) \end{bmatrix} \tag{4.37}$$

These are $(n + 1)$ equations in $(n + 1)$ unknowns and hence can be solved by using the Gaussian elimination technique.

Having seen the different methods of curve fitting, a comparison can be provided between them with respect to computer graphics application.

Interpolation methods	*Best-fit methods*
1. It is necessary that the curve produced will have to go through all the data points.	The curve will not pass through all the points, but will result in a curve that will be closest to as many points as possible.
2. Cubic splines and Lagrange interpolation methods are used.	Regression and least-square methods are used for the purpose. Bêzier curves also fall in this category.
3. The shape of the curve is affected to a great extent by manipulating a single data point. The nature of tweaking is unpredictable.	It is possible to have a local modification easily by tweaking a single point where the behaviour is more predictable.

4.7.7 Synthetic Curves

The analytical curves described earlier are not sufficient to fully describe some of the geometries that are encountered in real life. For example, modelling of aircraft bodies, automobile body shapes, moulds and die profiles, horse saddles, ship hulls, etc., are rather difficult do represent by analytical curves. For this purpose, free-form or synthetic curves are developed. These synthetic curves have many curve segments. Essentially, these are interpolation curves similar to those that were explained earlier for the curve-fitting application. They need to have a smooth curve to pass through all or as many of the data points. The properties expected of these curves are

- Easy to enter the data and easy to control the continuity of the curves to be designed
- Require much less computer storage for the data representing the curve
- Have no computational problems and faster in computing time

Since synthetic curves have different types of curve segments, they need to maintain certain continuity requirements. This means how smoothly the curve transition takes place between the connection points of the curves. Normally, three types of continuity are possible: the first one is C^0 continuity, which is simply connecting two curves; the second one is C^1 continuity, when the gradients at the point of joining must be same; while the third one C^2 is curvature continuity, which means in addition to the gradient, it also has the same centre of curvature. Some examples of synthetic curves that are commonly used in CAD/CAM are

- Hermite cubic spline
- Bézier curves
- B-spline curves
- Rational B-splines (including Non-uniform rational B-splines – NURBS)

Hermite Cubic Spline Hermite cubic splines are the more general form of curves that can be defined through a set of vertices (points). A spline is a piecewise parametric representation of the geometry of a curve with a specified level of parametric continuity. Each segment of a Hermite cubic spline is approximated by a parametric cubic polynomial to maintain the C^2 continuity (See Fig. 4.60). The functions served by a cubic function are the following:

- A cubic polynomial generates C^0, C^1 and C^2 continuity in the associated curves.
- A cubic polynomial permits inflection within a curve segment and allows representation of non-planar space curves.
- Higher-order polynomials oscillate about the control points, and are uneconomical for storing information and computation time.

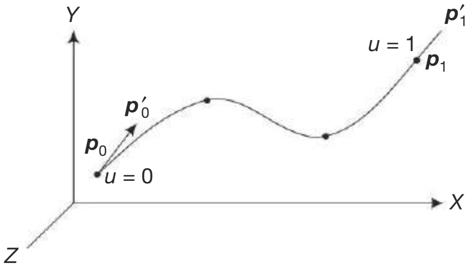


Fig. 4.60 *Hermite cubic spline curve*

The parametric equation of a Hermite cubic spline is given by

$$p(u) = \sum_{i=0}^{3} C_i u^i \quad u \in [0, 1] \tag{4.38}$$

In an expanded form, it can be written as

$$p(u) = \boldsymbol{C_0} + \boldsymbol{C_1}\, u^1 + \boldsymbol{C_2}\, u^2 + \boldsymbol{C_3}\, u^3 \tag{4.39}$$

where u is a parameter, and $\boldsymbol{C_i}$ are the polynomial coefficients.

In the matrix form, it can be written as

$$p(u) = [1 \quad u \quad u^2 \quad u^3] \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} \tag{4.40}$$

The tangent vector is obtained by differentiating the curve at any point with respect to u.

$$p'(u) = \sum_{i=0}^{3} C_i\, i\, u^{i-1} \quad u \in [0, 1] \tag{4.41}$$

Applying the boundary conditions at the two end points ($\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_0', \boldsymbol{p}_1'$) of the spline, it is possible to show (see Ibrahim Zeid)

$$\boldsymbol{p}(u) = (1 - 3u^2 + 2u^3)\,\boldsymbol{p_0} + (3u^2 - 2u^3)\,\boldsymbol{p_1} + (u - 2u^2 + u^3)\,\boldsymbol{p_0'} + (-u^2 + u^3)\,\boldsymbol{p_1'} \tag{4.42}$$

This can be rewritten in the matrix form as

$$p(u) = [1 \quad u \quad u^2 \quad u^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_0' \\ p_1' \end{bmatrix} \tag{4.43}$$

The same equation can be written in the compact form as follows:

$$p(u) = \{\mathbf{U}\}\ [\mathbf{M}]\ [\mathbf{P}] \tag{4.44}$$

where the basis function is

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$

$$\mathbf{U} = [1 \quad u \quad u^2 \quad u^3] \tag{4.45}$$

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \boldsymbol{p}'_0 \\ \boldsymbol{p}'_1 \end{bmatrix} \tag{4.46}$$

This equation helps in calculating the Hermite cubic spline in terms of the endpoints and their tangent vectors. It is possible to utilise the same for fitting a number of data points by having a number of spline segments and their blending function. However, this procedure is not readily used in the design process, because of the difficulty in visualising the nature of the curve when a single point is moved. Tweaking of any single point changes the entire Hermite spline curve. Also, the need for tangent vectors at the endpoints makes it more difficult to be used.

Example 4.6 Find the midpoint of a Hermite cubic spline with the two points as [1, 1], and [6, 5] and the tangent vectors as [0, 4] and [4, 0].

Solution From the given end points and tangent vectors,

$$x_0 = 1, \quad y_0 = 1, \quad \text{and} \quad x'_0 = 0, \quad y'_0 = 4$$

$$x_1 = 6, \quad y_1 = 5, \quad \text{and} \quad x'_1 = 4, \quad y'_1 = 0$$

From the spline equation (4.43), we can write

$$p_x(u) = [1 \quad u \quad u^2 \quad u^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x'_0 \\ x'_1 \end{bmatrix}$$

$$p_y(u) = [1 \quad u \quad u^2 \quad u^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y'_0 \\ y'_1 \end{bmatrix}$$

Midpoint means, $u = 0.5$. Substituting this value in the above two equations, and solving, we get

$$\mathbf{p}_x(0.5) = [1 \quad 0.5 \quad 0.5^2 \quad 0.5^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{6} \\ \mathbf{0} \\ \mathbf{4} \end{bmatrix} = 3$$

Similarly,

$$\mathbf{p}_y(0.5) = [1 \quad 0.5 \quad 0.5^2 \quad 0.5^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{5} \\ \mathbf{4} \\ \mathbf{0} \end{bmatrix} = 3.5$$

The required midpoint is [3, 3.5].

Bernestein Polynomials Polynomials are simply defined, can be calculated quickly on computer systems and represent a tremendous variety of functions. They can be differentiated and integrated easily, and can be pieced together to form spline curves that can approximate any function to any accuracy desired as shown above.

$$p(u) = a_0 + a_1 u + a_2 u^2 + \ldots + a_{n-1} u^{n-1} + a_n u^n \tag{4.47}$$

This polynomial, whose basis is called the *power basis*, is only one of an infinite number of bases for the space of polynomials. The *Bernstein basis*, called Bernestein polynomials, has many useful properties. They are written as

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i} \tag{4.48}$$

where $\binom{n}{i} = \dfrac{n!}{i!(n-i)!}$

First-order Bernestein polynomials can be written as

$$\begin{aligned} B_{0,1}(u) &= 1 - u \\ B_{1,1}(u) &= u \end{aligned} \tag{4.49}$$

Second-order Bernestein polynomials can be written as

$$\begin{aligned} B_{0,2}(u) &= (1-u)^2 \\ B_{1,2}(u) &= 2u(1-u) \\ B_{2,2}(u) &= u^2 \end{aligned} \tag{4.50}$$

Third-order Bernestein polynomials can be written as

$$\begin{aligned} B_{0,3}(u) &= (1-u)^3 \\ B_{1,3}(u) &= 3\,u(1-u)^2 \\ B_{2,3}(u) &= 3\,u^2(1-u) \\ B_{3,3}(u) &= u^3 \end{aligned} \tag{4.51}$$

Bézier Curves Bézier chose Bernestein polynomials (Eq. 4.48) as the basis functions for the curves. Based on these basis functions, the equation for the Bézier curve is given by

$$p(u) = \sum_{i=0}^{n} p_i B_{i,n}(u) \quad u \in [0, 1] \tag{4.52}$$

For $n = 3$, substituting Eq. 4.51 in 4.52, we get

$$\boldsymbol{p}(u) = (1-u)^3 \boldsymbol{p_0} + 3u\,(1-u)^2 \boldsymbol{p_1} + 3u^2\,(1-u)\,\boldsymbol{p_2} + u^3 \boldsymbol{p_3} \tag{4.53}$$

This can be written in the matrix form as

$$p(u) = [u^3 \quad u^2 \quad u \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{4.54}$$

The same equation can be written in the compact form as follows:

$$\boldsymbol{p(u)} = \mathbf{U\,M_B\,P} \tag{4.55}$$

where

$$\mathbf{M_B} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.56}$$

$$\mathbf{U} = [u^3 \quad u^2 \quad u \quad 1] \tag{4.57}$$

$$\mathbf{P} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{4.58}$$

As explained above, when the number of control points in a Bézier curve increases, the degree of the polynomial increases, and this provides for a global modification effect rather than local. What this means is that whenever a single vertex is modified, the entire curve gets modified. However, it is also possible to break the Bézier curve into smaller segments and then all of them are linked. An example is shown in Fig. 4.61 where two Bézier curves, one defined by [p_0 p_1 p_2 p_3] and the other by [p_3 p_4 p_5 p_6] are joined together maintaining continuity.

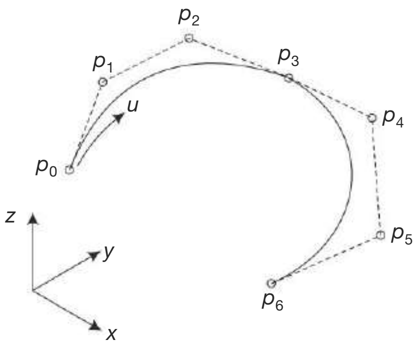


Fig. 4.61 *Two cubic Bézier curves joined at p_3*

Some important properties of Bézier curves that are relevant for CAD applications are as follows:

- The Bézier curve passes through the first and last control points while it maintains proximity to the intermediate control points. As such, the entire Bézier curve lies in the interior of the convex hull of the control points.
- If a control point is moved, the entire curve moves.
- Being polynomial functions, Bézier curves are easily computed, and infinitely differentiable.
- If the control points of the Bézier curve are transformed, the curve moves to the corresponding new coordinate frame without changing its shape.

Example 4.7 A cubic Bézier curve is defined by the control points as (20, 20), (60, 80), (120, 100), and (150, 30). Find the equation of the curve and its midpoint.

Solution The equation of the curve is given by

$$p(u) = [u^3 \quad u^2 \quad u \quad 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 20 & 20 \\ 60 & 80 \\ 120 & 100 \\ 150 & 30 \end{bmatrix}$$

$$= [u^3 \quad u^2 \quad u \quad 1]\begin{bmatrix} -50 & -50 \\ 60 & -120 \\ 120 & 180 \\ 20 & 20 \end{bmatrix}$$

$$= [(-50\,u^3 + 60\,u^2 + 120\,u + 20), (-50\,u^3 - 120\,u^2 + 180\,u + 20)]$$

It is possible to calculate the x and y coordinates of the Bézier curve by varying the parameter u from 0 to 1 in the above equations on the right-hand side.

Midpoint is when $u = 0.5$

$$\mathbf{p}(0.5) = [-50\,(0.5)^3 + 60\,(0.5)^2 + 120\,(0.5) + 20, -50\,(0.5)^3 - 120\,(0.5)^2 + 180\,(0.5) + 20] = [88.75, 73.75]$$

Example 4.8 Fit a cubic Bézier curve for the following control points: (1, 3), (4, 5), (5, 7) and (8, 4). Calculate the points at $u = 0.4$ and 0.6.

Solution The equation of the curve is given by

$$p(u) = [u^3 \quad u^2 \quad u \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 4 & 5 \\ 5 & 7 \\ 8 & 4 \end{bmatrix}$$

$$= [u^3 \quad u^2 \quad u \quad 1] \begin{bmatrix} 4 & -5 \\ -6 & 0 \\ 9 & 6 \\ 1 & 3 \end{bmatrix}$$

$$= [(4u^3 - 6u^2 + 9u + 1), (-5u^3 + 6u + 3)]$$

For $u = 0.4$

$$\mathbf{p}(0.4) = [4(0.4)^3 - 6\,(0.4)^2 + 9\,(0.4) + 1, -5\,(0.4)^3 + 6\,(0.4) + 3] = [3.896, 5.08]$$

For $u = 0.6$

$$\mathbf{p}(0.6) = [4(0.6)^3 - 6(0.6)^2 + 9(0.6) + 1, -5(0.6)^3 + 6(0.6) + 3] = [5.104, 4.08]$$

B-splines One of the problems associated with the Bézier curves is with an increase in the number of control points, the order of the polynomial representing the curve increases. To reduce this complexity, the curve is broken down into more segments with better control exercised with individual segments, while maintaining a simple continuity between the segments. An alternative is to use a B-spline to generate a single piecewise parametric polynomial curve through any number of control points with the degree of the polynomial selected by the designer. B-splines exhibit a local control in that whenever a single vertex is moved, only those vertices around that will be affected while the rest remain the same. In fact, B-splines are generalisations of Bézier curves. Thus, they have many similarities but at the same time have many advantages compared to Bézier curves.

B-spline curves have the flexibility of choosing the degree of the curve irrespective of the number of control points. With four control points, it is possible to get a cubic Bézier curve, while with a B-spline curve one can get a linear, quadratic or cubic curve. Similar to a Bézier curve, a B-spline also uses the basis (blending) functions and the equation is of the form

$$p(u) = \sum_{i=0}^{n} p_i \, N_{i,k}(u) \quad 0 \le u \le u_{\max} \tag{4.59}$$

where $N_{i,k}(u)$ are the basis functions for B-splines.

$$N_{i,1}(u) = 1 \quad \text{if } u_i \le u \le u_{i+1} \tag{4.60}$$
$$= 0 \quad \text{otherwise}$$

and

$$N_{i,k} = \frac{(u - u_i)\, N_{i,k-1}(u)}{u_{i+k-1} - u_i} + \frac{(u_{i+k} - u)\, N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}} \tag{4.61}$$

where k controls the degree $(k - 1)$ of the resulting polynomial in u and also the continuity of the curve. The u_i are the knot values, which relate the parametric variable u to the $\mathbf{p}_i$ control points. B-spline functions have the following properties:

- The plotting of a B-spline curve is done by varying the parameter u over the range of knot values (u_{k-1}, u_{n+1}).
- The knot vector adds flexibility to the curve and provides better control of its shape.
- *Partition of Unity* For any knot span, $[u_i, u_{i+1}]$, $\sum_{i=0}^{n} N_{i,k}(u) = 1$
- *Positivity* $N_{i,k}(u) \geq 0$ for all i, k and u.
- *Local Support Property* $N_{i,k}(u) = 0$ if $u \notin [u_i, u_{i+k+1}]$

  This property can be deduced from the observation that $N_{i,k}(u)$ is a linear combination of $N_{i,k-1}(u)$ and $N_{i+1,k-1}(u)$.
- *Continuity* $N_{i,k}(u)$ is $(k-2)$ times continuously differentiable, being a polynomial.
- The curve follows the shape of the control points and lies in the convex hull of the control points.
- The entire B-spline curve can be affinely transformed by transforming the control points and redrawing the curve from the transformed points.
- B-splines exhibit local control, i.e., when a control point is moved only that segment is influenced.

  For $k = 4$, the equation can be written in the compact form as follows:

$$\mathbf{p}_i(\boldsymbol{u}) = \mathbf{U}\,\mathbf{M}_S\,\mathbf{P}_i \tag{4.62}$$

where

$$\mathbf{M}_S = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & 6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{4.63}$$

$$\mathbf{U} = [u^3 \quad u^2 \quad u \quad 1] \tag{4.64}$$

$$\mathbf{P}_i = \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix} \quad i \in [1 : n-2] \text{ for open curves} \tag{4.65}$$

Some of the properties of B-splines that are useful for CAD applications are given here:

- Local control can be achieved by changing the position of the control points.
- A B-spline curve tightens by increasing its degree. As the degree of the B-spline is lowered, it comes closer to the control polygon.
- If $k = n + 1$, then the resulting B-spline curve is a Bézier curve.
- The B-spline curve is contained in the convex hull of its control points.
- Affine transformations of the coordinate system do not change the shape of the B-spline curve.
- Increasing the degree of the curve makes it more difficult to control, and hence a cubic B-spline is sufficient for a majority of applications.

Rational Curves The curves discussed so far are called non-rational curves. A rational curve utilises the algebraic ratio of two polynomials. They are important in CAD because of their invariance when geometric transformations are applied. It is possible to have rational Bézier curves, rational conic sections, rational B-splines, etc.

The homogenous coordinates as discussed in Chapter 3 are required to formulate rational curves. As discussed earlier, in homogeneous representation, an n-dimensional space is mapped into $(n + 1)$-dimensional space by adding an additional dimension. For example, a point in E^3 space with coordinates (x, y, z) is represented in homogeneous space as (x^*, y^*, z^*, h) where h is simply a scalar factor, and 1 is often used for convenience. A rational curve defined by (n + 1) points is given by

$$\mathbf{p}(u) = \sum_{i=0}^{n} \mathbf{p}_i \, R_{i,\,k}(u) \quad 0 \le u \le u_{\max} \tag{4.66}$$

where $R_{i,\,k}(u)$ is the rational B-spline basis function and is given by

$$R_{i,\,k}(u) = \frac{h_i \, N_{i,\,k}(u)}{\sum_{i=0}^{n} h_i \, N_{i,\,k}(u)} \tag{4.67}$$

Equation 4.67 is the generalisation of the basis function used in B-splines (Eq. 4.61) $N_{i,\,k}(u)$. As a result, all the advantages claimed earlier are available for the rational basis function. A rational B-spline is considered to be a single representation that can define a variety of curves used in all wireframe, surface and solid entities.

NURBS Uniform cubic B-splines are the curves with the parametric intervals defined at equal lengths. The most common scheme used in all the CAD systems is the non-uniform rational B-spline (commonly known as NURB), allowing a non-uniform knot vector. It includes both the Bézier and B-spline curves. As a result, IGES used NURB as the standard curve definition for data exchange. NURBS uses an additional set of (n + 1) parameters, w_i, called weights to add greater flexibility to the curve.

For example, a rational form of the Bézier curve can be written as

$$\mathbf{p}(u) = \frac{\sum_{i=0}^{n} p_i \, w_i \, B_{i,\,n}(u)}{\sum_{i=0}^{n} w_i \, B_{i,\,n}(u)} \qquad u \in [0,1] \tag{4.68}$$

where w_i is the weighing factor for each of the vertices. When all $w = 1$, this expression becomes that of the conventional Bézier form, since

$$\sum_{i=0}^{n} B_{i,\,n}(u) = 1 \tag{4.69}$$

Similarly, the rational form of the B-splines can be written as

$$\mathbf{p}(u) = \frac{\sum_{i=0}^{n} w_i \, p_i \, N_{i,\,k}(u)}{\sum_{i=0}^{n} w_i \, N_{i,\,k}(u)} \tag{4.70}$$

Some of the reasons why NURBS have found such a widespread acceptance in the CAD/CAM and graphics community are as follows:

- They have all of B-spline surface abilities. In addition, they overcome the limitation of B-spline surfaces by associating each control point with a weight.
- Uniform representation for a large variety of curves and surfaces. This helps with the storage of geometric data.
- NURBS are invariant during geometric transformations as well as projections.
- NURBS is flexible for designing a large variety of shapes by manipulating the control points and weights. Weights in the NURBS data structure determine the amount of surface deflection toward or away from its control point.

- It makes it possible to create curves that are true conic sections. Surfaces based on conics, arcs or spheres can be precisely represented by a NURBS surface.
- Evaluation of NURBS is reasonably fast and numerically stable.
- Number of facilities available in NURBS such as knot insertion/refinement/removal, degree elevation, splitting, etc., makes them ideal to be used throughout the design process.
- NURBS surfaces can be incorporated into an existing solid model by 'stitching' the NURBS surface to the solid model.
- Reverse engineering is heavily dependent on NURBS surfaces to capture digitised points into surfaces.

Some of the problems associated with the use of NURBS are the following:

- Analytical curves and surfaces require additional storage.
- NURBS parameterisation can often be affected by improper application of the weights, which may lead to subsequent problems in surface constructions.
- Not all geometric interrogation techniques work well with NURBS.

4.8 SURFACE-REPRESENTATION METHODS

As explained earlier, there are a number of surfaces that are used in CAD applications whose representations will be discussed now. The surface representation will be an extension of the curve representation that was studied previously. Similar to curves, surfaces can also be represented by both non-parametric (implicit) as well as parametric methods. Curve-fitting methods described earlier can be used to fit surface data points. Alternatively, there are many methods available to fit either a single equation for a surface or a series of surface patches that are connected together having some kind of continuity between the patches.

Non-parametric (implicit) representation of surfaces, is defined by a polynomial of three variables as

$$Z = f(x, y) = \sum_{m=0}^{p} \sum_{n=0}^{q} a_{mn} x^m y^n \tag{4.71}$$

The surface can be described by xy grid of size $(p + 1) \times (q + 1)$ points. Many implicit surfaces do not have any parametric form. Therefore, in terms of expressive power, implicit surfaces are more powerful than parametric surfaces. These are also called *algebraic surfaces*. For example, spheres and all quadric surfaces are algebraic surfaces of degree two, while a torus is a degree-four algebraic surface.

It is also possible that some algebraic surfaces have rational parametric forms, which are called *rational algebraic surfaces*. It is possible for an algebraic surface in rational parametric form to be converted into implicit form by eliminating the parameters u and v. This process is called *implicitisation*.

The parametric equation of a surface (Fig. 4.62) is defined by a set of three functions, one for each coordinate, as follows:

$$\mathbf{p}(u, v) = [x(u, v), y(u, v), z(u, v)] \quad u_{\min} \le u \le u_{\max}, \text{ and } v_{\min} \le v \le v_{\max} \tag{4.72}$$

where u and v are the parameters for defining the surface in the u–v plane.

Surfaces can be broadly divided into analytical and synthetic surfaces. Analytical surfaces are based on wireframe entities discussed earlier. Some of the analytical surfaces are plane surfaces, ruled surfaces, surfaces of revolution, and tabulated cylinders. Synthetic surfaces, also called *sculptured surfaces*, are formed using a given set of data points or curves and include the Bézier, B-spline, and Coons surface patches.

From Eq. 4.72, it can be seen that a general parametric surface can be modelled as a combination of surface patches. Many parametric surface patches are joined together side by side to form a more complicated shape. It is possible to have only one patch or a number of patches. In Fig. 4.63 is seen a surface consisting of two patches. These two patches have their *u* and *v* parameters changed from 0 to 1. The patch topology can be of two types: rectangular and triangular as shown in Fig. 4.64. The triangular patches are more versatile compared to rectangular since they do not have to be restricted to an ordered rectangular array of points.

Synthetic surface representation can be defined using the tensor-product method, rational method, and blending method. The rational method extends the procedure developed with curves for the surfaces, while the blending method approximates a surface by piecewise surfaces.

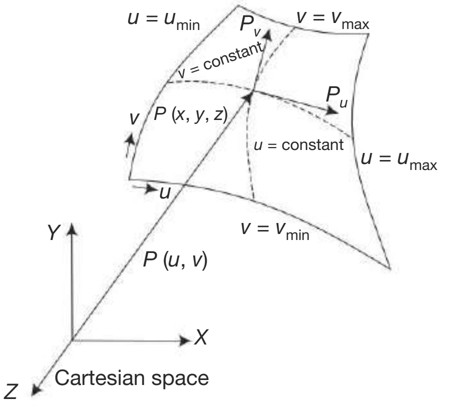


Fig. 4.62 *A general surface representation*

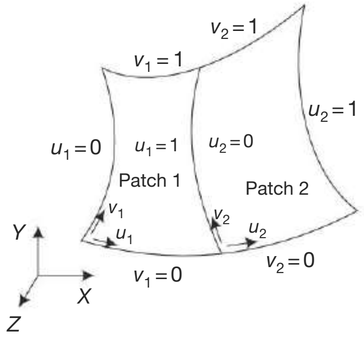
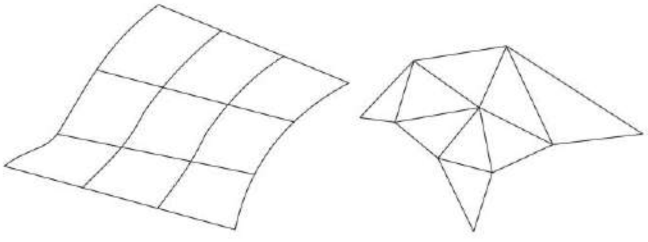


Fig. 4.63 *A composite surface as an assembly of two patches*



(a) Rectangular patches

(b) Triangular patches

Fig. 4.64 *Different types of surface patches*

However, the most popular method used for surface modelling is the tensor-product method. Procedures used are simple and do not add any complication in spite of the higher dimensionality of surfaces compared to curves. Surface properties can be easily deduced from the underlying curve formulations. A rectangular domain described by the *u* and *v* values; e.g., $0 \leq u \leq 1$ and $0 \leq v \leq 1$ is utilised in the tensor-product formulation. It naturally fits rectangular patches due to the explicit unique orientation and special parametric or coordinate directions associated with each independent parametric variable.

In the tensor-product method, the surface is constructed by 'multiplying' two curves. For example, if there are two Bézier curves, it constructs a surface by multiplying the basis functions of the first curve (in *u* direction) with the basis functions of the second curve (in *v* direction) and use the results as the basis functions for a set of two-dimensional control points. Bézier surfaces, B-spline surfaces and NURBS surfaces are all tensor-product surfaces.

To generate a rectangular patch, a set of boundary conditions need to be specified. As shown in Fig. 4.65, there are sixteen vectors (four corner points, *P*; eight tangent vectors, *P′*, two each at the corners; and four

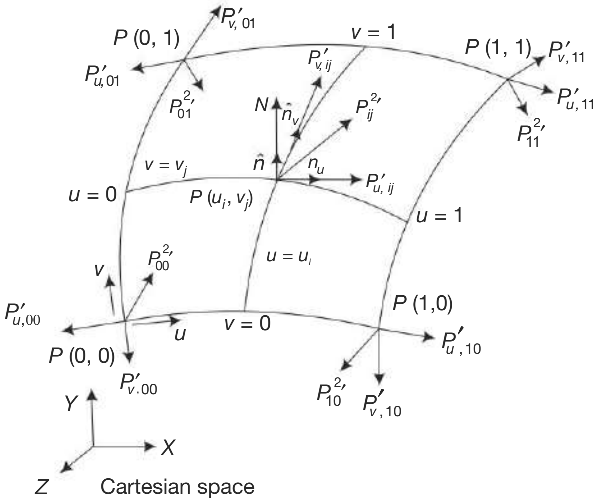


Fig. 4.65 *A parametric surface patch showing all the boundary conditions*

twist vectors, $P^{2'}$ at the corner points) and four boundary curves for a rectangular patch. The boundary curves are specified by the curve equations, $\mathbf{P}(u, v)$; $u = 0$, $u = 1$, $v = 0$, and $v = 1$. These boundary curves are described by holding one parametric variable.

It is important to analyse the surface for utilising these surfaces for different purposes such as CNC tool-path generation. The normal vector to the surface provides the orientation of the cutting tool with respect to the surface. The parametric surface $P(u, v)$ is directly amenable to differential analysis. This can be done by introducing a few parametric derivatives.

The tangent vector at any point $P(u, v)$ on the surface is obtained by the partial derivative of P holding one parameter constant and differentiating with respect to the other. Therefore, there are two tangent vectors for two variables u and v, a tangent to each of the intersecting curves passing through the point as shown in Fig. 4.65. These vectors are given by

$$\text{Tangent vector in } u \text{ direction, } P'_u(u, v) = \frac{\partial \mathbf{P}}{\partial u} = \frac{\partial x}{\partial u}\hat{\mathbf{i}} + \frac{\partial y}{\partial u}\hat{\mathbf{j}} + \frac{\partial z}{\partial u}\hat{\mathbf{k}} \tag{4.73}$$

$$u_{\min} \le u \le u_{\max}, \quad \text{and} \quad v_{\min} \le v \le v_{\max}$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, $\hat{\mathbf{k}}$ are the unit vectors along the x, y and z directions in the Cartesian space.

$$\text{Tangent vector in } v \text{ direction, } P'_v(u, v) = \frac{\partial \mathbf{P}}{\partial v} = \frac{\partial x}{\partial v}\hat{\mathbf{i}} + \frac{\partial y}{\partial v}\hat{\mathbf{j}} + \frac{\partial z}{\partial v}\hat{\mathbf{k}} \tag{4.74}$$

If the dot product of the above two vectors (Eq. 4.73 and 4.74) is zero then these are perpendicular to each other at that point. In Fig. 4.65, the tangent vectors at the corner points of a rectangular patch and at a point P_{ij} were shown. The notation used is $P'_{u,\,ij} = \partial P/\partial u|_{Pij}$, meaning that the derivative is calculated at the point P_{ij} defined by $u = u_i$ and $v = v_j$. Tangent vectors are useful in determining boundary conditions for patching surfaces together as well as defining the motion of cutters along the surfaces during machining processes. The magnitudes of the tangent vectors are given by

$$|\mathbf{P}_u| = \sqrt{\left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2} \tag{4.75}$$

$$|\mathbf{P}_v| = \sqrt{\left(\frac{\partial x}{\partial v}\right)^2 + \left(\frac{\partial y}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial v}\right)^2}$$

The unit vectors of the tangent vectors are given by

$$\hat{n}_u = \frac{\mathbf{P}_u}{|\mathbf{P}_u|} \tag{4.76}$$

$$\hat{n}_v = \frac{\mathbf{P}_v}{|\mathbf{P}_v|}$$

The twist vector at a point on a surface is a second-order partial derivative such as the rate of change of the tangent vector P_u with respect to v which measures the twist in the surface at the point. It can be written in terms of its Cartesian components as

$$\mathbf{P}^{2\prime}(u, v) = \frac{\partial^2 \mathbf{P}}{\partial u\, \partial v} = \frac{\partial^2 x}{\partial u\, \partial v}\,\hat{\mathbf{i}} + \frac{\partial^2 y}{\partial u\, \partial v}\,\hat{\mathbf{j}} + \frac{\partial^2 z}{\partial u\, \partial v}\,\hat{\mathbf{k}} \tag{4.77}$$

$u_{\min} \le u \le u_{\max}$, and $v_{\min} \le v \le v_{\max}$

The normal is used to calculate the cutter offsets for three-dimensional NC programming to machine surfaces, volume calculations, and shading of a surface model. In Fig. 4.65, the surface normal at a point is a vector which is perpendicular to both tangent vectors at the point and is,

$$\mathbf{N}\,(u, v) = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} \tag{4.78}$$

The unit normal vector $\hat{\mathbf{n}}\,(u, v)$, is the cross-product of these partial derivatives:

$$\hat{\mathbf{n}} = \frac{\dfrac{\partial P}{\partial u} \times \dfrac{\partial P}{\partial v}}{\left|\dfrac{\partial P}{\partial u} \times \dfrac{\partial P}{\partial v}\right|} \tag{4.79}$$

The above geometric surface analysis is quite useful in CAM to drive a cutting tool along the surface to machine it and knowing the normal vectors to the surface provides the proper directions for the tool to approach and retract from the surface.

4.8.1 Analytic Surfaces

These are surfaces that can be defined in implicit equation form similar to the curves that were described earlier.

Spherical Surface The Cartesian representation of a spherical surface with radius r, can be defined as a set of points (x, y, z) that satisfy the following equation:

$$x^2 + y^2 + z^2 = r^2 \tag{4.80}$$

The same surface in parametric form can be defined in terms of the angular parameters ϕ and θ (see Fig. 4.66) as

$$\begin{aligned} x &= r\cos\phi\cos\theta, \quad \text{for } -\pi/2 \le \phi \le \pi/2 \\ y &= r\cos\phi\sin\theta, \quad \text{for } -\pi \le \theta \le \pi \\ z &= r\sin\phi \end{aligned} \tag{4.81}$$

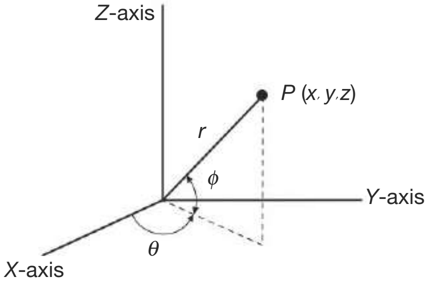


Fig. 4.66 *Cartesian coordinate system with a point specified in polar coordinates*

4.8.2 Surface of Revolution

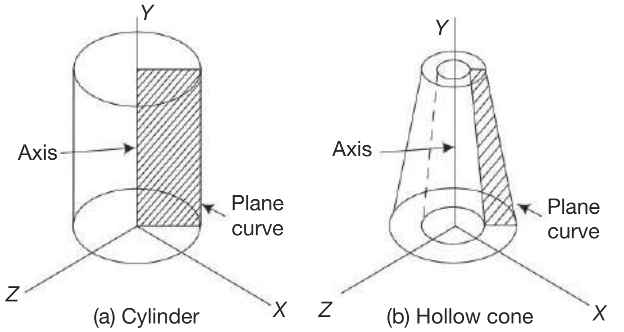
A surface of revolution is generated by revolving a 2D closed curve around an axis. The plane curve is assumed to be in the *xy* plane. As an example, if a rectangular war entity is rotated through an angle of 2π (360°) with an axis coinciding with one of the sides, it will produce a circular cylinder as shown in Fig. 4.67a. The radius of the cylinder is same as the side of the rectangle which is perpendicular to the axis of revolution, while the height of the cylinder is same as the other side of the rectangle. Closed or open polygons can also be used to generate surfaces of revolution. Closed polygons will generate a solid, while open polygons generate a shell surface only. Similarly, an example of a cone with a taper hole is shown in Fig 4.67b. It may be noticed that the axis of revolution is away from the closed polygon used for revolving. Surfaces of revolution can also be generated by rotating plane curves. For example, a sphere is generated by rotating a semicircle in the *xy* plane about the *x*-axis or *y*-axis whose centre is at the origin.

As shown earlier, the parametric equation of the entity to be rotated is

$$P(u) = [x(u)\ y(u)\ z(u)] \tag{4.82}$$

The parametric equation of the revolved surface will be a function of the rotation angle ϕ in addition to u as shown in Fig. 4.68. Hence, any point on the revolved surface is

$$Q\,(u, \phi) = [x(u)\ \ y(u)\cos\phi\ \ y(u)\sin\phi] \tag{4.83}$$



(a) Cylinder (b) Hollow cone

Fig. 4.67 *Cylinder and hollow cone produced by revolving*

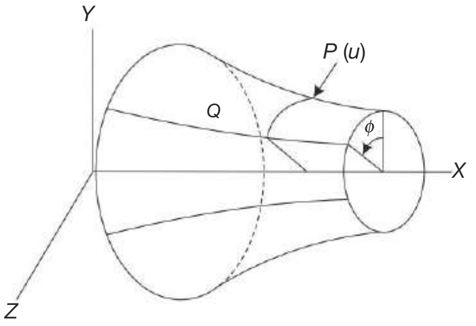


Fig. 4.68 *Surface of revolution of a plane curve P*

Example 4.9 Find the point (0.25, 90°) on the surface of revolution of a line segment with endpoints (1, 1, 0) and (5, 2, 0). This line segment is rotated about the *x*-axis.

Solution The parametric equation for the line segment from P_1 to P_2 is

$$P(u) = [x(u)\ y(u)\ z(u)] = P_1 + (P_2 - P_1)\,u \qquad 0 \le u \le 1$$

From this equation,

$$x(u) = x_1 + (x_2 - x_1)\,u = 1 + 4u$$

$$y(u) = y_1 + (y_2 - y_1)\,u = 1 + u$$

$$z(u) = z_1 + (z_2 - z_1)\,u = 0$$

The parametric equation of the revolved surface is

$$Q(u, \phi) = [x(u)\ \ y(u)\cos\phi\ \ y(u)\sin\phi]$$

The required point is

$$Q(u, \phi) = [1 + 4u(1 + u)\cos\phi\ (1 + u)\sin\phi]$$

$$= [1 + 4 \times 0.25,\ (1 + 0.25)\cos 90°,\ (1 + 0.25)\sin 90°] = [2, 0, 1.25]$$

A few examples of revolved surfaces are given here.

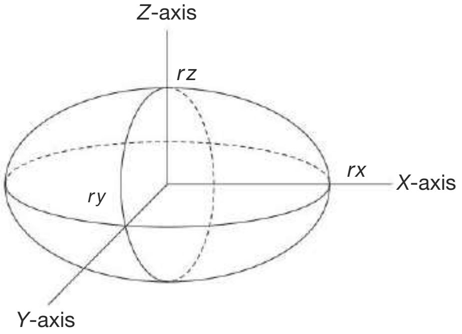


Fig. 4.69 *Ellipsoid*

Ellipsoid An ellipsoid is generated by revolving an ellipse about its central axis. The implicit representation of an ellipsoid centred on the origin as shown in Fig. 4.69 in the Cartesian space is

$$\left(\frac{X}{r_x}\right)^2 + \left(\frac{Y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1 \tag{4.84}$$

And a parametric representation for the ellipsoid in terms of the latitude angle ϕ and the longitude angle θ is

$$\begin{aligned} X &= r_x \cos\phi \cos\theta \quad -\pi/2 \le \phi \le \pi/2 \\ Y &= r_y \cos\phi \sin\theta \quad -\pi \le \theta \le \pi \\ Z &= r_z \sin\phi \end{aligned} \tag{4.85}$$

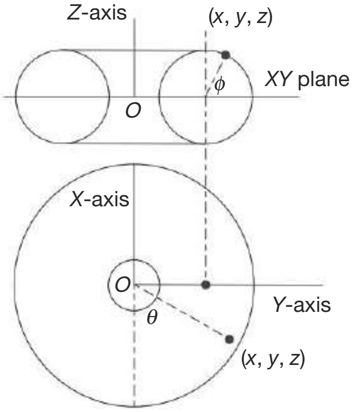


Fig. 4.70 *Torus*

Torus A torus is a doughnut-shaped object (Fig. 4.70). It can be generated by rotating a circle or a conic section about a specified axis. The implicit form of a torus equation in Cartesian coordinates is of the form

$$\left[r - \sqrt{\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2}\right]^2 + \left(\frac{z}{r_z}\right)^2 = 1 \tag{4.86}$$

where r is any given offset value.

Parametric representations for a torus are similar to those for an ellipse, except that the angle ϕ extends over 360°. Using latitude and longitude angles ϕ and θ, we can describe the torus surface as the set of points that satisfy

$$\begin{aligned} X &= r_x (r + \cos\phi) \cos\theta \quad -\pi \le \phi \le \pi \\ Y &= r_y (r + \cos\phi) \sin\theta \quad -\pi \le \theta \le \pi \\ Z &= r_z \sin\phi \end{aligned} \tag{4.87}$$

Table 4.4 *Parametric equations of some quadric surfaces*

Quadric Surface	*Parametric equation*
Ellipsoid	$\left(\frac{X}{r_x}\right)^2 + \left(\frac{Y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1$
Cone	$\left(\frac{X}{r_x}\right)^2 + \left(\frac{Y}{r_y}\right)^2 - \left(\frac{z}{r_z}\right)^2 = 0$
Elliptic cylinder	$\left(\frac{X}{r_x}\right)^2 + \left(\frac{Y}{r_y}\right)^2 = 1$
Hyperbolic cylinder	$\left(\frac{X}{r_x}\right)^2 - \left(\frac{Y}{r_y}\right)^2 = 1$
Parabolic cylinder	$X^2 + Y^2 = 0$
Elliptic paraboloid	$\left(\frac{X}{r_x}\right)^2 + \left(\frac{Y}{r_y}\right)^2 + 2Z = 0$
Hyperbolic paraboloid	$\left(\frac{X}{r_x}\right)^2 - \left(\frac{Y}{r_y}\right)^2 + 2Z = 0$

4.8.3 Ruled Surfaces

As explained earlier, a ruled surface is obtained by joining two or more space curves by means of straight lines (Fig. 4.19). Let us consider two space curves $C_1(u)$ and $C_2(u)$ with the independent parameter u varying between 0 and 1. We can now construct a curve from $C_1(u)$ to $C_2(u)$ with the parameter v varying between 0 and 1 which is called *ruling* or *generator*. Thus, the surface $\mathbf{p}(u, v)$ formed by the rulings between $C_1(u)$ and $C_2(u)$ is called a ruled surface of curves $\mathbf{C_1}(u)$ and $\mathbf{C_2}(u)$, as shown in Fig. 4.71.

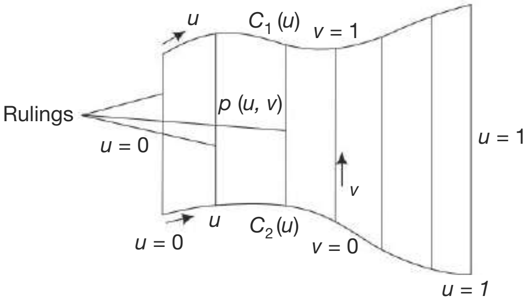


Fig. 4.71 *Ruled surface formed between two space curves $C_1(u)$ and $C_2(u)$*

It is possible to choose any two curves which do not have to be of the same degree with the same number of control points and the same number of knots. To develop the parametric equation of the ruled surface, consider the ruling u_i, joining two points on the rails $\mathbf{C_1}(u)$ and $\mathbf{C_2}(u)$. The equation of the ruling is

$$\mathbf{P}(u_i, v) = \mathbf{C_2} + v(\mathbf{C}_{1i} - \mathbf{C}_{2i}) \tag{4.88}$$

Generalising the above equation, we get

$$\mathbf{P}(u, v) = \mathbf{C_2}(u) + v[\mathbf{C_1}(u) - \mathbf{C_2}(u)] = (1 - v)\,\mathbf{C_2}(u) + v\,\mathbf{C_1}(u)] \tag{4.89}$$

If the value of u is held constant, it will produce the rulings as seen in Fig. 4.71 in the v direction of the surface. It can be easily seen that $\mathbf{C_2}(u)$ and $\mathbf{C_1}(u)$ are $\mathbf{P}(u, 0)$ and $\mathbf{P}(u, 1)$ respectively. The shape of the surface is greatly influenced by the rail that is closer to it. For example, closer to v equal to 0, $\mathbf{C_2}(u)$ will have greater influence. Similarly, as v approaches 1, the influence of $\mathbf{C_1}(u)$ on the surface shape increases. It can also be seen that ruled surfaces cannot be used for modelling doubly curved surfaces, since the surface curvature in the v direction is zero.

4.8.4 Synthetic Surfaces

The analytical surfaces discussed so far are simple, but do not have sufficient flexibility for designers to work with. Synthetic surfaces offer designers with various tools that help in intuitively developing the desired surface shape. There are many design situations where the designers have to get surfaces that pass through a specified set of points in order to achieve the required properties. For example, the profile of an aerofoil to get the required aerodynamic lift or the complex surfaces of injection moulds have to use these surfaces.

Hermite Bicubic Surface Four data points connected by means of a bicubic equation is called a Hermite bicubic surface. To get the full equation, a total of sixteen vector conditions are required. These are the four corner points, eight tangent vectors at two at each of the corner points in the direction of u and v, and the four twist vectors at the corner points. The equation can be written as

$$P(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} C_{ij}\, u^i\, v^j \quad u, v \in [0, 1] \tag{4.90}$$

It is possible to expand it in the matrix form as

$$\mathbf{P}(u, v) = \mathbf{U}^{\mathrm{T}}\,[\mathbf{C}]\,\mathbf{V} \quad 0 \le u \le 1, \text{ and } 0 \le v \le 1 \tag{4.91}$$

where $\mathbf{U} = [u^3 \quad u^2 \quad u \quad 1]^{\mathrm{T}}$,

$\mathbf{V} = [v^3 \quad v^2 \quad v \quad 1]^{\mathrm{T}}$, and

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{C}_{13} & \mathbf{C}_{14} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{C}_{23} & \mathbf{C}_{24} \\ \mathbf{C}_{31} & \mathbf{C}_{32} & \mathbf{C}_{33} & \mathbf{C}_{34} \\ \mathbf{C}_{41} & \mathbf{C}_{42} & \mathbf{C}_{43} & \mathbf{C}_{44} \end{bmatrix} \tag{4.92}$$

Applying the boundary conditions, Eq. 4.89 can be rewritten as

$$\mathbf{P}(u, v) = \mathbf{U}^{\mathrm{T}}\, [M_H][B]\, [M_H]^{\mathbf{T}}\, \mathbf{V} \quad 0 \le u \le 1, \text{ and } 0 \le v \le 1 \tag{4.93}$$

where the basis function is
$$[M_H] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 6 & -6 & 3 & 3 \\ -6 & 6 & -4 & -2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.94}$$

and,
$$[B] = \begin{bmatrix} [P] & [P_v] \\ [P_u] & [P_{uv}] \end{bmatrix} \tag{4.95}$$

The tangent and twist vectors are given by

$$\mathbf{P}_u\, (u, v) = \mathbf{U}^{\mathrm{T}}\, [M_H]^{\mathbf{u}}\, [B]\, [M_H]^{\mathbf{T}}\, \mathbf{V} \tag{4.96}$$

$$\mathbf{P}_v\, (u, v) = \mathbf{U}^{\mathrm{T}}\, [M_H]\, [B]\, [M_H]^{\mathbf{vT}}\, \mathbf{V} \tag{4.97}$$

$$\mathbf{P}_{u\,v}\, (u, v) = \mathbf{U}^{\mathrm{T}}\, [M_H]^{\mathbf{u}}\, [B]\, [M_H]^{\mathbf{vT}}\, \mathbf{V} \tag{4.98}$$

Bicubic surface permits C^1 continuity across the patches and not C^2.

Bézier Surface The Bězier surface is the direct extension of the Bězier curve. Points on a Bězier surface can, therefore, be specified as an extension of the Bězier curve.

$$p(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} p_{ij}\, B_{i,\,m}(u)\, B_{j,\,n}(v) \quad 0 \le u \le 1, \text{ and } 0 \le v \le 1 \tag{4.99}$$

where p_{ij} represents the rectangular array of control points $(m + 1) \times (n + 1)$ defining the vertices of the characteristic polyhedron of the Bězier patch as shown in Fig. 4.72 for 4 × 4 points, and $\boldsymbol{B}_{i,\,m}(u)$ and $\boldsymbol{B}_{j,\,n}(u)$ are the ith and jth Bězier basis functions in the u- and v-directions, respectively which are defined as follows:

$$\mathbf{B}_{\boldsymbol{i},\,\boldsymbol{m}}(\mathbf{u}) = \frac{m!}{i!(m-i)!}\, u^i (1-u)^{m-i} \tag{4.100}$$

$$\mathbf{B}_{\boldsymbol{j},\,\boldsymbol{n}}(\mathbf{v}) = \frac{n!}{j!(n-j)!}\, v^j (1-v)^{n-j} \tag{4.101}$$

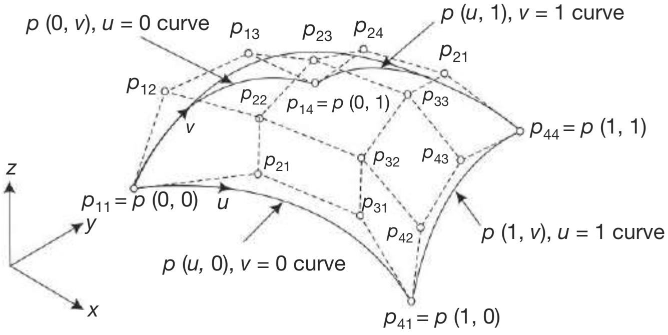


Fig. 4.72 *A bi-cubic Bézier surface patch (4 × 4)*

Since $\mathbf{B}_{i,\, m}(u)$ and $\mathbf{B}_{j,\, n}(v)$ are degree m and degree n functions, the Bėzier surface is of degree (m, n). The set of control points is usually referred to as the *Bézier net* or *control net*. The basis functions of a Bėzier surface are the coefficients of control points. These two-dimensional basis functions are the product of two one-dimensional Bėzier basis functions and, consequently, the basis functions for a Bėzier surface are parametric surfaces of two variables, u and v, defined on the unit square.

The matrix form of this equation for a 4 × 4 control points is

$$p(u, v) = [(1 - u)^3 \quad 3u(1 - u)^2 \quad 3u^2(1 - u) \quad u^3]\, P \begin{bmatrix} (1 - v)^3 \\ 3v(1 - v)^2 \\ 3v^2(1 - v) \\ v^3 \end{bmatrix} \tag{4.102}$$

$$\mathbf{p}(u, v) = \mathbf{U}^{\mathbf{T}}\, [M_B]\, [P]\, [M_B]^{\mathbf{T}}\, \mathbf{V} \tag{4.103}$$

The matrix **P** contains the points that define the characteristic polyhedron.

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_{11} & \mathbf{p}_{12} & \mathbf{p}_{13} & \mathbf{p}_{14} \\ \mathbf{p}_{21} & \mathbf{p}_{22} & \mathbf{p}_{23} & \mathbf{p}_{24} \\ \mathbf{p}_{31} & \mathbf{p}_{32} & \mathbf{p}_{33} & \mathbf{p}_{34} \\ \mathbf{p}_{41} & \mathbf{p}_{42} & \mathbf{p}_{44} & \mathbf{p}_{44} \end{bmatrix} \tag{4.104}$$

$$[M_B] = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 9 & 0 \end{bmatrix} \tag{4.105}$$

$$[U] = [u^3 \quad u^2 \quad u \quad 1]^T, \quad \text{and} \quad [V] = [v^3 \quad v^2 \quad v \quad 1]^T. \tag{4.106}$$

Some important properties of Bėzier surfaces:

- A Bėzier surface passes through the control points at the four corners of the control net.
- **Convex Hull Property** A Bėzier surface lies in the convex hull defined by its control net. The convex hull in this case is formed by connecting the farthest control points on the control polyhedron (Fig. 4.72).
- **Non-negativity** The product of $B_{i,\, m}(u)$ and $B_{j,\, n}(v)$ is non-negative for all m, n, i, j and u and v in the range of 0 and 1.
- **Partition of Unity** The sum of all $B_{m,i}(u)\, B_{n,j}(v)$ is 1 for all u and v.

$$\sum_{i=0}^{m} \sum_{j=0}^{n} B_{i,\, m}(u)\, B_{j,\, n}(v) = 1 \quad 0 \le u \le 1, \text{ and } 0 \le v \le 1 \tag{4.107}$$

- A Bėzier surface is tangent to the corner segments of the control polyhedron.
- The Bėzier surface can be manipulated by changing some vertices of its polyhedron or by keeping the polyhedron fixed and specifying multiple coincident points of some vertices.
- Affine transformation can be applied to a Bėzier surface.
- When the number of control points increases, the degree of Bėzier surface increases, thereby decreasing the local control. This can be compensated by making large surfaces as a combination of small surface patches. This will help reduce the degree of the Bėzier surface patch to a manageable value with local control. However, care has to be taken to see that appropriate continuity is maintained between surface patch boundaries.
- The rational Bėzier patch can be expressed as

$$p(u, v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} p_{ij}\, w_{ij}\, B_{i,m}(u)\, B_{j,n}(v)}{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{ij}\, B_{i,m}(u)\, B_{j,n}(v)} \tag{4.108}$$

B-spline Surfaces The B-spline surface is an extension of the B-spline curve in two parameter definition in u and v, utilising the tensor-product method similar to Bėzier surface. Similar to the Bėzier surface, a rectangular set of control points are used to create the B-spline surface. The degree of the surface is independent of the number of control points. Because of the blending functions used, the continuity is automatically maintained throughout the surface. A B-spline surface with $(m + 1) \times (n + 1)$ control points, is given by

$$P(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} P_{ij}\, N_{i,k}(u)\, N_{j,l}(v) \quad 0 \le u \le u_{\max}, \text{ and } 0 \le v \le v_{\max} \tag{4.109}$$

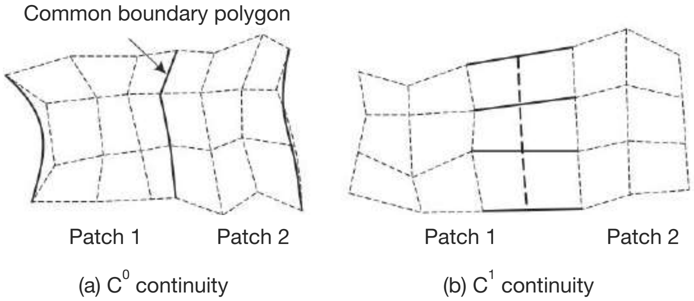
where P_{ij} are the control points and these form the polyhedron of the resulting B-spline surface. The surface also has a degree of $(k - 1)$ in the u direction and $(l - 1)$ in the v direction. Knot vectors in both u and v directions are constant but not necessarily equal. The basis functions are the same as given in Eq. 4.61 for B-spline curves.

Some important properties of B-spline surfaces are the following:

- **Non-negativity** The product of $N_{i,k}(u)$ and $N_{j,l}(v)$ is non-negative for all k, l, i, j and u and v in the range of 0 and 1.
- **Partition of Unity** The sum of all $N_{i,k}(u)$ and $N_{j,l}(v)$ is 1 for all u and v.

$$\sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,k}(u)\, N_{j,l}(v) = 1 \quad 0 \le u \le 1, \text{ and } 0 \le v \le 1 \tag{4.110}$$

- **Convex Hull Property** A B-spline surface lies in the convex hull defined by its control points P_{ij}.
- Composite B-spline surfaces can be generated with C^0 and C^1 continuity. C^0 (positional) continuity requires that a common boundary polygon between the two surface patches as shown in Fig. 4.73a. C^1 continuity (tangency) across the boundaries, the segments attached at the common boundary polygon as in Fig. 4.73a of patch 1 must be collinear with the corresponding segment of the second patch polyhedron, as shown in Fig. 4.73b.



(a) C^0 continuity (b) C^1 continuity

Fig. 4.73 *Continuity of composite B-spline surface patches*

- Affine transformation can be applied to a B-spline surface.
- The weakness of the B-spline surface is that primitive surfaces such as cylinders, spheres and cones cannot be represented precisely. Since it approximates these surfaces, dimensional errors occur when machining these surfaces.

Coons Surface Patch All the surfaces discussed so far rely on a finite set of control points to define a surface, while a Coons surface utilises closed intersecting boundary curves (Fig. 4.20) for interpolating a surface. One of the important applications of the Coons surfaces is the *auto-body styling*. The first step in new auto-body styling is the production of a clay or wooden model of the external shape of the car. This profile as envisaged by the artist is to be communicated to the CAD database for further refining.

This car-body style is digitised using a Coordinate Measuring Machine (CMM) where a probe of the appropriate tip touches the model surface to record a number of points. The probe is moved over the model along certain predefined lines, called *feature lines*. CMM digitises these feature lines into a sequence of points and feeds into the CAD database. The CAD system then fits these points into a network of curves from which a full surface description of the model is generated. For generating the surface from a network of curves, Coons and Gordon surface methods are used.

Shown in Fig. 4.74 is a bilinearly blended Coons patch interpolating the four boundary curves $P(u, 0)$, $P(0, v)$, $P(u, 1)$, and $P(1, v)$. A ruled surface interpolates between two boundary curves. Hence, superposition of two ruled surfaces connecting the two pairs of boundary curves might satisfy the boundary conditions to produce a Coons patch.

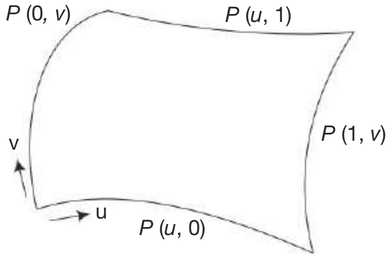


Fig. 4.74 *The four boundary curves of a Coons surface patch*

Utilising Eq. 4.89 for a ruled surface, it is possible to write the two ruled surfaces formed by the two pairs of boundary conditions along the u and v directions as

$$\mathbf{P}_1(u, v) = (1 - u)\,\mathbf{P}(0, v) + u\,\mathbf{P}(1, v)] \tag{4.111}$$

$$\mathbf{P}_2(u, v) = (1 - v)\,\mathbf{P}(u, 0) + v\,\mathbf{P}(u, 1)] \tag{4.112}$$

Adding these two equations gives the required surface as

$$\mathbf{P}(u, v) = \mathbf{P}_1(u, v) + \mathbf{P}_2(u, v) \tag{4.113}$$

Equation 4.113 does not satisfy the boundary conditions which can be seen by substituting $v = 0$ and 1 into this equation.

$$\mathbf{P}(u, 0) = \mathbf{P}(u, 0) + [(1 - u)\,v\,\mathbf{P}(0, 0) + u\,\mathbf{P}(1, 0)] \tag{4.114}$$

$$\mathbf{P}(u, 1) = \mathbf{P}(u, 1) + [(1 - u)\,v\,\mathbf{P}(0, 1) + u\,\mathbf{P}(1, 1)] \tag{4.115}$$

Observing the above two equations, it can be noticed that the terms in the square brackets are extra and need to be eliminated to get the Coons patch equation. The unwanted terms can be conceived as a third surface $\mathbf{P}_3(u, v)$ given by

$$\mathbf{P}_3(u, v) = (1 - v)\,[(1 - u)\,\mathbf{P}(0, 0) + u\,\mathbf{P}(1, 0)] + v\,[(1 - u)\,\mathbf{P}(0, 1) + u\,\mathbf{P}(1, 1)] \tag{4.115}$$

Hence, the required equation is

$$\mathbf{P}(u, v) = \mathbf{P}_1(u, v) + \mathbf{P}_2(u, v) - \mathbf{P}_3(u, v) \tag{4.116}$$

Its matrix form can be given as

$$\mathbf{P}(u, v) = -[-1 \quad (1 - u) \quad u] \begin{bmatrix} 0 & \mathbf{P}(u, 0) & \mathbf{P}(u, 1) \\ \mathbf{P}(0, v) & \mathbf{P}(0, 0) & \mathbf{P}(0, 1) \\ \mathbf{P}(1, v) & \mathbf{P}(1, 0) & \mathbf{P}(1, 1) \end{bmatrix} \begin{bmatrix} -1 \\ 1 - v \\ v \end{bmatrix} \tag{4.117}$$

The main drawback of the above formulation is that it only provides C^0 continuity and not C^1 continuity even when the boundary curves provide the C^1 continuity. In order to provide the C^1 continuity, the linear functions $(1 - u)$ and u are replaced by cubic equations as

$$F_1(u) = 2u^3 - 3u^2 + 1 \tag{4.118}$$

$$F_2(u) = -2u^3 + 3u^2$$

$$P(u, v) = -[-1 \quad F_1(u) \quad F_2(u)] \begin{bmatrix} 0 & \mathbf{P}(u, 0) & \mathbf{P}(u, 1) \\ \mathbf{P}(0, v) & \mathbf{P}(0, 0) & \mathbf{P}(0, 1) \\ \mathbf{P}(1, v) & \mathbf{P}(1, 0) & \mathbf{P}(1, 1) \end{bmatrix} \begin{bmatrix} -1 \\ F_1(v) \\ F_2(v) \end{bmatrix} \tag{4.119}$$

This is termed as a *bicubic Coons surface patch* and is used in the design environment. This formulation allows for describing a much richer variety of surfaces compared to the tensor-product surfaces seen earlier.

Offset Surfaces When a surface is defined by any method, it is possible to generate an offset surface as shown in Fig. 4.75 if the offset amount and the direction vector is specified. The equation for the offset surface can be written as

$$\mathbf{P}(u, v)_{\text{offset}} = \mathbf{P}(u, v) + \hat{\mathbf{n}}(u, v)\, d(u, v) \tag{4.120}$$

where $\mathbf{P}(u, v)$, $\hat{\mathbf{n}}(u, v)$, and $d(u, v)$ are the original surface, unit normal vector on the surface at point (u, v) and the offset distance, respectively.

Blending Surfaces Blending surfaces connect two primary or functional surfaces and provide a smooth transition between the two surfaces as shown in Fig. 4.76. They may also be sometimes called *filleted surfaces*. In order to maintain continuity between the surfaces, blending surfaces require more complex higher-order formulation compared to the underlying surfaces to be joined.

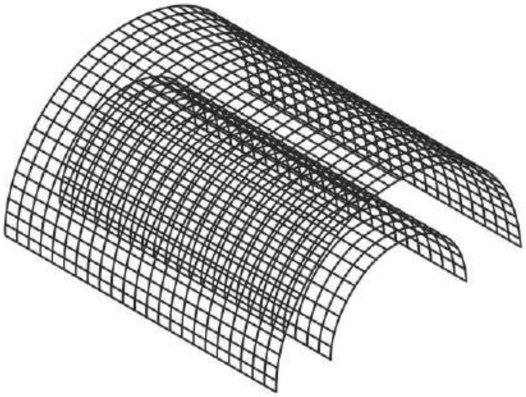


Fig. 4.75 *An offset surface*

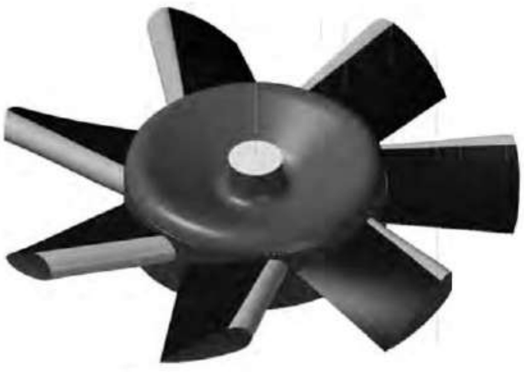


Fig. 4.76 *Blending of surfaces to provide smooth transitions in complex engineering components*

4.8.5 Tabulated Cylinder

When a planar curve is translated along a specified direction, the resulting surface is called a tabulated cylinder. Alternatively, it can also be defined as a surface obtained by moving a straight line called *generatrix* along a given planar curve called directrix as shown in Fig. 4.77. The generatrix stays parallel to the given vector and defines the v direction. Any planar curve $\mathbf{G}(u)$ can be used for generating a tabulated cylinder. The position vector of any point on the surface can be written as

$$\mathbf{P}(u, v) = \mathbf{G}(u) + v\,\hat{\mathbf{n}}_{\mathbf{v}} \qquad 0 \le u \le u_{\max}, \text{ and } 0 \le v \le v_{\max} \tag{4.121}$$

4.8.6 Sculptured Surface

Many types of surfaces have been described so far that are useful for engineering applications. However, it is rare that any one of the types of surface is generally sufficient to provide all the flexibility required for geometric modelling. As a result, in any application a number of different types of surfaces are used, and these surfaces need to be stitched together to form a composite surface. These are often called *sculptured surfaces* or *free-form surfaces*.

A sculptured surface, therefore, can be defined as a complex surface formed as a sum of different types of parametric surfaces and blending surfaces to get the smooth transition across the surfaces. The designer will have to select the most appropriate type of parametric surfaces depending upon the available data and the type of surface required, and then impose the appropriate transitions between the surfaces.

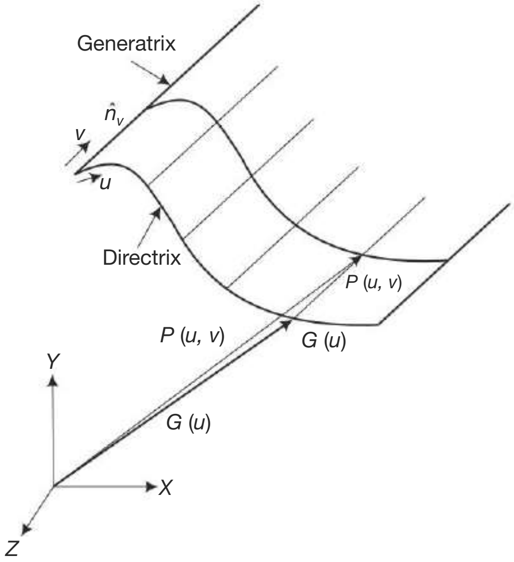


Fig. 4.77 *Tabulated cylinder*

4.8.7 Surface Manipulation

Manipulating the surface during the design phase is important to achieve the desired result. A few of the more common features found in the modern CAD systems are presented here.

Surface Display A surface is normally displayed in a CAD system as a mesh of curves on the surface. The easiest way to do it is to hold one parameter constant and vary the second parameter which will give rise to a curve. For a parametric surface $p(u, v)$, if u is fixed to a value, say 0.05, and v is varied from 0 to 1, this generates a curve on the surface whose u coordinate is a constant. This is the isoparametric curve in the v direction with $u = 0.05$. Similarly, fixing v to a value and letting u vary, it is possible to obtain isoparametric curves whose v direction is a constant. This is sometimes called *wireframe display*. This is a very inefficient type of display, as depending upon the parameter increments used for u and v, some fine details of the surface may be lost or the designer may have to opt for a very fine surface density which calls for a lot of computational time depending upon the complexity of the surface. Sometimes to have a better visualisation, surface normals may be added to the surface in addition to the wireframe display. A better form of visualisation is utilising the shading option as discussed in Chapter 3.

Segmentation Segmentation is a process of splitting a curve or a surface into a number of parts such that the composite curve or surface of all the segments is identical to the parent curve or surface. Segmentation is essentially a reparamterising transformation of a surface while keeping the degree of the surface in u and v parameter space remains unchanged.

For example, let a surface patch be defined in the range of $u_0 < u < u_m$, and $v_0 < v < v_m$. Let this surface patch be divided at a point (u_1, v_1) into four segments, with two divisions along the u direction and two along the v direction. A new variable set is introduced for each of the surface patch segments as (u^1, v^1) whose range is (0, 1) for each of the segments. The parametric transformation for the first segment is

$$\begin{aligned} u^1 &= u_0 + (u_1 - u_0)\,u \qquad 0 \le u^1 \le 1, \text{ and } 0 \le v^1 \le 1 \\ v^1 &= v_0 + (v_1 - v_0)\,v \end{aligned} \tag{4.122}$$

Similar equations can be written for other patch segments.

4.9 SOLID-REPRESENTATION METHODS

As explained earlier, a solid model is a complete model and is generally preferred for complete automation in the design process. A solid model can be more commonly represented internally by one of the following schemes:

- Boundary representation (b-rep)
- Constructive Solid Geometry (CSG)
- Sweeping

Though there are other methods such as analytical solid modelling, half spaces, cell decomposition, octree modelling, etc., for representing solids, sweeping, b-rep and CSG are the more common methods used in a majority of the commercial solid modellers.

4.9.1 Solid-Representation Concepts

In the Euclidean space (E^3), a solid body is represented such that a clear distinction is made as to which part of it is the interior and which part is exterior. Moreover, there will be a surface that separates the interior and exterior. If a solid is defined as a point set S in a three-dimensional space, then it is possible to write it as a union ($\cup$)

$$S = \mathrm{i}S \cup \mathrm{b}S \tag{4.123}$$

where iS is the interior set and bS is the boundary set separating the interior and the exterior.

Equation 4.123 introduces the geometric closure, which implies that the interior of the solid is completely closed by its boundaries.

$$W = \mathrm{i}S \cup \mathrm{b}S \cup \mathrm{c}S \tag{4.124}$$

where W is the universal set (all possible three-dimensional points) and cS is the exterior set.

Equation 4.123 can also be written as

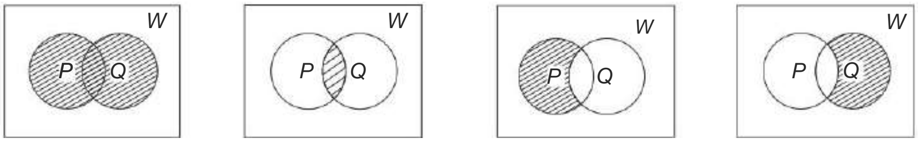
$$S = \mathrm{k}S = \mathrm{i}S \cup \mathrm{b}S \tag{4.125}$$

where kS is the closure of the solid.

The main operations carried out on sets that are used in solid modelling are the following:

Union (see Fig. 4.78a) indicated by $\cup$, e.g., $P \cup Q$ read as P union Q is a subset of elements of W (world) that are members of either P or Q.

$$P \cup Q = \{x: x \in P \text{ or} \in Q\} \tag{4.126}$$



(a) Union ($P \cup Q$) (b) Intersection ($P \cap Q$) (c) Difference ($P - Q$) (d) Difference ($Q - P$)

Fig. 4.78 *Venn diagram for various set operations*

Intersection (see Fig. 4.78b) indicated by $\cap$, e.g., $P \cap Q$ reads as P intersect Q is a subset of elements of W that are simultaneously members of both P and Q.

$$P \cap Q = \{x: x \in P \text{ and} \in Q\} \tag{4.127}$$

Another derived operator is the *difference* (see Fig. 4.78c), indicated by –, e.g., $P - Q$ read as P minus Q is a subset of elements of W that are members belonging to P and not Q

$$P - Q = \{x: x \in P \text{ and} \notin Q\} \tag{4.128}$$

The objects resulting from these operations may lack geometric closure, may be difficult to validate, or may be inadequate for application. Hence, these have to be refined to define the regularised Boolean set operations to avoid impossible solids being generated.

A regular set is defined as a set that is geometrically closed and is introduced in geometric modelling to ensure the validity of objects it represents. The boundary contains the interior and any point on the boundary is in contact with a point in the interior under geometric closure. A set S is regular if and only if

$$S = k\,i\,S \tag{4.129}$$

The set S shown in Fig. 4.79 is not regular because the closure of the interior set $S' \neq S$, i.e., Eq. 4.129 is not satisfied.

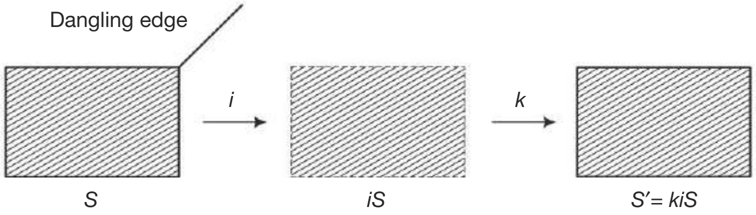


Fig. 4.79 *Regularisation of sets*

Boolean algebra operates with regular sets and is used in geometric modelling. With Boolean algebra, solid models built from well-defined primitives are always valid and represent valid objects. Regularised set operators preserve homogeneity which ensures that no dangling parts should result from using these operators. They also maintain spatial dimensionality ensuring that if two three-dimensional objects are combined by one of the operators, the resulting object is also three-dimensional.

Hence, regularised set operators ($\cup^*$, $\cap^*$, $-^*$) can be defined as follows:

$$P \cup^* Q = ki\,(P \cup Q) \tag{4.130}$$

$$P \cap^* Q = ki\,(P \cap Q) \tag{4.131}$$

$$P -^* Q = ki\,(P - Q) \tag{4.132}$$

where the superscript * indicates regularisation.

P and Q in the above equations can be any arbitrary sets. However, if X and Y are regular sets (true for geometric modelling) then the above equations become

$$X \cup^* Y = X \cup Y \tag{4.133}$$

$$X \cap^* Y = X \cap Y \tag{4.134}$$

$$X -^* Y = k\,(X - Y) \tag{4.135}$$

Set-Member Classification Building solid models require, as explained earlier, the interaction with various regular solids using the Boolean operations. The system, therefore, will have to identify the intersection of the various boundaries and evaluate the parts that should belong to the final solid of interest. The major intersection problems that need to consider are the point/solid, line/solid and solid/solid intersections.

To understand the problem, as seen above, two sets are required to evaluate the intersection problem: a reference set S and the tool set X. Using the same notations as above for the reference set, iS is the interior and bS is the boundary set. The tool geometry X is to be classified against S. The process is called set-member classification.

The set-membership classification function is defined as

$$M[X, S] = (X \text{ in } S, X \text{ on } S, X \text{ out } S) \tag{4.136}$$

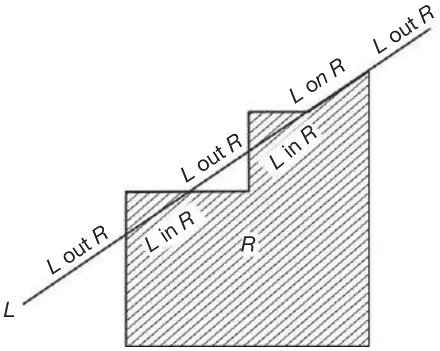


Fig. 4.80 *Set-membership classification–Line/Polygon*

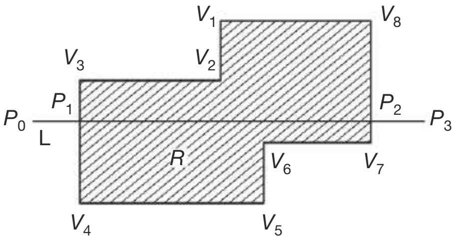


Fig. 4.81 *Set-membership classification–Line/Polygon for b-rep*

Figure 4.80 shows an example of classifying a portion of a line against a polygon R. The actual implementation of the above scheme depends to a great extent on the actual data structures used, and the representation of the sets X and S. A case in point is that used in b-rep as shown in Fig. 4.81. In this example, the line L is chosen such that no 'on' segments are present for the sake of simplicity.

The steps for the selection process (Fig. 4.81) can be indicated here as follows:

1. Utilise a line/edge intersection routine to find the two intersection points P_1 and P_2.
2. Sort all the border crossings as per any agreed sorting criteria for L. Let the list be (P_0, P_1, P_2, P_3).
3. Classify L with respect to R. It is known that odd crossings such as P_1 would start the 'in' segments and even crossings such as P_2 start the 'out' segments. Hence the classification should be

$$[P_0, P_1] \subset L \text{ out } R$$
$$[P_1, P_2] \subset L \text{ in } R$$
$$[P_2, P_3] \subset L \text{ out } R$$

The above scheme of odd and even crossings will not work if the line L coincides with an edge of the polygon. In such cases, traversing along the polygon boundaries would help. In Fig. 4.81, traverse counter-clockwise along the polygon vertices. As a result, the interior of R (iR) is always on the left side of the boundary. An edge of the polygon will be defined as two consecutive vertices V_i and V_{i+1}. An algorithm will flag 'in' when a boundary crosses on an edge whose V_i is above L and V_{i+1} is below L; and whenever V_i is below L and V_{i+1} is above, it is flagged 'out.'

It is possible to develop a similar algorithm for the CSG representation as well.

1. Line/edge intersection routine is utilised to find the intersection points of the line with each primitive of R.
2. Classify the line against each primitive of R by using these intersection points.
3. Use the same Boolean operations as that of the primitives to combine the 'in' and 'on' line segments obtained in Step 2.

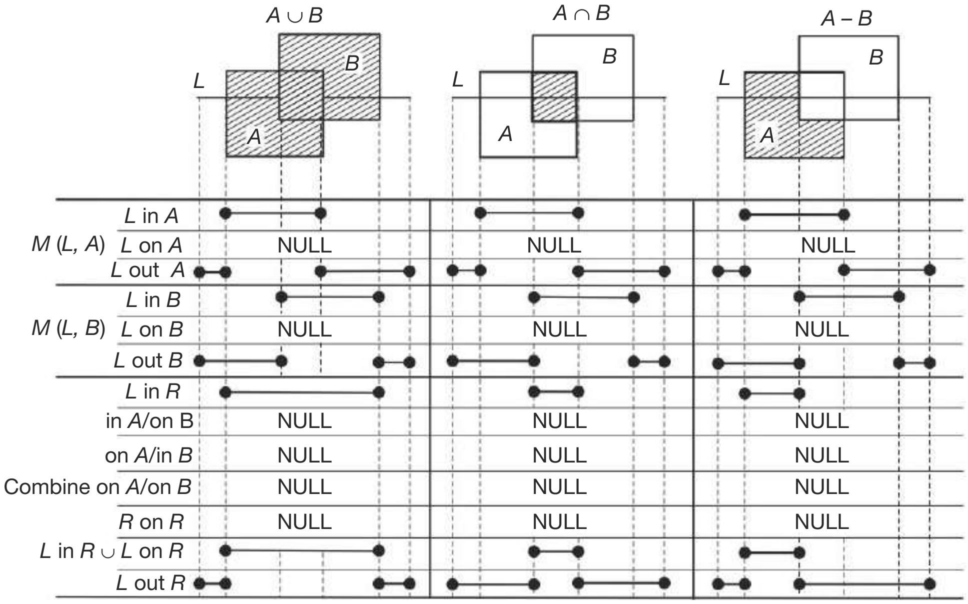


Fig. 4.82 *Line/Polygon classification for CSG rep*

4. The 'out' segments can now be obtained by taking the difference between the line (candidate set) and the 'in' and 'on' segments. The classification strategy for three Boolean operations on two blocks *A* and *B* is shown in Fig. 4.82. The case of union operation in CSG is identical to that of b-rep. First combine *L* in *A* and *L* in *B* to obtain *L* in *R,* using the proper Boolean operator. The *L* on *R* can result from combining three possibilities: *L* in *A* and *L* on *B, L* on *A* and *L* in *B* and *L* on *A* and *L* on *B.* All these possibilities are obtained and then combined to give *L* on *R.* The remaining classification *L* out *R* is obtained by adding *L* in *R* and *L* on *R* and subtracting the result from *L* itself.

4.9.2 Boundary Representation (b-rep)

A b-rep solid is represented as a volume contained in a set of faces together with topological information which defines the relationships between the faces. Unlike wireframe representation, boundary representation (b-rep) is based on the concept that a solid body is bounded by a set of faces as shown in Fig. 4.83. Thus, it is an extension of the wireframe by adding the face information. The main advantage of a b-rep model is that a solid is bounded by its surface and has its *interior* and *exterior* clearly defined. Because b-rep includes such topological information, a solid is represented as a closed space in 3D space. The boundary of a solid separates points inside from points outside of the solid. The geometry of the object can be described by its boundaries, namely, vertices, edges and surfaces. Each face is bounded by edges and each edge is bounded by vertices. Faces can be formed by either straight-line objects or curve segments. Only the boundary surfaces of the model are stored and the volumetric properties are calculated by the Gauss divergence theorem, which relates volume integral to surface integrals. This scheme can model a variety of solids depending on the primitive surfaces (planar, curved, or sculptured).

Some of the definitions of the objects that will be found in b-rep models are the following:

Vertex It is a unique point (an ordered triplet) in space.

Edge A finite, non-intersecting space curve bounded by two vertices that are not necessarily distinct.

Loop It is an ordered alternating sequence of vertices and edges. A loop defines a non-self-intersecting closed space curve, which may be a boundary of a face.

Face It is defined as a finite connected, non-self-intersecting, region of a closed oriented surface bounded by one or more loops. Normally, a face is a bounded region of a planar, quadratic, toroidal, or sculptured surface. The bounded region of the surface that forms the face is represented by a closed curve that lies on the surface.

Genus It is the topological name for the number of handles or through holes in an object.

Body It is an entity that has a set of faces that bound a single connected closed volume. A minimum body is a point.

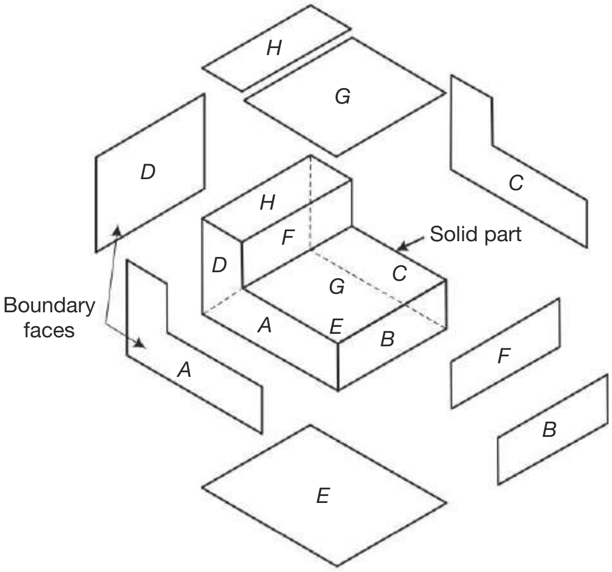
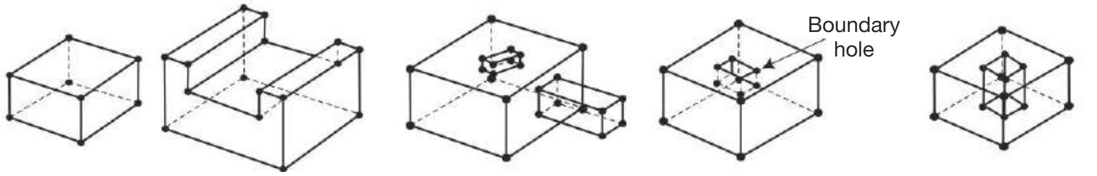


Fig. 4.83 *Representation of a solid by its faces*

The total information present in a b-rep model is classified into *topological* and *geometric* data. The topological part of the data provides the relationships among its objects such as vertices, edges and faces similar to that used in a wireframe model, along with the orientation of edges and faces. Geometric information is usually equations of the edges and faces.

There are two types of solid models in this scheme: (a) polyhedral solids, and (b) curved solids. Polyhedral objects consist of only planar surfaces such as a cube or a tetrahedron. A curved solid on the other hand has curved faces and edges. A few types of polyhedral objects are shown in Fig. 4.84. Polyhedral objects can be classified into four types depending upon the type of features associated with them. Simple polyhedra do not have holes and each face is bounded by a single set of connected edges, i.e., bounded by one loop of edges (see Fig. 4.84a). Polyhedra with faces of inner loops are similar to the first with the exception that a face may be bounded by more than one loop of edges (see Fig. 4.84b). Polyhedral with not through holes may have a face coincident with the object boundary or an interior hole (see Fig. 4.84c). Handles (through holes) in the object are the through passageways as shown in Fig. 4.84d. Developing valid solid models using faces, edges and vertices is rigorous and not easy.



(a) Simple polyhedra

(b) Polyhedra with faces of inner loops

(c) Polyhedra with not through holes

(d) Polyhedra with through holes

Fig. 4.84 *Types of polyhedral objects*

Euler has proved that polyhedra that are homomorphic to a sphere are topologically valid if they satisfy the following equations:

$$F - E + V - L = 2\,(B - G) \tag{4.137}$$

where F, E, V, L, B and G are the number of faces, edges, vertices, face's inner loop, bodies, and genus (handles or through holes) respectively. If the solid is a simple one (shown in Fig. 4.84a), the Euler Eq. (4.138) holds good.

$$F - E + V = 2 \tag{4.138}$$

Example 4.10 Verify Euler's law for the parts shown in Fig. 4.85 and Fig. 4.86:

We see in Fig. 4.85, that the part has six edges, four faces, and four vertices. From Euler's law we have

$$F - E + V = 4 - 6 + 4 = 2$$

which satisfies Euler's rule.

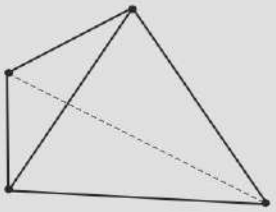


Fig. 4.85 *Example for verifying Euler's law—polyhedron*

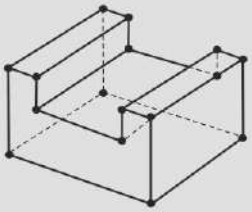


Fig. 4.86 *Example for verifying Euler's law*

We see from Fig. 4.86 that the part has twenty-three edges, nine faces, and sixteen vertices. From Euler's law, we have

$$F - E + V = 9 - 23 + 16 = 2$$

which satisfies Euler's rule.

To store the data in a b-rep model, relational database structure is the most convenient. The list of vertices, edges, loops, faces, and bodies are stored in tables. Winged-edge data structure, as used for data structures, is shown in Fig. 4.87. This is a picture of a single edge, ending in two vertices, which then each have two other edges leading off from them. The edge, for example, could be the edge of a cube. The edge has pointers to the vertices at its ends, and to the next edges. A pointer is essentially the address in the computer's memory where something is stored. The vertices have pointers to their (x, y, z) coordinates, and so on. Since the edge is formed by two faces, it is part of two loops F_1 and F_2. This winged-edge data structure is very efficient for manipulation (addition or deletion of edges, faces or vertices) utilising Euler's law.

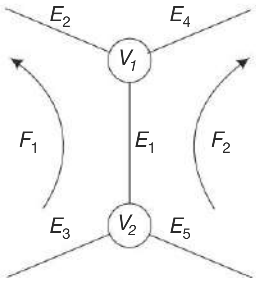


Fig. 4.87 *Winged-edge data structure*

Euler Operators Euler operators are used for building boundary models for complex objects. There are many operators such as MEV, KEV, MBFV, etc. In these operators, M refers to 'Make' and K refers to 'Kill', while other letters have the same meaning as in the Euler equation. A sample of the Euler operators is given in Table 4.5. It may be noted that in the process of editing a polyhedron with Euler operators, some intermediate results may not be valid solids at all.

Table 4.5 *Some sample Euler operators*

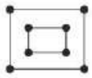
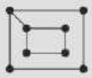
Operation	*Operator*	*Description*
Begin creation of object	MBFV	Make body, face and vertex
Create edges and vertices	MEV	Make edge and vertex
Create edges and faces	MEF	Make edge and face
	MEKL	Make edge, and kill loop
	MEKBFL	Make edge, kill body, face, and loop
Glue	KFEVB	Kill face, edge, vertex and body
Composite operations	KVE	Kill vertex and edge
	MME	Make multiple edges

(a) The Make Group These set of operators are used for adding some elements into the existing model creating a new one, while a Make–Kill operator is used for adding and deleting some elements at the same time. The details of these operators are given in Table 4.6. In this table, the change of values of ***V***, ***E***, ***F***, ***L***, ***B*** and ***G*** with the operators is shown. Note that adding a face produces a loop, the outer loop of that face. Therefore, when ***F*** is increased, ***L*** should also be increased. The geometric changes that occur with the Make operators are shown in Table 4.7. As shown, **MEKL** is adding an edge connecting the outer loop and the inner loop of a face. In this case, the number of edges **E** is increased by 1 and the number of loops **L** is decreased by 1 since that loop is killed.

Table 4.6 *Transition states of Make operators*

Operator	*Meaning*	*V*	*E*	*F*	*L*	*B*	*G*
MEV	Make edge and vertex	1	1	0	0	0	0
MEF	Make edge and face	0	1	1	1	0	0
MBFV	Make body, face and vertex	1	0	1	1	1	0
MBG	Make body and hole	0	0	0	0	1	1
MEKL	Make edge and kill a loop	0	1	0	− 1	0	0

Table 4.7 *Make operators in b-rep models*

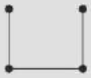
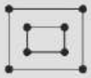
Input	*Operator*	*Meaning*	*Output*
	MEV	Make edge and vertex	
	MEF	Make edge and face	
	MEKL	Make edge and kill a loop	

(b) The Kill Group The Kill group is used to perform the opposite function of what the Make group does. The details of these operators are given in Table 4.8. The geometric changes that occur with the Kill operators are shown in Table 4.9.

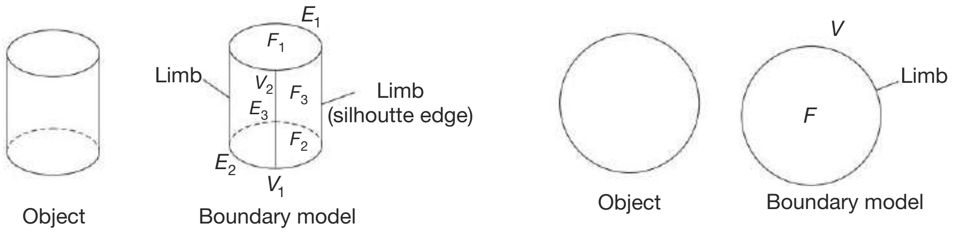
Table 4.8 *Transition states of Kill operators*

Operator	*Meaning*	*V*	*E*	*F*	*L*	*B*	*G*
KEV	Kill an edge and a vertex	– 1	– 1	0	0	0	0
KFE	Kill a face and an edge	0	– 1	– 1	– 1	0	0
KBFV	Kill a body, aface and a vertex	– 1	0	– 1	– 1	– 1	0
KBG	Kill a body and a hole	0	0	0	0	– 1	– 1
KEML	Kill an edge and make a loop	0	– 1	0	1	0	0

Table 4.9 *Kill operators in b-rep models*

Input	*Operator*	*Meaning*	*Output*
	KEV	Kill an edge and a vertex	
	KFE	Kill a face and an edge	
	KEML	Kill an edge and make a loop	

Curved Objects Curved objects such as cylinders and spheres are modelled similar to the polyhedral objects. The major difference to be noted between these two types of objects is the existence of closed curved edges or faces. As shown in Fig. 4.88, a closed cylindrical face has one vertex and no edges. The boundary model of a cylinder has three faces (top, bottom and cylindrical face, itself), two vertices, and three edges connecting the two vertices. The other 'edges' are for visualisation purposes, and are called limbs, virtual edges, or silhouette edges. These models satisfy Euler law: $F - E + V = 2$ for simple polyhedra. In exact b-rep scheme, the curved objects are represented by storing the equations of the underlying curves and surfaces of the object edges and faces respectively. In an alternative arrangement, the curved face is divided into planer facets, which is termed as the approximate or faceted b-rep.



(a) Cylinder

(b) Sphere

Fig. 4.88 *b-rep of a cylinder and a sphere*

Advantages of Boundary Representation

- Complex engineering objects can be modelled very easily compared with CSG. Some examples are aircraft fuselage and automobile body styling.
- Since the topology and geometry are treated separately, incorporating new geometries in the existing model is easy.
- It is particularly suitable for modelling parts having internal symmetry.
- Computational effort and time required to display the model are less compared with CSG.
- Combining wireframe and surface models are possible.
- This format gives efficient picture generation and easy access to other geometric information.
- The b-rep model is more widely used because in CSG, the number of basic primitives available is limited.
- It is easy to create objects by 'sweeping', i.e., a complex two-dimensional profile may be translated or rotated about an axis to give a shape in three dimensions.

Problems with Boundary Representation

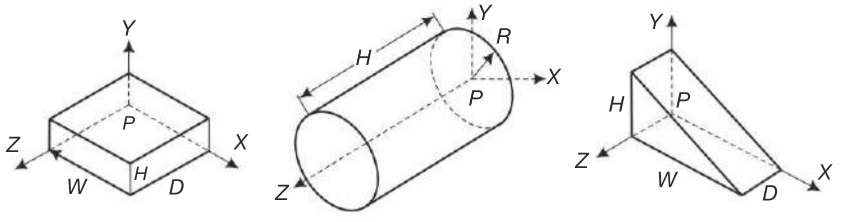
- The data to be stored is more and hence it requires more memory. It is also a verbose scheme. In addition, faceted b-rep is not suitable for many applications such as tool-path generation.
- There is no guarantee that the object created is valid (i.e., complete, unambiguous, and uniquely defined). Additional checks for validity, such as Euler's rule, will be needed to ensure this.
- It is usually less robust than the half-space method.
- Each object is defined independently, without reference to other objects in the system. It is not easy to define 'generic' or 'parametric' models for families of parts.
- Conversion of CSG to b-rep is possible, but conversion from b-rep to CSG is not possible.

4.9.3 Constructive Solid Geometry

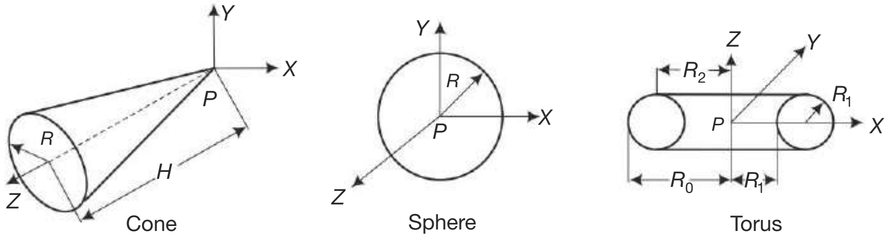
Constructive solid-geometry methods were explained earlier using the basic primitives shown in Fig. 4.13 utilising the Boolean operations. This is one of the most widely researched and understood methodology because of the applications. For example, the intersection operation is useful in understanding the interference problem in assemblies while the difference operator is useful for the material removal processes in CNC tool-path planning. A CSG model is held as a tree structure whose terminal nodes are primitive objects together with an appropriate transformation and whose other nodes are Boolean set operations as shown in Fig. 4.16 with block and cylinder as primitives.

As explained earlier, the data structure for b-rep is based on the winged-edge structure while the CSG representation is based on the concept of graphs and trees. A graph is defined as a set of nodes connected by a set of branches or lines. If the pairs of nodes in a graph are ordered pairs then the graph is called *diagraph*. In a diagraph, branches become arrows indicating the direction of going from one node to another. A CSG tree is called as inverted ordered binary tree, where the leaf nodes are primitives, and the interior nodes are regularised set operations. The total number of nodes in a CSG tree indicates the number of primitives the solid is composed of. If the solid has n primitives then it will have $(n - 1)$ Boolean operations. The CSG tree then will have a total of $(2n - 1)$ nodes.

The mathematical description of all the primitives is stored in the CAD system. The typical primitives (Fig. 4.89) used are



Block Cylinder Wedge



Cone Sphere Torus

Fig. 4.89 *Common primitives used in CSG modelling*

Block $\{(x, y, z): 0 < x < W, 0 < y < H, 0 < z < D\}$ (4.139)

Cylinder $\{(x, y, z): x^2 + y^2 < R^2, 0 < z < H\}$ (4.140)

Wedge $\{(x, y, z): 0 < x < W, 0 < y < H, 0 < z < D, yW + xH < HW\}$ (4.141)

Cone $\{(x, y, z): x^2 + y^2 < [(R/H)\, z]^2, 0 < z < D\}$ (4.142)

Sphere $\{(x, y, z): x^2 + y^2 + z^2 < R^2\}$ (4.143)

Torus $\{(x, y, z): (x^2 + y^2 + z^2 - R_2^2 - R_1^2)^2 < 4R_2^2\, (R_1{}^2 - z^2)\}$ (4.144)

Editing a CSG representation is easy, for example, changing the diameter of the hole in the example above is merely a case of changing the diameter of the cylinder. However, it is slow in producing direct rendered image from a CSG tree. For this purpose, the CSG representation has to be converted to b-rep before rendering. Hence, many solid modellers use internally b-rep but the user interface is based on the CSG representation. An object is stored as a tree with operators at the internal nodes and simple primitives at the leaves. For the object given in Fig. 4.16a, the construction tree is given in Fig. 4.16b. Some nodes represent Boolean operators, whereas others perform translation, rotation, and scaling.

To determine physical properties or to make pictures, it is necessary to combine the properties of the leaves to obtain the properties of the root by following the depth-first tree method. The complexity of this task depends on the representation in which the leaf objects are stored. In some implementations, the primitives are simple solids, such as cubes, cones or cylinders, ensuring that all regularised combinations are valid solids as well. In other systems, primitives include half-spaces, which themselves are unbounded solids. Using half-spaces introduces a validity problem, since not all combinations produce solids.

The notion of a regular set is introduced in geometric modelling to ensure the validity of objects they represent and, therefore, eliminate nonsense objects. A regular set is defined as a set that is geometrically closed. Under geometric closure, a regular set has interior and boundary subsets. The boundary contains the interior and any point on the boundary is in contact with a point in the interior. The main building operators in the CSG scheme are regularised union ($\cup$*), difference ($-$*), and intersection ($\cap$*). These are the set operators which are also called Boolean operators.

The concept of neighbourhood is required to resolve ambiguities when combining 'on' segments in a classification scheme as done earlier with line/polygon classification. $N(P, S)$ is the neighbourhood of a point P with respect to a solid S, and is defined as the intersection of a sphere with radius R (should be sufficiently small) centred at P with the solid. Neighbourhoods for points that are either interior (solid) or exterior (empty) to a solid can be very easily represented. Points on the boundaries of the solid are represented by the face and the surface normal. If the point is on an edge, the edge being shared by two faces, the normal and tangent signs of the faces and the underlying surfaces serve to represent the neighbourhood information.

Evaluating a CSG solid is done by classifying the faces with respect to S by using the face/solid classification and then combining the classifications using an <OP> to obtain the solid. Though face/solid classification is feasible in theory, in practice it is very complex and not attractive. Hence, it is replaced by edge/solid classification which is much simpler. The algorithm is as follows [Zeid]:

1. Generate a sufficient set of tentative faces (*t*-face). The faces of the primitives (A and B) form such a set.
2. Classify self-edges, including neighbourhoods (self-edges of A with respect to A). This is a trivial step, since such classification is already known and is merely used to generate the required data.
3. Using the divide-and-conquer paradigm, classify self-edges of A with respect to B, which include the neighbourhoods as well. The paradigm becomes recursive, if A or B is not a primitive.
4. With the help of desired Boolean operations, combine the classification results of steps 2 and 3 (see Fig. 4.82). The combining classification is

$$\langle \mathrm{OP} \rangle = \cup^*: E \text{ on } S = (E \text{ out } A \text{ INT* } E \text{ on } B) \text{ UN* } (E \text{ on } A \text{ INT* } E \text{ out } B)$$

$$\langle \mathrm{OP} \rangle = \cup^*: E \text{ on S} = (E \text{ in } A \text{ INT* } E \text{ on } B) \text{ UN* } (E \text{ on } A \text{ INT* } E \text{ in } B)$$

$$\langle \mathrm{OP} \rangle = -^*: E \text{ on S} = (E \text{ in } A \text{ INT* } E \text{ on } B) \text{ UN* } (E \text{ on } A \text{ INT* } E \text{ out } B)$$

5. The UN* and INT* are one-dimensional union and intersection operations and are not same as $\cup^*$ and $\cap^*$ respectively. The above combining rules are valid only if there are no on/on ambiguities. If there are, then they should be resolved using neighbourhoods and added. This step gives the classification of self-edges of A with respect to the solid S.

5\. By testing neighbourhoods of the segments, regularise the 'on' segments, by discarding the segments that belong to only one face of S.

6\. Store the final 'on' segments that results from the previous step as part of boundary of S. Steps 2 to 6 are performed for all the *t*-faces.

7\. Find cross-edges that result from intersecting faces of B with the same *t*-face in Step 6 by using surface intersection.

8\. Classify each cross-edge with respect to S by repeating steps 2 to 4. i.e., cross-edges are classified with respect to the faces of A and B. Hence it is two-dimensional classification.

9\. For each cross-edge, repeat steps 5 and 6.

10\. For each *t*-face of A, repeat steps 2 to 9

11\. For each *t*-face of B, repeat steps 2 to 6.

CSG is the most widely used solid modelling representation because of its ability to edit models by deleting, adding, replacing, and modifying subtrees coupled with the relatively compact form in which models are stored. The main disadvantage of CSG is its inability to represent sculptured surfaces and half-spaces.

4.9.4 Half-Space Method

The half-space method considers the geometric entities that divide the representative space into two infinite portions (not necessarily equal), one filled with material while the other being empty. Each such region is called a 'half-space'. By combining half-spaces (using set operations) in a building block fashion, various solids can be constructed. Surfaces can be considered half-space boundaries and half-spaces can be considered directed surfaces.

A half-space is defined as a regular point set in Euclidean space (E^3) as follows:

$$H = \{P: P \in E^3 \text{ and } f(P) < 0\} \tag{4.145}$$

where P is a point in E^3 and $f(P) = 0$ defines the surface equation of the half-space boundaries. A half-space can be combined together using set operations to create complex objects.

The most widely used half-spaces (unbounded, See Fig. 4.90) are

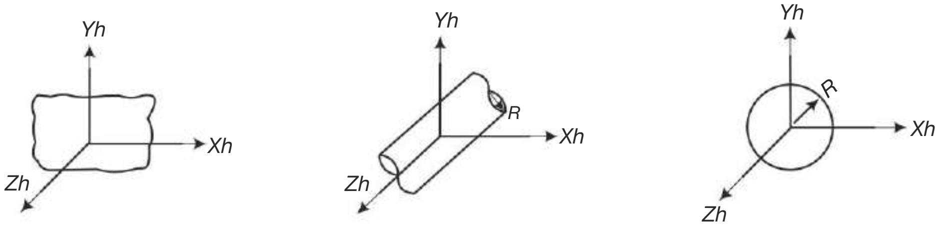
Planar half-space $H = \{(x, y, z): z < 0\}$ (4.146)

Cylindrical half-space $H = \{(x, y, z): x^2 + y^2 < R^2$ (4.147)

Spherical half-space $H = \{(x, y, z): x^2 + y^2 + z^2 < R^2\}$ (4.148)

Conical half-space $H = \{(x, y, z): x^2 + y^2 [\tan(\alpha / 2)\, z]^2\}$ (4.149)

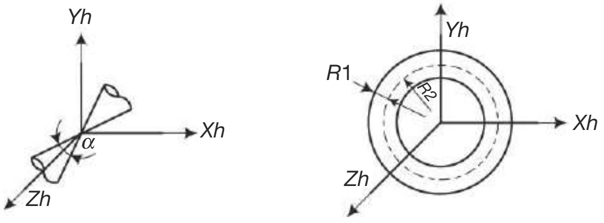
Toroidal half-space $H = \{(x, y, z): (x^2 + y^2 + z^2 - R_2^2 - R_1^2)^2 < 4R_2^2 (R_1^2 - z^2)\}$ (4.150)



Planar half-space

Cylindrical planar half-space ($R > 0$)

Spherical half-space ($R > 0$)



Two-sheet conical half-space ($0 < \alpha < \pi$)

Toroidal half-space ($R1 > 0$, $R2 > 0$, $R2 > R1$)

Fig. 4.90 *Unbounded half-spaces*

To build a complex object, the above half-spaces can be combined using the Boolean operators such as union, difference and intersection, similar to the CSG.

The main advantage of half-spaces is its conciseness in representing objects compared to other methods such as CSG. However, it is very cumbersome for designers to build parts. Also, this representation can lead to unbounded solid models resulting in missing faces and abnormal shaded images. It is useful only for research purposes.

4.10 MODELLING FACILITIES DESIRED

The total modelling facilities that one would look for in any system can be broadly categorised as follows:

- The geometric modelling features
- The editing or manipulation features
- The display control facilities
- The drafting features
- The programming facility
- The analysis features
- The connecting features

4.10.1 Geometric Modelling Features

The various geometric modelling and construction facilities that one should expect to have in any good system are as follows:

1. Various features to aid geometric construction methods, such as Cartesian and polar coordinates, absolute and incremental dimensions, various types of units, grid, snap, object snap, layer, etc.
2. All 2D analytical features, such as points, lines, arcs, circles, conics, splines, fillets, chamfers, etc. In each of these features, various constructional features including interactive and dynamic dragging facilities.
3. Majority of the 3D wireframe modelling facilities including 3D lines, 3D faces, ruled surfaces, linear sweep from 2D topology with any sweep direction, rotational sweep, and tapered sweep, general sweep with twist, revolving about an axis with axis or radial offset for generating helical or spiral shapes.
4. Solid modelling with various basic primitives such as block, cylinder, sphere, cone, prism, torus, pyramid, quadrilateral, along with the ability to apply the Boolean operators on any solid that can be constructed using the other techniques available in the modeller.
5. Skinning around regular and arbitrary surfaces. Profiles (cross-sections), both analytical and arbitrary placed across any 3D curve.
6. Sculptured surfaces of the various types like Bėzier, Coons and other free-form surfaces.
7. Comprehensive range of transformation facilities for interactively assembling the various solid models generated by the modeller with features such as surface filleting and trimming.

4.10.2 Editing or Manipulation Features

These set of facilities refer to the way the geometric data, once created, would be used to advantage for further modelling. Using these facilities, it would be possible to use the geometry created earlier to complete the modelling, thus improving the productivity of the designer. The facilities desired in this category are the following:

— Transformations such as move, copy, rotate, scale, elongate or compress, mirror or to any arbitrary coordinate frame.

— The editing features used to alter the already drawn geometric entities, such as stretching, trimming or trimming to any intersection, delete or erase, undo or redo.

— Symbols in drawing refer to an often-repeated set in a number of drawings, which may consist of a number of geometric entities that are grouped together and stored as a symbol. This symbol can be recalled at any scale, at any angle or exploded if necessary to treat all of them as separate entities. Symbols can also be of parametric type so that a large variation in symbols can be done without much effort.

4.10.3 Display Control Facilities

In this range of features are all the facilities needed for interacting with the modelling system so as to obtain the necessary feedback at the right time during the modelling stage. The facilities required are the following:

Window To identify a set of entities for any possible display or editing function

Zoom To change the scale of display of the image selected on the screen

Pan To move the image on screen without changing the scale at which the drawing is displayed on the screen

Hidden To remove hidden lines or hidden surfaces for viewing the geometry in proper form (Fig. 4.91)

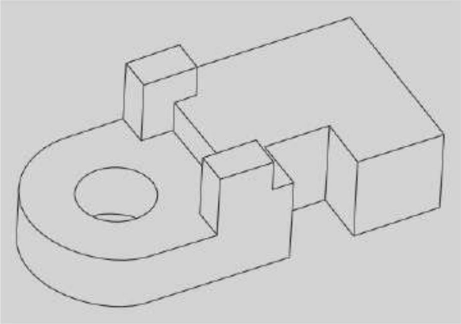


Fig. 4.91 *Elimination of hidden lines in display*

Shading To show the 3D view of the image on the screen complete with the light source location and the resultant light and shade as it appears on the image (Fig. 4.92)

Animation
Orthographic views
Axonometric views
Clipping of images
Perspective views
Isometric views
Sectioning

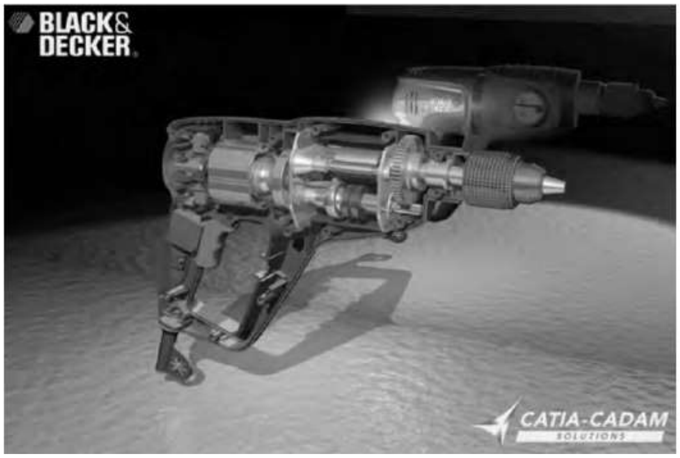


Fig. 4.92 *Shaded image of a CAD geometric model (image appears with the permission of IBM World Trade Corporation/Dassault Systems—Model generated using CATIA)*

4.10.4 Drafting Features

These facilities refer to the way the model developed can be utilised for the purpose of transmitting the information in hard-copy form for other applications, such as part prints onto the shop floor, or maintenance manuals for the equipment. A really large range of facilities are required in this particular category, and it is sometimes treated as a separate module in the modelling system.

The facility to get various types of lines drawn and provide ample notes in the form of text addition at various locations in the drawing should be there. The text-handling capability in terms of font changing and different methods of text presentation should be available.

A large number of types of views should be obtained from the solid model of the geometry stored in the database. The types of views required may be as for display functions, such as perspective views, orthographic views (Fig. 4.93), isometric views, and axonometric views. It is also necessary that the views being shown should be sectioned to get a better appreciation of the model. For this purpose, the section planes may be simple or complex orientations. After sectioning, the system should have the automatic ability to show the sectioning details (Fig. 4.94) in the form of typical cross-hatching depending upon the standard practice.

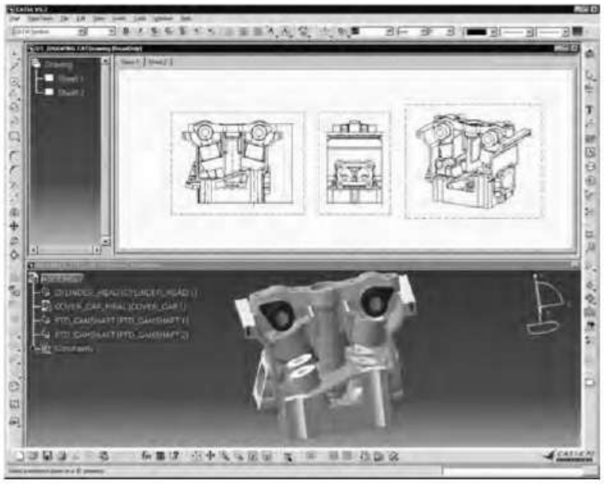


Fig. 4.93 *Orthographic views from a geometric model (image appears with the permission of IBM World Trade Corporation/Dassault Systems—Model generated using CATIA)*

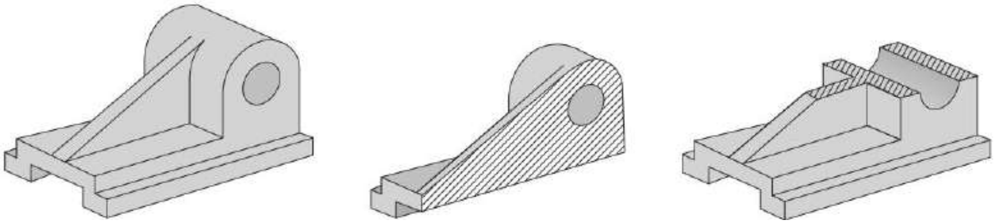


Fig. 4.94 *Section view generation from a geometric model*

The dimensioning facilities one should look at are automatic or semi-automatic dimensioning with associated dimensioning, if possible. In associative dimensioning, the dimension generated is associated with the particular entity and whenever the entity changes during the process of revision or modification, the dimension automatically changes without the operator intervention.

The dimension types that should be available are linear, angular, radius, circular, leader, base line, continuing, dual dimensions, tolerancing, form tolerance symbols, limit dimensioning, and dimensioning to standards (ISO).

4.10.5 Programming Facility

Programming ability (MACRO programming) within a CAD system is a very useful feature. It is well known that not all kinds of facilities are available in any general-purpose CAD system. Therefore, it is necessary that the CAD system is customised for a given range of application processes specific to the company. For this purpose, if a programming facility exists in a CAD system, it is possible to program specifically for an application, making use of all the features available in the system for either modelling or for any specific application based on the information generated during the modelling. Some such examples are the GRIP in Unigraphics and GLUE in CAM-X. The availability of such a program helps the user to input the least amount of information for any required design if the application programs are written well using the programming language.

4.10.6 Analysis Features

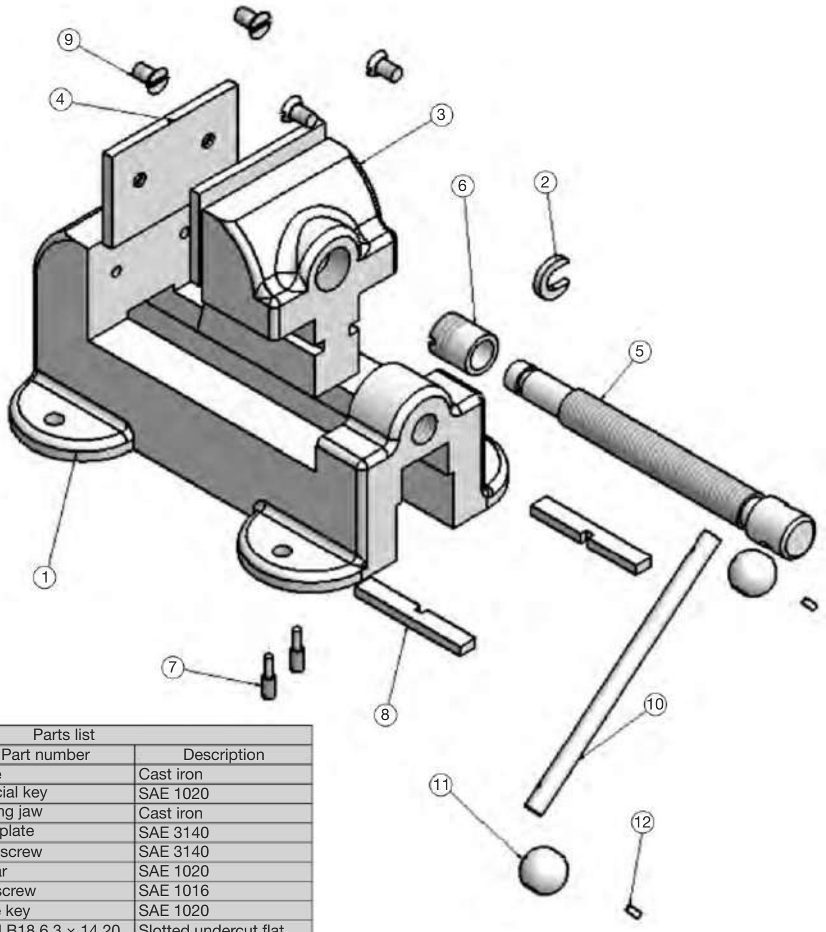
In this range, the kind of analysis facilities that are required to be carried on the product models being generated should be considered. The simplest kind to the most sophisticated features may be available under this category. The simplest facilities may be calculating perimeter, area, volume, mass, centre of gravity, moment of inertia, radius of gyration, etc.

Besides these simple features of analysis, a general-purpose analysis that is normally carried is the Finite Element Analysis (FEA). The geometric model created as above could be conveniently passed onto the FEA through an intermediate processor called a Finite Element Modeller (FEM), which converts the geometric data into the finite element mesh and calculates all the data required for the analysis and then transmits it to the FEA program. Examples are the SUPERTAB for GEOMOD and the GFEM for the Unigraphics.

Another important feature essential in the modelling systems used by the mechanical engineering industries is the assembly facility with the associated interference checking. By this, products individually modelled can be assembled within the modelling system and the interferences or clearances occurring at the assembly joints are analysed. This would be further used along with animation facility, if present, to see the performance of the assembly in service. Along with the assembly facility, the other facility needed is the ability to explode an assembly (Fig. 4.95) for the creation of technical illustrations for the user and maintenance manual preparation.

4.10.7 Connecting Features

Modelling is only the start of the complete process of a product evolution, and as such the data generated is used directly by the other systems. It is, therefore, necessary that the internal data format in which the data is stored by the modelling system should be well documented and should also have very good connectivity (data interfacing) with other allied modules. Ideally, an integrated database structure would be useful wherein all the various modules share the common database. However, this would only be possible if all the modules are developed at a single developer as in the case of ProEngineer or Unigraphics for CAD/CAM integration.



Parts list			
Item	Qty	Part number	Description
1	1	Base	Cast iron
2	1	Special key	SAE 1020
3	1	Sliding jaw	Cast iron
4	2	Jaw plate	SAE 3140
5	1	Vice screw	SAE 3140
6	1	Collar	SAE 1020
7	2	Set screw	SAE 1016
8	2	Slide key	SAE 1020
9	4	ANSI B18.6.3 × 14.20 UNS × 0.4375	Slotted undercut flat countersunk head machine screw
10	1	Handle rod	CRS
11	2	Handle ball	SAE 1020
12	2	ANSI B18.8.2 × No. 4.0 × 14″	Taper Pin

Fig. 4.95 *Exploded view and bill of materials of an assembly modelled*

But in the present days, it is possible that one may choose a special module because of its direct application in a specific instance. And therefore, it is necessary that a neutral data interface standard as described in Chapter 5 is needed with each of the modelling systems so that geometric data may be transferred through this for further processing.

4.11 RAPID PROTOTYPING (RP)

Developing a prototype as a solid representation of the part, without all of the mechanical properties required for the actual product has been used in the manufacturing industry with wood, wax or clay models. The term 'Rapid Prototyping' (RP) is normally used in the current day to specify a series of processes utilising specialised equipment, software and materials that are capable of using 3D CAD design data (as well as 3D scan data such as from a coordinate measuring machine) to directly fabricate geometrically complex objects. The new terms of 'additive fabrication' or 'additive manufacturing' are also being increasingly used for the same technologies. Sometimes these processes are also called *additive processes* since the material is added compared to the conventional processes of material removal that are generally used for prototyping. The first commercial system was demonstrated at the 1987 AUTOFACT show in Detroit utilising the stereolithography process. Though the initial demonstration had a number of limitations, a large number of processes and materials have been developed since then. As a result, this process has been widely adopted by a majority of the manufacturing industries. The use of these processes drastically reduced the development process of a new product with significant cost saving.

Rapid prototyping technologies are generally based upon a Layered Manufacturing (LM) concept. In this method (Fig. 4.96), first, a 3D model of the object as a CAD file is transferred into the system and then sliced into equidistant layers with parallel horizontal planes. The system then generates trajectories for the material to be added in each layer by the RP machine. The sacrificial supporting layers are also simultaneously generated to keep the unconnected layers in proper position. The resultant separate cross-sectional layers of very small thickness when assembled (glued) together will form the final object required.

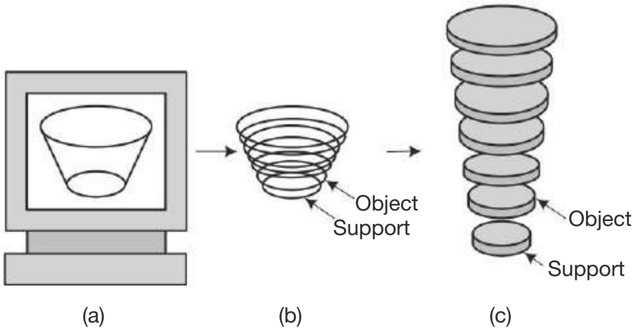


Fig. 4.96 *Concept of layer manufacturing: (a) shape data as input in the CAD system, (b) CAD model is sliced into layers, and (c) each of the layer is then deposited starting from the bottom until the model is completed*

There are a number of ways the 3D CAD design data can be represented. However, the STL (standard triangular language or stereolithography language) format is most common and is generally supported by all the RP equipment. Each physical layer from above is then deposited and fused to the previous layer using one of the many available depositions and fusion technologies.

Computer Numerical Control (CNC) machining can also be considered as rapid prototyping, though it requires custom fixtures and has inherent geometric limitations. Still, machining can be effective in many rapid-prototyping applications.

Advantages

- Product-development time and cost will be greatly reduced compared to the conventional prototyping techniques used.
- By reducing the prototype-development time, the total product design cycle will be shorter, thereby getting products to the market sooner.

- Communications between marketing, engineering, manufacturing, and purchasing are enhanced due to the availability of physical prototypes.
- It is possible to have the physical model at critical design reviews, thereby the decision making process becomes more accurate.
- In some cases, it is possible to perform functional prototype testing before committing to the actual tooling.
- By making available an accurate physical prototype, it is possible to generate precise production tooling.
- Utilising RP technology in metal casting for direct mould-and core-making from CAD files without the need of tooling drastically reduces lead time and cost of produced castings.

Limitations

- The initial cost of the equipment is relatively high.
- The build material choices for different processes are limited in terms of their properties. In some cases, only plastic materials could be used.

Applications

Some areas of the uses of rapid prototyping are

- Checking the feasibility of new design concepts
- Making functional models with the limitation of the material to do any testing
- Conducting market tests/evaluation
- Creating tooling for metal casting (investment casting, permanent mould casting, die casting) injection moulding, and some metal forming processes
- Building a prototype sand or metallic moulds and cores for sand, permanent mould and diecasting
- Fabricating a master pattern for silicon and epoxy moulds
- Manufacture many exact copies of models simultaneously

The current trend of the industrial applications of RP is given in Fig. 4.97. As can be seen, tooling (almost 30%) is one of the major areas of RP applications compared to the other applications. It is also noticed that the percentage of functional parts is increasing over the years with the development of the newer materials and processes.

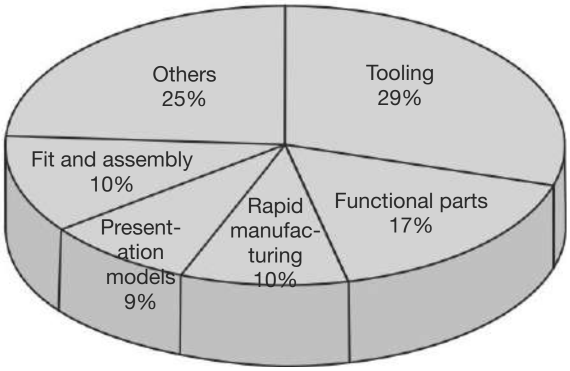


Fig. 4.97 *Industrial applications of rapid prototyping*

4.11.1 Rapid-Prototyping Technologies

The typical steps involved in all current RP techniques, can be summarised as follows:

- A CAD model is constructed and then converted to STL (standard triangular language) format to input into the software for creating the slice data.
- The software processes the STL file by creating sliced layers of the model. The resolution of the built model depends upon the layer thickness.

- The RP device creates the first layer of the physical model. The model is then lowered by the one layer thickness, and the process is repeated until the model is completed.
- The model and any supports are removed; the surface of the model is then finished and cleaned.

A large number of RP techniques have been developed in the past few years. However, the major technologies commercially in use are

- Stereolithography (SLA)
- Selective Laser Sintering (SLS)
- Fused Deposition Modelling (FDM)
- 3D Printing (3DP)
- Laser Engineering Net Shaping (LENS)

Stereolithography (SLA) The most commonly used process for rapid prototyping is stereolithography or *photolithography*. These systems build shapes using light to selectively solidify photocurable resins. A stereolithography machine (Fig. 4.98) converts three-dimensional CAD data of physical objects into vertical stacks of slices. A low-power ultraviolet laser beam is then carefully traced across a vat of photocurable liquid polymer, producing a single layer of solidified resin—the first slice of the object under construction. The laser beam (UV helium–cadmium or argon) is guided across the surface (by servo-controlled galvanometer mirrors), drawing a cross-sectional pattern in the x-y plane to form a solid section. The initial layer is then lowered incrementally by the height of the next slice. A re-coating blade passes over the surface to ensure that a consistent layer thickness is achieved. The re-coating blade was found to be necessary without which air entrapment caused build problems. This procedure is repeated until the entire part is fabricated. On completion, the model is carefully removed and washed in a solvent to remove uncured resin and placed in an UV oven to ensure all resin is cured. Though this was the first process commercialised, it is expensive and is limited to some of the photocurable plastic materials only. However, a large variety of photocurable materials are developed providing a large range of properties as shown in Table 4.10. It is widely used for conceptual visualisation, form and fit analysis, and pattern creation.

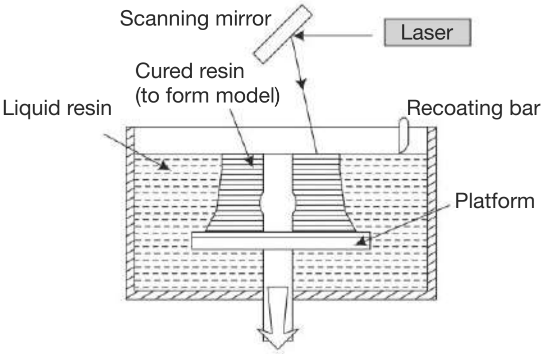


Fig. 4.98 *Schematic of stereolithography device*

Accuracy available with these machines is much higher compared to the other RP machines. As a result, it is widely used for a number of different applications. This resulted in the development of a number of machines to cater to the different applications with different part sizes ranging from 250 × 250 × 250 mm to 650 × 350 × 300 mm.

Selective Laser Sintering In the Selective Laser Sintering (SLS) process originally developed at the University of Texas at Austin, a modulated laser beam follows the shape of a slice of a CAD-generated object; it traces the object across a bin of special heat-fusible powders, heating the particles

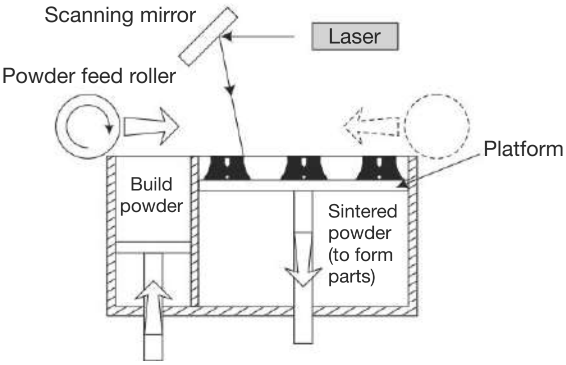


Fig. 4.99 *Schematic of selective laser sintering device*

Table 4.10 *Some materials available for stereolithography (SLA) machines*

Resin	*Appearance*	*Viscosity cps @ 30°C*	*Flexural modulus, MPa*	*Elongation at break (%)*	*Notched Izod impact (J/cm)*	*Heat deflection temperature @0.46 MPa*	*General application*
AccuGen Nd[a]	Clear amber	500	1930	4.6	0.171	—	Prototype parts, master patterns, RTV mould inserts, flow testing, etc.
AccuGen HC and Ar[a]	Clear amber	485	2494 to 2632	4.6 to 5.6	0.202 to 0.245	59°C	Prototype parts, master patterns, RTV mould inserts, flow testing, etc.
Accura 25[a]	White	250	1380 to 1660	13 to 20	0.19 to 0.24	58 to 63°C	Similar to polypropylene, functional components for mockups, master patterns, RTV and silicone moulding, etc.
Accura SI 10[a]	Clear amber	485	2827 to 3102	3.1 to 5.0	0.187 to 0.277	56°C	Investment casting
Accura Amethyst SL[a]	Purple	350	3652 to 3721	0.56 to 1.04	0.009 to 0.012	77°C	High-quality patterns for RTV moulding, design evaluation models, patterns for direct casting
Watershed 11120[b]	Clear, colourless	260	2000 to 2400	11 to 20	0.2 to 0.3	46 to 54°C	Master patterns, investment casting
Somos White[b]	White, opaque	240	2200	8	0.24	53°C	Master patterns, functional parts
ProtoGen O-XT Clear 18120[b]	Clear, colourless	300	2600	6 to 10	0.15 to 0.17	68 to 74°C	High-accuracy master patterns
Proto Cast AF[b]	Green, clear	250	2100	7	0.15 to 0.17	55°C	Very low ash, investment casting

a – ® 3D Systems

b – ® DSM Somos

RTV – Room Temperature Vulcanising

so they fuse or sinter together. In SLS, a layer of powdered material is spread out and levelled in the plane where the layer is to be formed. A CO_2 laser then selectively traces the layer to fuse those areas defined by the geometry of the cross-section along with fusing to the bottom layer (Fig. 4.99). The powders can be joined by melting or surface bonding. The unfused material remains in place as the support structure. After the initial layer is formed, the powder is reapplied, and the laser processes the next layer. Some of the materials used are plastics, waxes and low-melting-temperature metal alloys. Because of the use of metal powders, this process is greatly used in applications such as direct tooling applications for investment and die-casting applications.

A large variety of build materials are available for the SLS systems as shown in Table 4.11. Plastic materials provide increased stiffness, heat resistance and mechanical integrity to make them perfect for functioning prototypes. The materials will have properties similar to ABS and polypropylene such that the parts made with them will have properties that are similar to injection moulding. Some materials also have flexibility similar to rubber so that parts requiring flexibility can also be made with these materials. Metallic materials have properties that are sufficient to use them for making direct metallic parts as well as tool making. They provide good surface finish and excellent machinability.

Table 4.11 *Some materials available for SLS machines*

Material	*Melting point, ºC*	*Tensile strength, MPa*	*Flexural modulus, MPa*	*Specific gravity*	*Notched Izod impact (J/cm)*	*General application*
DuraForm PA®	184	44	1285	0.97	2.14	Prototype plastic parts, patterns for sandcasting and silicone tooling
DuraForm GF®	185	38.1	3300	1.40	1.01	Prototype plastic parts, patterns for sandcasting and silicone tooling
LaserForm A6 Steel®	—	470	138 000	7.8	—	Tooling inserts for injection moulding and diecasting
LaserForm ST-200®	—	250	137 000	6.73	—	Direct metal parts, tooling inserts for injection moulding and diecasting
CastForm PS®	< 63ºC	2.84	—	0.86	0 < 11	Investment casting patterns

® 3D Systems Co.

Large ranges of machines are available with various build volumes. These provide large build area so that large parts or tooling inserts can be easily produced.

3D Printing Originally developed at the Massachusetts Institute of Technology (MIT) in 1993, 3D printing can be compared to SLS; the difference is that instead of a laser beam, liquid binder is applied to bond the powder particles. A 3D printer is operated in the following sequence. The printer spreads a layer of powder from the feed box to cover the surface of the build platform and then prints the binder solution onto the loose powder, forming the first cross-section of the part (Fig. 4.100). Where the binder is printed, the powder's particles are glued together. The

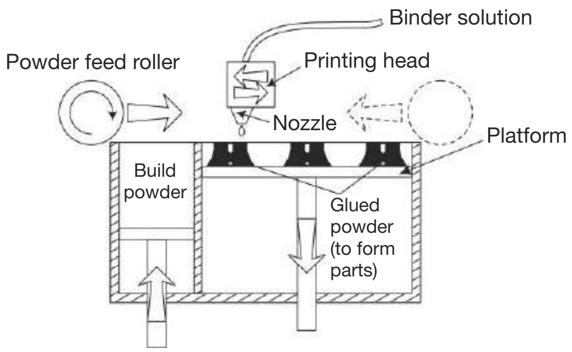


Fig. 4.100 *Schematic of three-dimensional printing device*

remaining powder is loose and supports the part as it is being printed. When the cross-section is complete, the build platform is lowered slightly, and a new layer of powder is spread over its surface. The process is repeated until the whole model is completed. The build platform is raised and the loose powder is removed revealing the completed part.

Dispensing the glue is similar to an inkjet printer; it is possible to print in multicolour to make the built part to have the requisite colours to add for better visualisation. This is a low-cost process compared to the other processes considered so far. However, these parts do not have the necessary mechanical strength, and are used only for the visualisation purpose.

Fused Deposition Modelling In this process (Fig. 4.101), a plastic filament is unwound from a coil and supplies material to an extrusion nozzle. The nozzle is heated to melt the plastic and has a mechanism which allows the flow of the melted plastic to be turned on and off. The nozzle is mounted to a mechanical platform, which can be moved in both horizontal and vertical directions. As the nozzle is moved over the table in the required geometry, it deposits a thin bead of extruded plastic to form each layer. The plastic hardens immediately after being squirted from the nozzle and bonds to the layer below. Several materials are available for the process including investment casting wax. Some FDM systems utilise two extrusion nozzles—one for the deposition of a build material, and second for deposition of washable material to make support environment. In one of this techniques, a temporary support structure is dissolved with water jets, rather than removing it by hand or with a chemical solvent.

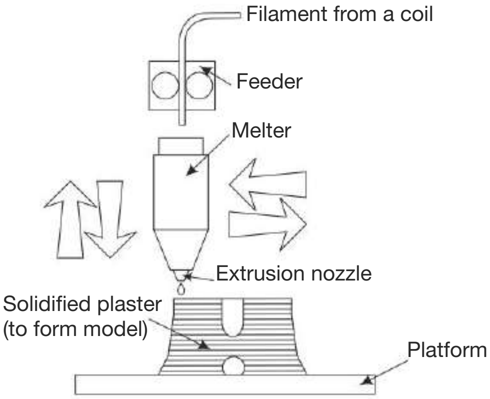


Fig. 4.101 *Schematic of fused deposition modelling device*

A large range of FDM materials are available that include ABS (Acrylonitrile Butadiene Styrene), polycarbonate, polypropylene, PMMA (polymethyl methacrylate), and various polyesters. ABS is by far the largest used material in FDM. Some FDM materials and their properties are shown in Table 4.12.

Table 4.12 *Some materials and their properties used in FDM machines to make prototype parts and tooling*

Material	*Appearance*	*Tensile strength, MPa*	*Flexural modulus, MPa*	*Specific gravity*	*Notched Izod impact (J/cm)*	*Heat deflection temperature @ 0.46 MPa*
Polyphenylsulfone	Tan (silk)	55	2206	1.28	0.5873	189°C
ABS	Black, Blue, Green, Red, Yellow, Grey	22	1834	1.05	2.14	96°C
Polycarbonate	White	52	2137	1.2	0.5339	127°C

One of the major problems with FDM machines is the surface finish of the built part. Since the build material is melted and extruded through the nozzle, the minimum feature size that can be expected is about 0.4 to 0.6 mm, while with SLA it is possible to get as small as 0.08 to 0.25 mm. Another problem associated is the removal of support structure material, particularly for parts with complicated interiors. Since the water-soluble support structure material is available, this problem is taken care of. Also, since these parts require very little cleaning and no postprocessing, it is much preferred for functional parts. Also, the materials provide better stability for the part dimensions with time and environmental exposure. This process is widely used for concept models, form, fit and function models, along with patterns for the creation of moulds and tooling. The size ranges of the machines range from a low build envelope of 200 × 200 × 300 mm to a high of 600 × 500 × 600 mm.

Laser Engineering Net Shaping (LENS) So far, this is the most advanced process in terms of the level of achieved mechanical properties of generated metallic parts among all commercialised processes based on layered manufacturing build principle.

The process uses a high-power laser focused onto a substrate to create a molten puddle on the substrate surface (Fig. 4.102). Metal powder is then injected in the stream of an inert carrier gas into the melt pool to increase its volume. The powder ejection head moves back and forth according to the geometry of the first layer. After the first layer is completed, new layers are then built upon it until the entire object represented in the three-dimensional CAD model is reproduced. Employment of a substrate makes this process different from others considered so far.

This method can utilise a wide range of metals and alloys (including super alloys) as build materials. Relatively high cost of operation and of produced parts on the one hand, and very high mechanical properties generated by this method objects on the other, do not allow considering the method as a plain RP technique or as a means of visualisation. This technology became efficient only in case of functional parts or tooling production.

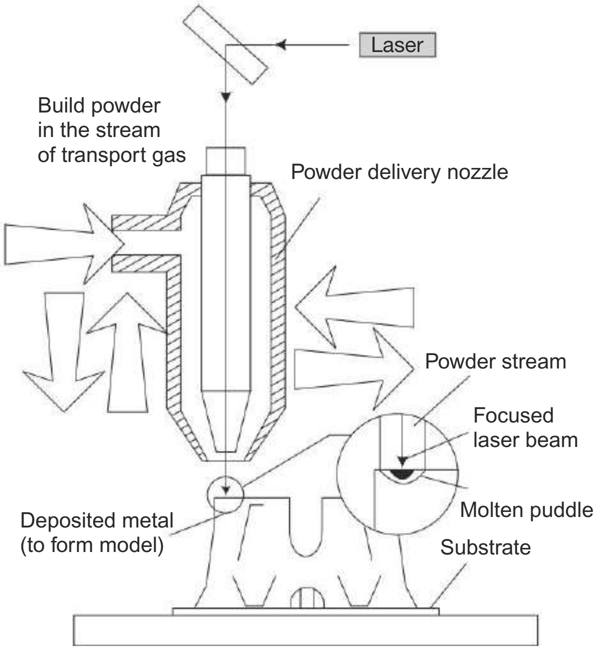


Fig. 4.102 *Schematic of Laser Engineering Net Shaping (LENS) device*

Some of the uses of rapid prototyping are

- Checking the feasibility of new design concepts
- Conducting market tests/evaluation
- Assessing the fit of complex mechanisms
- Promoting concurrent product development
- Making many exact copies simultaneously
- Making moulds for wax cores in castings
- Use as a master for silicon and epoxy moulds

A comparative evaluation of these processes is given in Table 4.13.

Table 4.13 *Summary of RP Technologies*

System	*Max. build size (mm)*	*Accuracy (mm)*	*Comparative costs*	*Comparative build time*	*Materials*	*Advantages*	*Disadvantages*
Stereo-lithography	508 × 508 × 600	0.1 – 0.2	1.0	1.0	Liquid photosensitive resins	High accuracy, medium range of materials, large build size	High cost process, support structures needed, post cure required
Selective Laser Sintering	380 × 330 × 420	0.1 – 0.2	1.0	1.0	Nylon based materials, elastomer, rapid steel, cast form, sand form	Large range of materials, good accuracy, large build size	High cost process, poor surface finish
Fused Deposition Modelling	254 × 254 × 254	0.1 – 0.2	0.3	1.0	ABS, elastomer and wax	Good accuracy, functional materials, medium range of materials, office friendly	Support structures needed

Summary

- Geometric modelling plays a crucial part in the overall application of CAD/CAM systems in manufacturing industries. However, it is important to consider a number of factors before finalising the selection of a CAD/CAM system that suits a given purpose.
- Information entered through geometric modelling is utilised in a number of downstream applications such as drafting, manufacturing, inspection and planning.
- Geometric models are three types, viz., line model, surface model and solid model. Line model, though simple, is rarely used because of the ambiguity present. Surface and solid models are extensively used in industrial applications.
- Among the geometric construction methods, sweep or extrusion is most widely used, because of its simplicity and elegance in developing 3D models.
- Solid modelling provides the most unambiguous representation of the solid model, but is more computing intensive. However, to get the correct geometric model, it is essential to utilise solid modelling approach.
- Surfaces are more widely used and it is necessary to use different types of surfaces such as B-splines, Bézier, NURBS, lofted, to get the user requirements fulfilled.
- Constraint- or parametric-based modelling is the main methodology used by most of the 3D CAD systems. This system helps in grasping the designer’s intent and greatly facilitates the modification and reuse of the existing designs.
- Some variant modelling systems are used based on tabular data for specific applications.
- Form features is another form of modelling system that helps in designing CAD systems with more intelligence built into the geometric entities that is possible by purely geometric systems discussed thus far.

- The mathematical representation of the geometric entities can be in implicit or parametric form, the latter being the preferred method used in CAD systems because of its easier adaptation in software development.
- Solid modelling utilises the b-rep and CSG schemes for maintaining the data internally. b-rep is more common because of the many advantages found for its formulation.
- The curve-representation methods can be extended for surface representations such as used in free-form surfaces.
- A number of modelling facilities need to be considered while selecting a CAD/CAM system for any given application. The facilities that should be considered are the geometric modelling features, the editing or manipulation features, the display control facilities, the drafting features, the programming facility, the analysis features, and the connecting features.
- Rapid prototyping is used to generate the product directly from the 3D CAD model data. A number of different processes such as stereolithography, selective laser sintering, 3D printing, fused deposition modelling, laminated object manufacturing, are used for this purpose.

Questions

1. Specify the range of applications for which typical geometric modelling information is used.
2. What do we expect a geometric modelling system to accomplish, in a broad sense, in the total manufacturing scene?
3. What should be the basic requirements of geometric modelling such that the data generated would be unambiguous?
4. How do you classify the various geometric modelling systems based on their capabilities?
5. Specify the three principal classifications of the geometric modelling systems and write in brief about each of them.
6. Give the reasons for the importance of 3D geometry in modern CAD systems.
7. Why is it important to remove hidden lines and surfaces from 3D geometry?
8. Explain the different types of database organisation methods used in 3D CAD systems.
9. Why do you consider studying geometric modelling is important in relation to CAD in the manufacturing industry?
10. What do you understand by geometric entities in relation to CAD? Explain a few of the types in which these entities can be defined.
11. What are the limitations found in the general wireframe modelling systems? Explain with an example.
12. What is the best kind of a modelling system? Explain with suitable sketches.
13. What are the various three-dimensional construction methods suitable for mechanical-engineering applications?
14. Take any typical 3D geometry for an engineering component. Show how it can be represented in CAD by wireframe modelling.
15. Compare 2D and 3D wireframe modelling with respect to their utility for an engineering industry.
16. Explain different types of geometric modelling methods used in CAD. Give a comparative application of each of them.
17. What is meant by sweep? Discuss in detail the various types of sweep techniques available for 3D geometric construction.
18. What are the limitations in utilising the sweep method for geometric construction?
19. Explain the concept of the three basic Boolean operations used in solid modelling. Give neat sketches showing the effect of these operators on any two basic primitives.

20. Give a classification of the different surfaces that can be used in geometric modelling applications.
21. Explain the Coons and Bėzier surfaces. What are the differences and applications for which these are used?
22. Write a short note on NURBS bringing out their important advantages.
23. Write a short note on B-splines bringing out their important advantages.
24. Explain the importance of parametric representation of curves. Why is it more used compared to non-parametric (implicit) representation?
25. What do you understand by the non-parametric (implicit) representation of curves? Why is it less used compared to parametric representation in CAD?
26. Explain what is meant by a synthetic curve. Give some types of synthetic curves and their applications.
27. Explain any one of the curve-fitting techniques that is relevant in CAD application.
28. What are the different types of curve-fitting techniques used? Compare them from the curve design point of view.
29. How is the curvature of a closed curve obtained? Find the radius of curvature and curvature of a circular cylinder using the parametric representation.
30. Describe the method of defining Bėzier curve. Give some of its advantages in CAD application.
31. What are the continuity conditions that are required for a B-spline surface patch?
32. Compare Bėzier curve and B-splines for CAD application.
33. Specify the parametric and implicit equations for the following surfaces:
    (a) Sphere (b) Torus
    (c) Ellipsoid
34. Explain about reparametrisation of a surface.
35. How is a surface patch subdivided?
36. What is a tabulated cylinder? Give its parametric form.
37. Give the parametric representation of a ruled surface. What are its applications?
38. Explain why Bėzier surfaces are used in creating automobile bodies. Give their formulation.
39. What are the specific properties of Bėzier surfaces that make them useful for complex surface creation?
40. Give a brief description about the spline curves.
41. How do you model an object, which is irregular, as the blade of a propeller-type windmill?
42. Briefly explain the variant method of geometric construction. Explain its applications and limitations.
43. What do you understand by the form element method of geometric construction? Specify the applications of this method of modelling in comparison to that of the variant type.
44. What are the problems with Bėzier curves? How are these taken care of?
45. What is a classification algorithm? Give an example for CSG representation.
46. What are the most common primitives used in solid modelling? Give their parametric equations.
47. What are the advantages of using a B-rep scheme?
48. For CSG scheme, give the classification algorithm.
49. Give the method of representing curved edges in b-rep scheme.
50. Compare CSG and b-rep schemes giving their relative advantages.
51. Specify the range of facilities desired in any general-purpose modelling system.
52. Explain the range of modelling facilities desired in a system suitable for predominantly mechanical engineering oriented with mass production.
53. Specify the drafting features that one should consider for a modelling system required in a CIM environment.
54. What is the importance of a programming language within a modelling system?
55. Why is rapid prototyping used?
56. Give the details of any one rapid-prototyping process you are familiar with.

Problems

1. Write down the implicit form of a line passing through (50, 50) as its midpoint.
2. Find the intersection points of a line passing through (30, 30) and (100, 80) and a circle with centre at (100, 50) and a radius of 75 mm.
3. Find the equation for a line passing through (80, 60) and (30, 30). Find the equation of a line that is perpendicular to the above line and passing through a point (60, 30).
4. Find the equation for a line passing through (80, 60) and (30, 30). Find the equation of a line that is parallel to the above line and passing through a point (40, 50).
5. Represent a circle with centre (0, 0) and a radius of 50 mm through the implicit form as well as parametric form.
6. Write down the implicit and parametric form of circle with centre at (50, 50) and a radius of 100 mm.
7. Find the radius and centre of a circle that is passing through three points (30, 30), (60, 30) and (50, 40).
8. Find the intersection points of a line whose equation is given by $Y = 5X + 30$, and a circle with centre (0, 0) and a radius of 65 mm.
9. Two lines are passing through (20, 30), (120, 80) and (80, 20), (40, 90). Find the equation of a circle that is tangent to the above lines and having a radius of 40 mm.
10. Find the equation of a line that is tangent to a circle whose equation is $x^2 + y^2 = 25$ and passing through the point (8, 1).
11. Find the equation of a Bézier curve which is defined by the four control points as (80, 30, 0), (100, 100, 0), (200, 100, 0), and (250, 30, 0).
12. A cubic Bézier curve is defined by the control points as (30, 30), (50, 80), (100, 100), and (150, 30). Find the equation of the curve and its midpoint.
13. Using any of the CAD package available in your laboratory, create the models shown in Fig. 4.103 to 4.108.

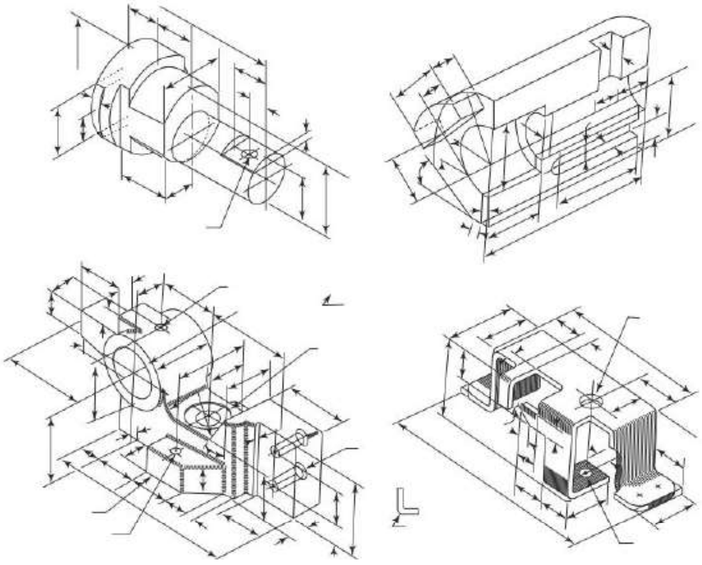


Fig. 4.103

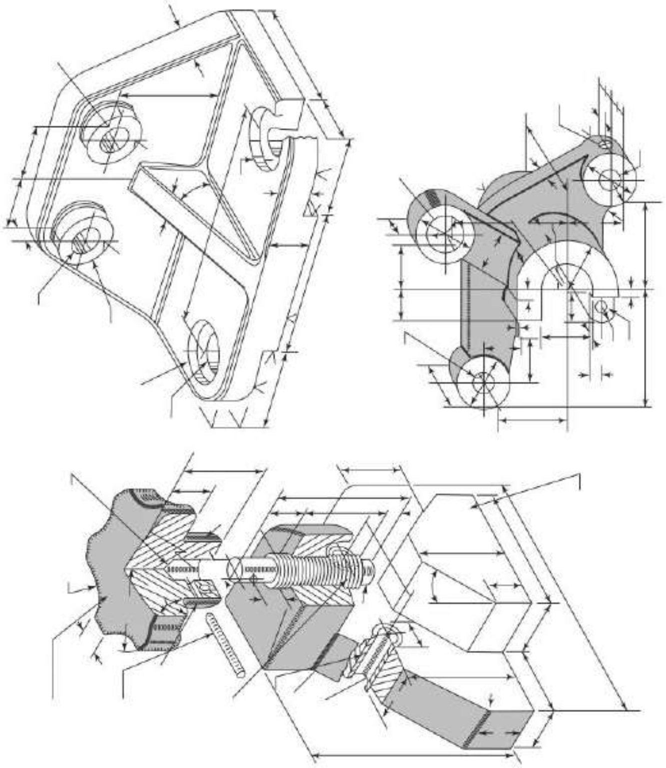


Fig. 4.104

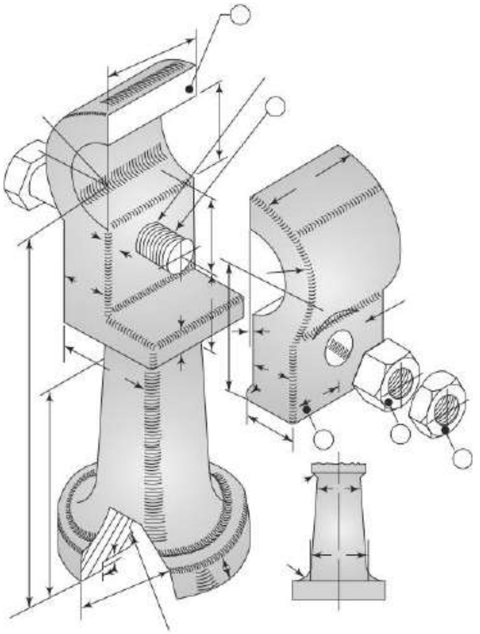


Fig. 4.105

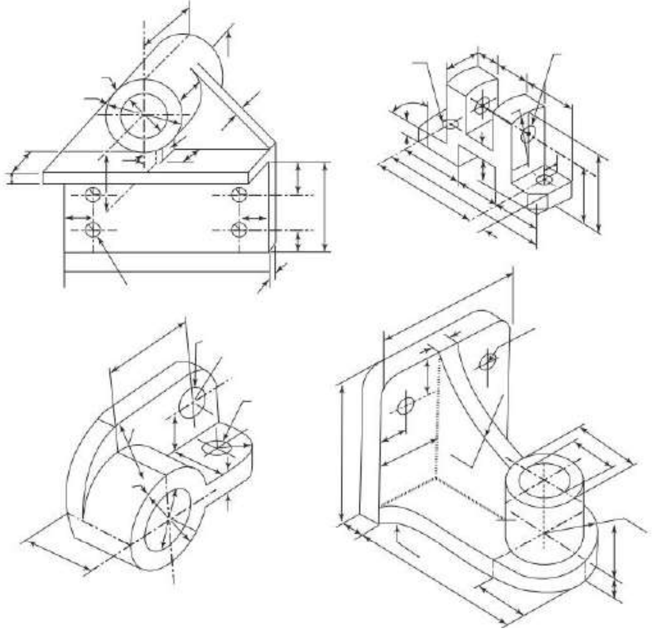


Fig. 4.106

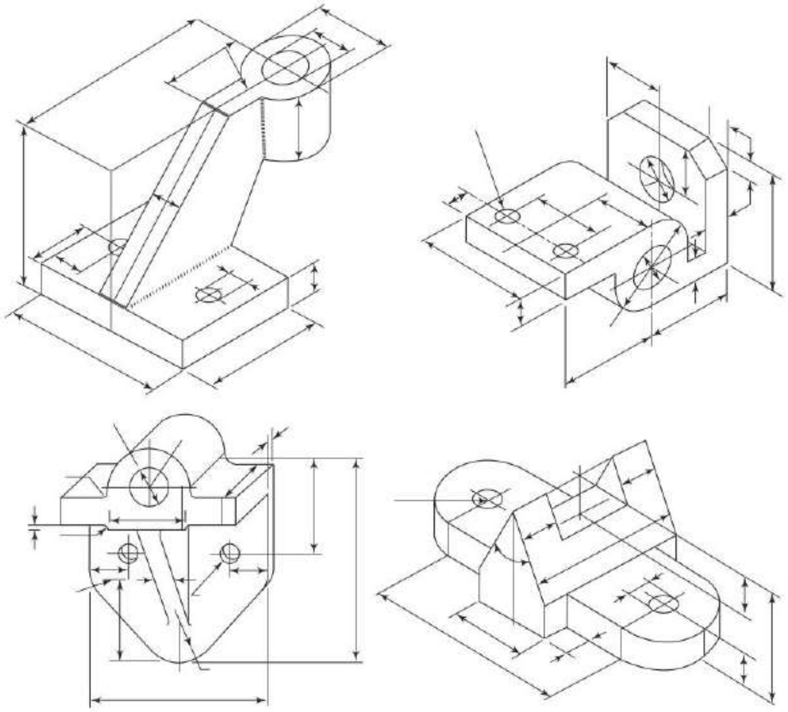


Fig. 4.107

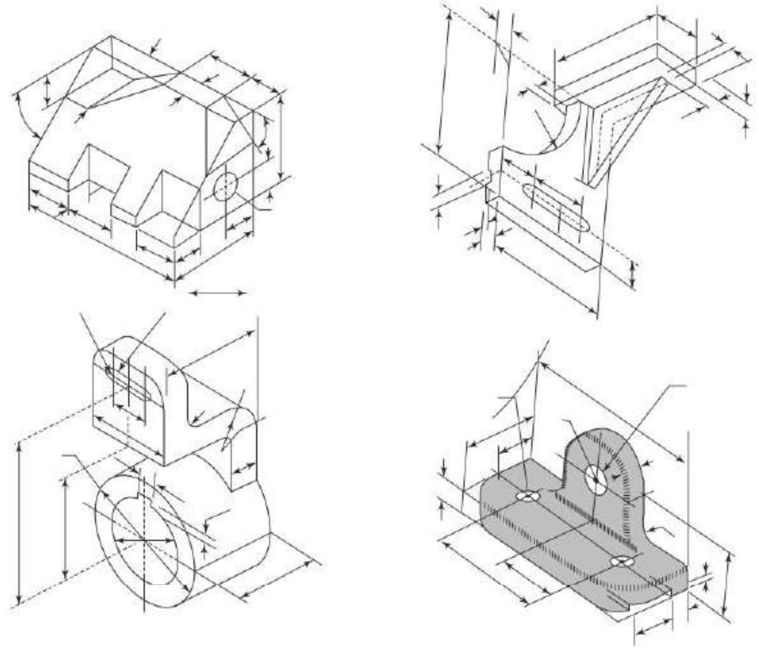


Fig. 4.108

5

CAD STANDARDS

Objectives

A large number of applications are used in CAD/CAM, which are manufactured by different vendors. Therefore, there is a need to establish standards in CAD that help in linking different hardware and software systems from different vendors to be integrated to serve the requirements of the industry. After completing the study of this chapter, the reader should be able to understand, the

- Need for CAD data standardisation
- The graphic kernel system and its extensions for developing the graphic software systems
- Requirements of graphic data exchange formats and their details such as IGES, DXF and STEP
- Dimensional measurement interface specification for communication between coordinate measuring machine and the CAD data.

5.1 STANDARDISATION IN GRAPHICS

With the proliferation of computers and software in the market, it became necessary to standardise certain elements at each stage, so that investment made by companies in certain hardware or software was not totally lost and could be used without much modification on the newer and different systems. Standardisation in engineering hardware is well known. Further, it is possible to obtain hardware and software from a number of vendors and then be integrated into a single system. This means that there should be compatibility between various software elements as also between the hardware and software. This is achieved by maintaining proper interface standards at various levels. The following are some of them.

• GKS	(Graphical Kernel System)
• PHIGS	(Programmer's Hierarchical Interface for Graphics)

CORE	(ACM-SIGGRAPH)
GKS-3D	
IGES	(Initial Graphics Exchange Specification)
DXF	(Drawing Exchange Format)
STEP	(Standard for the Exchange of Product Model Data)
DMIS	(Dimensional Measurement Interface Specification)
VDI	(Virtual Device Interface)
VDM	(Virtual Device Metafile)
GKSM	(GKS Metafile)
NAPLPS	(North American Presentation Level Protocol Syntax)

Schematically, the operation of these standards with application programs is depicted in Fig. 5.1.

Details of some of these standards are discussed in the following sections. To study the other standards in greater details, references mentioned in the bibliography can be used.

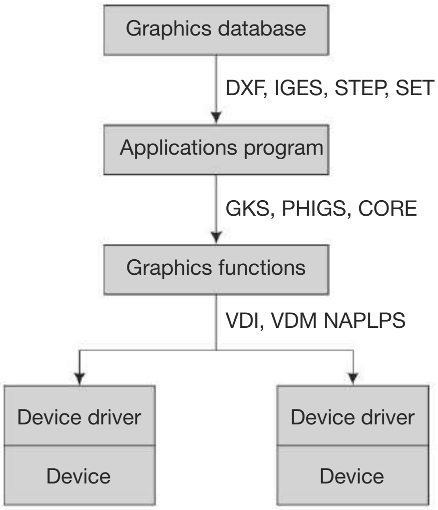


Fig. 5.1 *Various standards in graphics programming*

5.2 GRAPHICAL KERNEL SYSTEM (GKS)

People with experience in graphical programming for various hardware systems know how difficult it is if a program developed for a particular system were to run on a different system. The program would have to be completely recoded. If a comparison is made of a number of graphical programs, one may find that a substantial portion of it is similar. As a result, the same code is rewritten a number of times by different people. Some of the tasks such as drawing a line from one location to another, drawing a circle, setting a point, drawing a marker, form parts of such common features. Therefore, it is desirable to have programs interchangeable with a number of systems and also to make programmers learn the system once and then repeatedly use it on different systems. With this in mind, a number of attempts have been made in the past to provide a set of useful procedures (routines) for graphical manipulation, but with specific objective or hardware in mind. Examples are

- Tektronics Plot 10
- GINO-F from CAD Centre, Cambridge

However, the real effort as regards standardisation came from ACM-SIGGRAPH in the form of CORE standard in 1977. In 1979, DIN released the version of GKS. Taking all the existing graphic packages, ISO has standardised the GKS as a 2D standard in 1982. The main objectives that were put forward for GKS are the following.

1. To provide the complete range of graphical facilities in 2D, including the interactive capabilities.
2. To control all types of graphic devices such as plotters and display devices in a consistent manner.
3. To be small enough for a variety of programs.

The major contribution of GKS for the graphics programming is in terms of the layer model, as shown in Fig. 5.2. An environment for user to work is termed workstation in GKS. This could be a VDU, plotter or

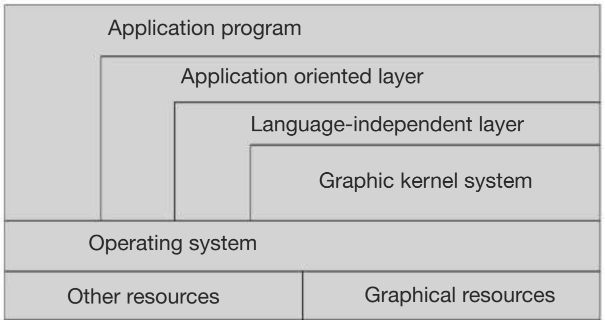


Fig. 5.2 *Layer model of graphics kernel system*

printer. For a programmer, all workstations are identical. The characteristics of these workstations are built into GKS. It is also possible to work simultaneously on more than one workstation.

The coordinate frames available to the user are of the following three types.

1. World Coordinates (WC) which are the user-oriented drawing coordinates.
2. Normalised Device Coordinates (NDC) which is a uniform system for all workstations.
3. Device Coordinates (DC) are the actual coordinate system for the particular workstation.

Input methods into GKS environment are organised in the following way:

LOCATOR	a means of entering the location in world coordinates
VALUATOR	real value in terms of distances
CHOICE	integer options such as 0, 1, 2, 3, etc.
PICK	to select an object or segment in a drawing already created
STRING	character values
STROKE	to provide continuously the location values in world coordinates

For drawing lines, the concept of PEN is used. PEN has the attributes of colour, thickness and line type. Lines can be drawn with any PEN that can be defined. The basic graphic primitives that were made available are

- POLYLINE for lines after specifying the line type, line width and line colour
- POLYMARKER for specific marker types after specifying the type, size and colour
- GENERALISED DRAWING PRIMITIVES (GDP) for specific graphic primitives such as arc, circle, ellipse, spline, etc.
- TEXT after specifying font type, precision, colour, height of the box, expansion factor, spacing, up vector and the path (left, right, up or down)
- FILLAREA for hatching and filling of areas

In essence, the GKS is essentially a set of procedures that can be called by user programs for carrying out certain generalised functions. In the interest of interchangeability, ISO has identified certain calling conventions for all these functions in various languages in order to take care of the variability of the programming languages. A typical binding for FORTRAN as implemented by IBM is given in Table 5.1.

GKS is defined in terms of a number of levels describing the level of support in terms of facilities. The highest level is 2c, though level 2b is the most commonly available facility with marginal difference in terms of the length of input queue (5 in case of 2c and 0 in case of 2b). A number of implementations are available for GKS on all types of computers starting from the micros to the main frame computers.

Table 5.1 *GGDP*

Purpose:	Output routine. Outputs one of the four generalised drawing primitives.	
Format:	CALL GGDP (n, px, py, primid, ldr, datrec)	
n	Integer*2	Number of points
px(n)	Real	List of *X* coordinates (WC)
py(n)	Real	List of *Y* coordinates (WC)
primid	Integer*2 1 = bar 2 = arc 3 = pie slice 4 = circle	GDP identifier (Fig. 5.3)
ldr	Integer*2	Length of data record
datrec	Character*80	Data record

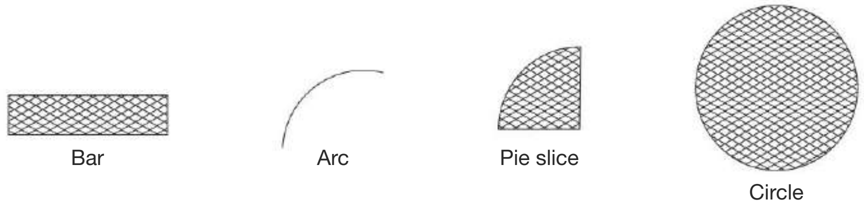


Fig. 5.3 *Graphics primitives in IBM GKS*

5.3 OTHER GRAPHIC STANDARDS

Besides GKS, which is well accepted, a number of separate standards have been developed to address the aspects which were not covered by GKS. Some of these standards are briefly described below.

GKS 3D The GKS has been subsequently enhanced to provide a separate standard for the three dimensions as GKS 3D, which maintains compatibility with the 2D standard.

PHIGS The other 3D graphic standard is PHIGS (Programmer's Hierarchical Interface for Graphics), being accepted by the CAD vendors as the system capable of taking care of the 3D graphical work as well as animation. Some of the features that are specific to PHIGS and are not well supported by GKS are the following:

- very high interactivity
- hierarchical structuring of data
- real time modification of graphic data
- support for geometric animation
- adaptability to distributed user environment

NAPLPS The North American Presentation Level Protocol Syntax (NAPLPS) is the presentation standard de-veloped jointly by the Canadian government and AT & T and other computer communication companies as a basis for transferring data from computers to the video display systems such as teletext and other video presentation systems.

The NAPLPS is a means of encoding the graphic data consisting of both graphics and text into an electronically transferrable format (ASCII). Some of the major features of NAPLPS are the following.

- The NAPLPS code is compact and is roughly about 10 per cent in comparison with the other formats.
- The graphic format used is resolution-independent. As a result, if the output produced on a lower resolution system is displayed on a higher resolution system, the output would be more clear.
- The NAPLPS is capable of being integrated into all communication networks such as television broadcasting signals, video tapes, etc.
- The colour look-up tables form part of NAPLPS transmission which enable a large range of colours to be produced instantaneously on the host system.

With the developments in the VLSI technology, these graphic standards already started appearing in hardware in the form of graphic adapter cards or special purpose chips (IC) to improve the graphic performance.

5.4 EXCHANGE OF MODELLING DATA

With proliferation of computers to do a number of tasks related to design and manufacturing, it becomes a necessity to have a means of communication between the various systems in the same plant or between different plants. Since the CAD/CAM software is available from a number of vendors, it becomes necessary that there should be a means by which different systems are able to interchange information to avoid the duplication of effort involved in the geometric model creation, which often happens to be the highest.

This means that the data format used by all the software should be the same. However, as explained earlier, the database formats are identified on the basis of the modelling requirements and is therefore not possible to have identical format for all the systems. However, it is possible to identify a certain format for drawing exchange and make it a standard so that the various systems can convert their internal format to this standard format or vice versa as shown in Fig. 5.4. This standard, established by the initiative of National Bureau of

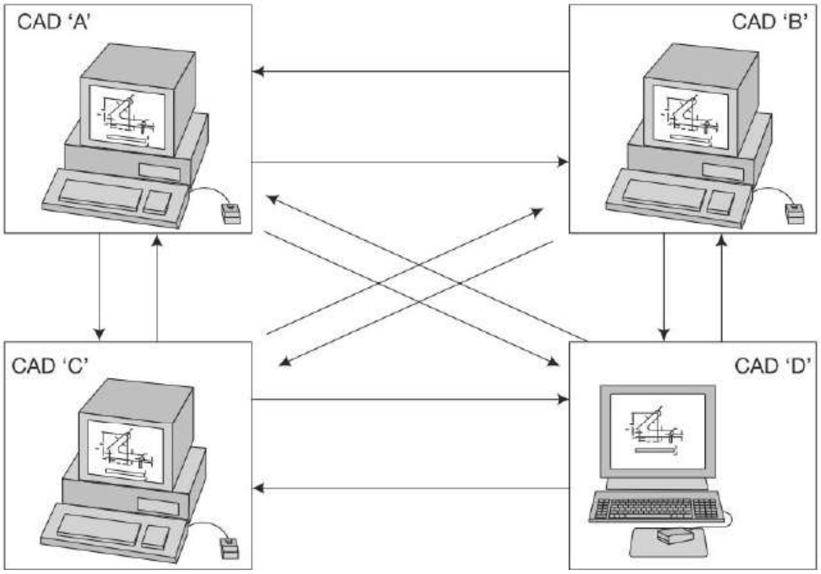


Fig. 5.4 *Data exchange between various systems*

Standards (NBS), USA is termed as Initial Graphics Exchange Specification (IGES). Some other formats which have been used because of the popularity of the corresponding systems are, Drawing Exchange Format (DXF) of AutoCAD software on IBM PC compatibles and the earlier CALCOMP vector format. The various developments in data exchange formats for CAD/CAM systems is shown in Fig. 5.5.

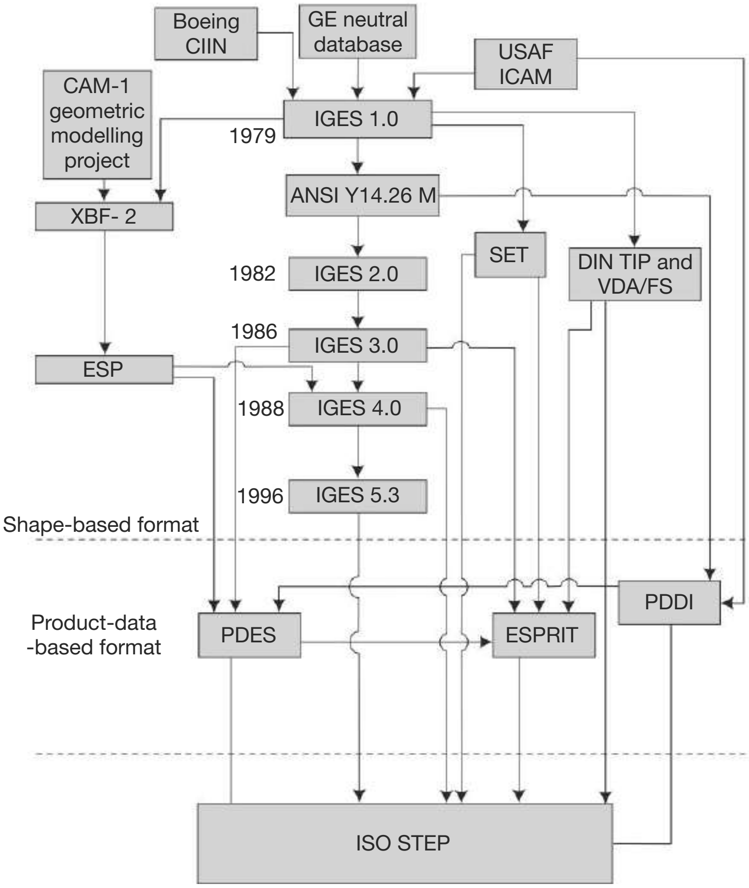


Fig. 5.5 *Developments in the drawing data exchange formats*

5.4.1 Initial Graphics Exchange Specification (IGES)

The IGES is the most comprehensive standard and is designed to transmit the entire product definition including that of manufacturing and any other associated information (Fig. 5.6). A brief description of the IGES version 3.0 is given below highlighting the philosophy of the conversion methodology.

In IGES, the records are present with 80 column fields, with columns 1 to 72 providing the data and columns 73 to 80 providing a sequence number for the record with identification as to the location of the sub-section. This sequence number is utilised as a pointer for the data. The IGES file consists of the following 6 sub-sections.

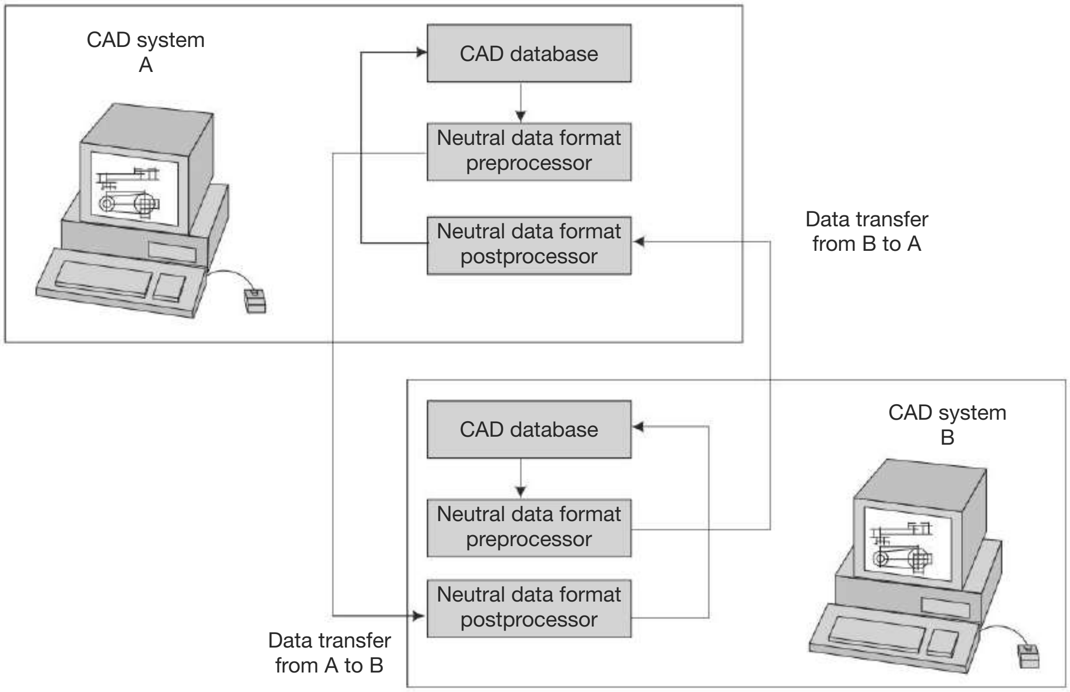


Fig. 5.6 *Data interchange method between two different CAD systems using neutral data format such as IGES or STEP*

Flag Section This is optional and is used to indicate the form in which the data is specified. Originally, the initial versions contained the data in ASCII format with a very detailed structure. This has been criticised by a number of people in view of the very large file sizes. From version 3.0 onwards, the format has been standardised in the following three modes.

- ASCII mode — default option
- Binary form
- Compressed ASCII form

The other two options provided help in reducing the bulk of the drawing exchange file size.

The sequence number has a starting character signifying the sub-section. They are

S for Start section
G for Global section
D for Directory entry section
P for Parameter entry section
T for Terminate section

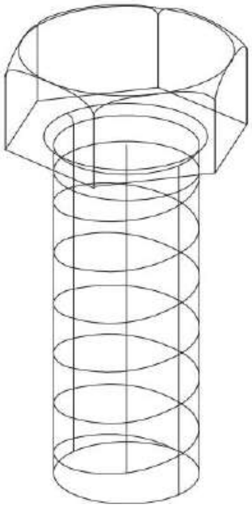


Fig. 5.7 *Component drawing for IGES file generation*

Start Section This section contains a man-readable prologue to the file. The information contained in this section is essentially for the person who would be postprocessing this for any other application. Any number of lines can be contained in this section. A sample listing of an IGES file for the drawing shown in Fig. 5.7 is shown in Fig. 5.8.

```
PTC IGES file: inpart.igs                                                            S           1
1H,,1H;,1H1,10Hinpart.igs,                                                           G           1
49HPro/ENGINEER by Parametric Technology Corporation,4H9741,32,38,7,38,              G           2
15,1H1,1.,1,4HINCH,32768,0.5,13H970430.104743,0.000396166,3.96182,                   G           3
6Hmjiang,7HUnknown,10,0,13H970430.104743;                                            G           4
      124        1        1        1        0        0        0          001000000D           1
      124        0        0        1        0                         XFORM         1D           2
      100        2        1        1        0        0        1          001010000D           3
      100        0        0        1        0                           ARC         ID           4
      110        5        1        1        0        0        0          001010000D           9
      110        0        0        1        0                          LINE         1D          10
      126       23        1        1        0        0        0          001010000D          35
      126        0        0       23        0                      B_SPLINE         ID          36
      128      333        1        1        0        0        0          001010000D         109
      128        0        0        5        0                        SPLSRF         1D         110
      102      338        1        1        0        0        0          001010000D         111
      102        0        0        1        0                        CCURVE         1D         112
      142      392        1        1        0        0        0          001010500D         119
      142        0        0        1        0                        UV_BND         1D         120
      144      393        1        1        0        0        0          000000000D         121
      144        0        0        1        0                       TRM_SRF         1D         122
124,1D0,0D0,0D0,0D0,0D0,-1D0,0D0,0D0,0D0,0D0,-1D0,-2.5D0;                           1P           1
110,5.055D-1,4.892211811710D-13,-6D-2,5.055D-1,0D0,-2.5D0;                          9P           5
124,-4.773959005888D-13,-1D0,9.771523867830D-14,5.655D-1,0D0,                      15P           9
-9.771523867830D-14,-1D0,5.525788160377D-14,1D0,                                   15P          10
-4.773959005888D-13,0D0,-6D-2;                                                     15P          11
126,16,3,0,0,1,0,0D0,0D0,0D0,0D0,7.142857142857D-2,                                35P          23
1.428571428571D-1,2.142857142857D-1,2.857142857143D-1,                  35P         24
3.571428571429D-1,4.285714285714D-1,5D-1,5.714285714286D-1,                        35P          25
6.428571428571D-1,7.142857142857D-1,7.857142857143D-1,                  35P         26
8.571428571429D-1,9.285714285714D-1,1D0,1D0,1D0,1D0,1D0,1D0,1D0,                   35P          27
1D0,1D0,1D0,1D0,1D0,1D0,1D0,1D0,1D0,1D0,1D0,1D0,1D0,1D0,                           35P          28
-8.515916470547D-1,0D0,5.314082992637D-1,-8.515916467998D-1,                       35P          29
4.415366701333D-10,5.3706512091D,-1,-8.507889031425D-1,                            35P          30
1.390393241360D-3,5.484405776395D-1,-8.465647755634D-1,                            35P          31
8.706796825981D-3,5.664743901286D-1,-8.394904277888D-1,                            35P          32
2.095992660211D-2,5.831252168121D-1,-8.298362442327D-1,                            35P          33
3.768146302686D-2,5.980689991631D-1,-8.180844276660D-1,                            35P          34
5.803620640165D-2,6.106867519845D-1,-8.051896631971D-1,                            35P          35
8.037059361337D-2,6.206445833208D-1,-7.908357348530D-1,                            35P          36
1.052323267936D-1,6.286780792337D-1,-7.748679794470D-1,                            35P          37
1.328892904395D-1,6.349611836816D-1,-7.570565254117D-1,                            35P          38
1.637396337855D-1,6.396365264201D-1,-7.371925938084D-1,                            35P          39
1.981449725603D-1,6.427982783358D-1,-7.152461016837D-1,                            35P          40
2.361574119683D-1,6.446068238875D-1,-6.910310080480D-1,                            35P          41
2.780991844553D-1,6.453689461482D-1,-6.650729444190D-1,                            35P          42
3.230598695269D-1,6.455109192813D-1,-6.476063658371D-1,                            35P          43
3.533128710651D-1,6.455000000001D-1,-6.387215448846D-1,                            35P          44
3.687018323709D-1,6.455D-1,0D0,1D0,ODO,ODO,1DO;                                    35P          45
120,147,151,-5.078087829273D-2,3.216475481452D0;                                  153P         415
110,1.602212258399D0,3.153643628373D0,ODO,3.141593160559D-2,                      163P         422
3.153643628380D0,0D0;                                                             163P         423
142,0,279,289,281,1;                                                              291P         703
144,279,1,0,291;                                                                  293P         704
142,0,449,465,453,1;                                                              467P        1126
102,13,557,559,561,563,565,567,569,571,573,575,577,579,581;                       583P        1545
110,0D0,0D0,6.455D-1,0D0,0D0,5.314083529453D-1;                                   585P        1546
S      1G      4D      586P     1546                                                 T           1
```

Fig. 5.8 *Partial listing of the IGES file for the drawing shown in Fig. 5.7*

Global Section This contains information about details of the product, the person originating the product, name of the company originating it, date, the details of the system which generated it, drafting standard used and some information required for its postprocessing on the host computer.

Directory Entry Section For each entity present in the drawing is fixed in size and contains 20 fields of 8 characters each. The purpose of this section is to provide an index for the file and to contain attribute information. Some of the attribute information such as colour, line type, transformation matrix, etc., may be present directly or through a pointer (to a record in the same file) where the necessary information is stored. It also contains the pointer to the parameter data section entry which actually contains the requisite parameter data.

Parameter Data Section This contains the data associated with the entities. A free format is allowed for maximum convenience. It may contain any number of records. The total number of entities that are present in IGES version 5.1 are as given.

(a) Geometric Entities

101	Circular arc
102	Composite curve
104	Conic arc
106	Copious data — centre line — linear path — section line — simple closed curve — witness line
108	Plane
110	Line
112	Parametric spline curve
114	Parametric spline surface
116	Point
118	Ruled surface
120	Surface of revolution
122	Tabulated cylinder
124	Transformation matrix
125	Flash
126	Rational B-spline curve
128	Rational B-spline surface
130	Offset curve
132	Connect point
134	Node
136	Finite element
138	Nodal displacement and rotation
140	Offset surface
141	Boundary
142	Curve on a parametric surface

143	Bounded surface
144	Trimmed parametric surface
146	Nodal results
148	Element results
150	Block
152	Right angular wedge
154	Right circular cylinder
156	Right circular cone frustum
158	Sphere
160	Torus
162	Solid of Revolution
164	Solid of linear extrusion
168	Ellipsoid
180	Boolean Tree
182	Selected Component
184	Solid assembly
186	Manifold solid B-rep object
190	Plane surface
192	Right circular cylindrical surface
194	Right circular conical surface
196	Spherical surface
198	Toroidal surface

(b) Annotation Entities

106	Copious data — centre line — section — witness line
202	Angular dimension
204	Curve dimension
206	Diameter dimension
208	Flag note
210	General label
212	General note
214	Leader (Arrow)
216	Linear dimension
218	Ordinate dimension
220	Point dimension
222	Radius dimension
228	General symbol
230	Sectioned area

(c) Structure Entities

302	Associativity definition
304	Line font definition
306	Macro definition
308	Sub-figure definition
310	Text font definition
312	Text display template
314	Colour definition
316	Units data
320	Network sub-figure definition
322	Attribute table definition
402	Associativity instance
404	Drawing entity
406	Property entity
408	Singular sub figure instance entity
410	View entity
412	Rectangular array sub figure instance entity
414	Circular array sub figure instance entity
416	External reference
418	Nodal load/constraint
420	Network sub figure instance
422	Attribute table instance
502	Vertex
504	Edge
508	Loop
510	Face
514	Shell
600–699	MACRO instance entity
or 10000–99999	as specified by the user

The data present in the parameter section varies with the type of entity. Some typical examples of data recorded are shown below.

For a *circular arc* (type 100), the parameter data stored are

(i) Parallel displacement of the X,Y plane containing the arc along the Z axis
(ii) Arc centre coordinate, X
(iii) Arc centre coordinate, Y
(iv) Start point of the arc, X
(v) Start point of the arc, Y
(vi) End point of the arc, X

(vii) End point of the arc, Y

(viii) Back pointers as required for the properties

Terminate Section This contains the sub-totals of the records present in each of the earlier sections. This always contains a single record.

Though all the geometric entities such as line, arc, circle are defined in a number of ways depending upon the construction facilities provided by the geometric modeler, the IGES provides for a single type of geometric transformation. Thus, it is possible that some design technique used would be lost in the process of IGES conversion which is inevitable, so is the accuracy of internal representation.

It is possible through IGES to transmit the property data with each entity in the form of associated property. This would be useful for bill of materials. Similarly, the wire frame, surface modelling and solid modelling information can all be conveniently handled in the current Version 5.3 released in 1996. However, problems have been faced with IGES for faithful translation of geometry between different systems because of the following reasons:

(a) Export Choices There are many choices for exporting CAD geometric data through IGES. These export choices can make the resulting IGES file better or worse for its intended reader, depending on compatibility issues that are often poorly understood. For example, a Catia user can export analytic surfaces such as cones and planes or change them into spline surfaces before exporting. Some CAM systems would prefer the first format, others the second.

(b) Tolerances, Accuracy and Resolution All surface calculations are performed within the specific CAD system's accuracy or resolution. Most surface calculations are not analytic. Surface intersection calculations are repeated, refining the result each time, until it is within the specified tolerance. Tighter tolerances require significantly more calculation time. A CAD system treats two points closer than its tolerance as one point. The IGES problem this creates is when IGES files are moved between two CAD/CAM products using different accuracies. Moving a coarse-toleranced IGES file to a fine toleranced system produces curves that do not close and surfaces that have gaps and overlaps. Moving a fine-toleranced IGES to a coarse-toleranced system loses details for the opposite reason.

Solid modelling is even more sensitive to IGES data problems than the surface modelling. The goal of a solid modeller is to produce a part definition as a mathematically precise, valid solid. This means no gaps, cracks or overlaps of surfaces. It means the surfaces themselves must not self-intersect or fold on themselves.

The solids that are included for CSG modelling (IGES 4.0) are the following.

- Block
- Right angular wedge
- Right angular cylinder
- Right circular cone frustum
- Sphere
- Torus
- Solid of revolution
- Solid of linear extrusion
- Ellipsoid
- Boolean tree
- Solid instance
- Solid assembly

5.4.2 Standard for the Exchange of Product Model Data (STEP)

Standard for the Exchange of Product Model Data (STEP), officially the ISO standard 10303, Product Data Representation and Exchange, is a series of international standards with the goal of defining data across the full engineering and manufacturing life cycle. The ability to share data across applications, across vendor platforms and between contractors, suppliers and customers, is the main goal of this standard.

The broad scope of STEP is as follows.

1. The standard method of representing the information necessary for completely defining a product throughout its entire life, i.e., from the product conception to the end of useful life.
2. Standard methods for exchanging the data electronically between two different systems.

The STEP documentation is split into eight major areas.

Overview It gives the general introduction and overview of the standard and forms part one of the ISO standard 10303.

Description Methods The application protocols planned in STEP are far reaching compared to any other existing standard or models. Hence, a new descriptive formal information modelling language called EXPRESS is developed such that the protocols be properly defined. These are given in parts 11 to 13.

Implementation Methods This provides specifications as to how the STEP information be physically represented for the exchange environment. This, therefore, refers to the actual implementation levels. Details can be found in parts 21 to 26.

Conformance and Tools This part provides the specifications for conformance testing of the processors used for STEP information. They provide information on methods for testing of software-product conformance to the STEP standard, guidance for creating abstract-test suites and the responsibilities of testing laboratories. These are given in parts 31 to 35. The STEP standard is unique in that it places a very high emphasis on testing and actually includes these methods in the standard.

Integrated-generic Resources These contain the specifications of the information models that support various application areas that form part of STEP. The topics that form part of this specification are: geometric and topological representation, product structure organisation, materials, visual presentation, tolerances, form features and process structure and properties. These are given in parts 41 to 46.

Application Information Models These specify the information models to be used for specific application areas such as draughting, finite element analysis, kinematics, building core model and Engineering analysis core. These are given in part numbers 101 upwards.

Application Protocols These are the main protocols to be used as subsets of STEP information model for exchange of data between specific application systems (such as between two finite element systems or between a CAD and Process Planning system). These are given in part numbers 201 upwards.

Application Interpreted Constructs These relate to the specific resources useful for defining the generic structures useful for applications. These are reusable groups of information resource entities that make it easier to express identical semantics in more than one application protocol. Examples include edge-based wireframe, draughting elements, constructive solid geometry, etc. These are given in part numbers 501 onwards. Examples include edge-based wireframe, shell-based wireframe, geometry-bounded 2D wireframe, draughting annotation, drawing structure and administration, draughting elements, geometry-bounded surface, non-manifold surface, manifold surface, geometry-bounded wireframe, etc.

Application Protocols These define the context for the use of product data for a specific industrial need. These are more complex data models used to describe specific product-data applications. These parts are known as application protocols and describe not only what data is to be used in describing a product, but how the data is to be used in the model. The application protocols use the integrated information resources in well defined combinations and configurations to represent a particular data model of some phase of product life.

Application protocols currently in use are the Explicit Draughting AP 201 and the Configuration Controlled Design AP 203. Some details of these are given in Figs 5.9 to 5.11. Other examples are associative draughting,

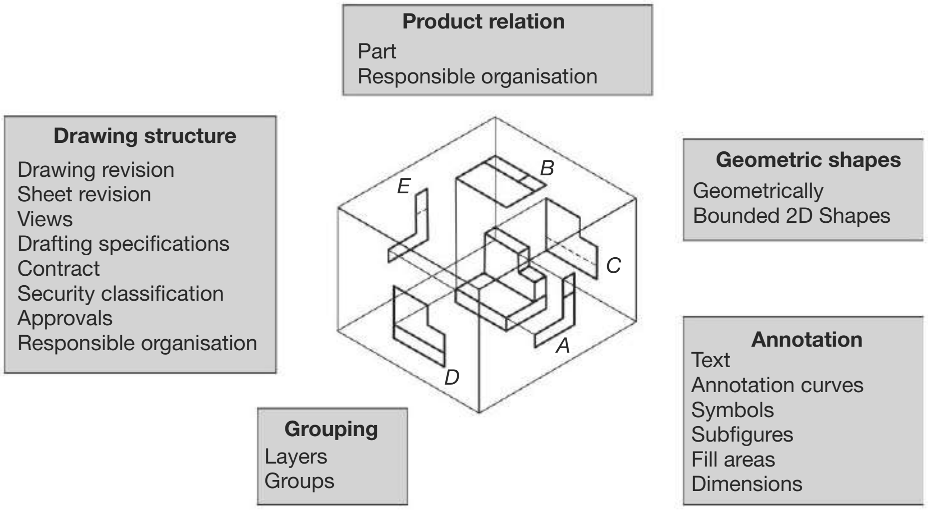


Fig. 5.9 *STEP application protocol AP 203 explicit draughting*

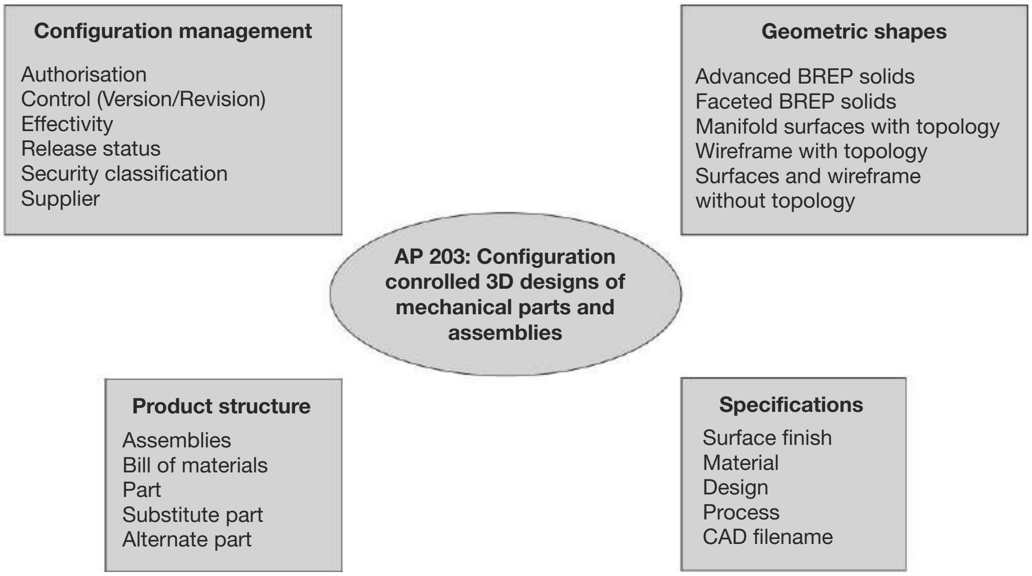


Fig. 5.10 *STEP application protocol AP 207 configuration controlled design*

mechanical design using boundary rep, mechanical design using surface rep, sheet metal die planning and design, electrotechnical design and installation, Numerical Control (NC) process plans for machined parts, core data for automotive mechanical design processes, Ship arrangements and mechanical parts definition for process planning using machining features.

As can be seen from the above, the philosophy of STEP goes beyond a traditional standard. It encompasses all the knowledge that has been gathered so far by the various partners in the total

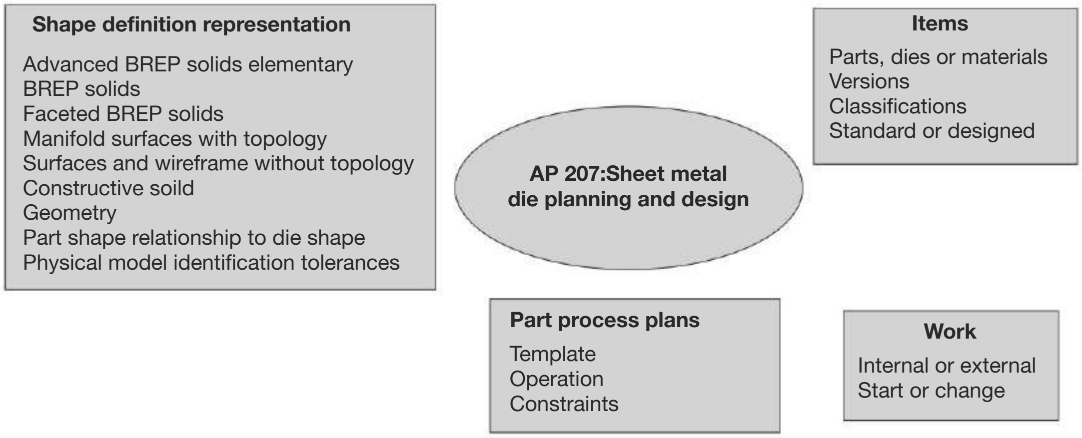


Fig. 5.11 *Example for STEP file generation sheet metal die planning and design*

CAD/CAM arena. The standard also makes enough provisions for future developments and knowledge gathering with the available provisions. This was made possible because of the cooperation of a large number of users and vendors that have direct interest in CAD/CAM systems. This is also the reason why it took so long (ISO initiated the work on STEP in 1985) for coming to this form. Even today (early 2001), the standard is still evolving and only some parts are released for use by the industry as ISO 10303. The other parts are in various stages of proposal to standard stages and would take some time before completion.

5.4.3 Drawing Exchange Format (DXF)

The DXF format has been developed and supported by Autodesk for use with the AutoCAD drawing files. It is not an industry standard developed by any standards organisation, but in view of the widespread use of AutoCAD made it a default standard for use of a variety of CAD/CAM vendors.

A Drawing Interchange File is simply an ASCII text file with a file extension of .DXF and specially formatted text. The overall organization of a DXF file is as follows.

HEADER Section This section contains general information about the drawing similar to the Global section of IGES. It consists of the AutoCAD database version number and a number of system variables. Each parameter contains a variable name and associated value. This information is used for database conversion purpose.

CLASSES Section It holds the information for application-defined classes, whose instances appear in the BLOCKS, ENTITIES and OBJECTS sections of the database. A class definition is permanently fixed in the class hierarchy.

TABLES Section This contains definitions for the following symbol tables which directly relates to the object types available in AutoCAD.

- Linetype table
- Layer table
- Text style table
- View table
- User coordinate system table
- Viewport configuration table
- Dimension style table
- Application identification table
- Block reference table

BLOCKS Section This contains block (symbol) definition and drawing entities that make up each block reference in the drawing.

ENTITIES Section This contains the graphical objects (entities) in the drawing, including block references (insert entities).

OBJECTS Section This contains the non-graphical objects in the drawing. All objects that are not entities or symbol table records or symbol tables are stored in this section. Examples of entries in the OBJECTS section are dictionaries that contain mline (multiple lines) styles and groups.

A DXF file is composed of many groups, each of which occupies two lines in the DXF file. The first line is a group code. The second line is the group value, in a format that depends on the type of group specified by the group code. DXF files are either standard ASCII text or special binary form files which are more compact.

5.4.4 DMIS

Dimensional Measurement Interface Specification (DMIS) is a new standard in communication being established by CAM-I for manufacturing. Most of the standards that are existing or discussed in this book pertain to the translation of data when the data is generated in design form. However, this standard tries to establish a means of knowing what has been made by the CAM process. The objective of DMIS is therefore to provide a bi-directional communication of inspection data between computer systems and inspection equipment so as to see what has to be made and has been made.

The database in the form of geometric instructions and manufacturing information is already present, which is being used by some of the part programming systems for automatically converting into CNC part programs. From the same database, it is also possible to generate the inspection programs for the Coordinate Measuring Machines (CMM).

The type of instructions needed for CMM are inspection probe selection, speed for positioning the probe, the path to be followed by the probe, speed and angle at which the probe approaches the workpiece, tolerance based information, etc. After a part has been produced on the CNC machine tool, the part would be checked on a CMM with the inspection program down loaded from the computer directly into the CMM for checking the part. After CMM checks the part, data about the part is sent back to the computer, where the original part geometry is stored as shown in Fig. 5.12. Thus, part geometry as designed is compared with the part produced and the resultant deviations existing could be identified, which would help in identifying problems in manufacturing and be suitably rectified.

DMIS provides a complete vocabulary for passing inspection program to the dimensional measuring equipment and to pass results back to the computer. A typical vocabulary details for the DMIS language is presented in Table 5.2 which is very similar in syntax to APT.

Table 5.2

TOL/CIRLTY	
Function:	Specifies a circularity tolerance for a feature and assigns to it a label.
Default:	None
Format:	T(label) = TOL/CIRLTY, tolzon, var_1, F(label)
Where:	var_1 can be: MMC or: LMC or: RFS Label, is an alphanumeric label assigned to the tolerance and is up to 10 characters in length.
CIRLTY	signifies circularity.

Table Contd.

Tolzon	is the delta radius (width) of the tolerance zone bounded by two concentric circles within which elements of the surface of F(label) must lie.
MMC	signifies that maximum material condition is applied to the feature for this tolerance.
LMC	signifies that least material condition is applied to the feature for this tolerance.
RFS	signifies regardless of feature size.
F(label)	is the feature to which the tolerance is to be applied.

Note: A circularity tolerance specifies a tolerance zone bounded by two concentric circles within which the feature must lie as shown in Fig. 5.13.

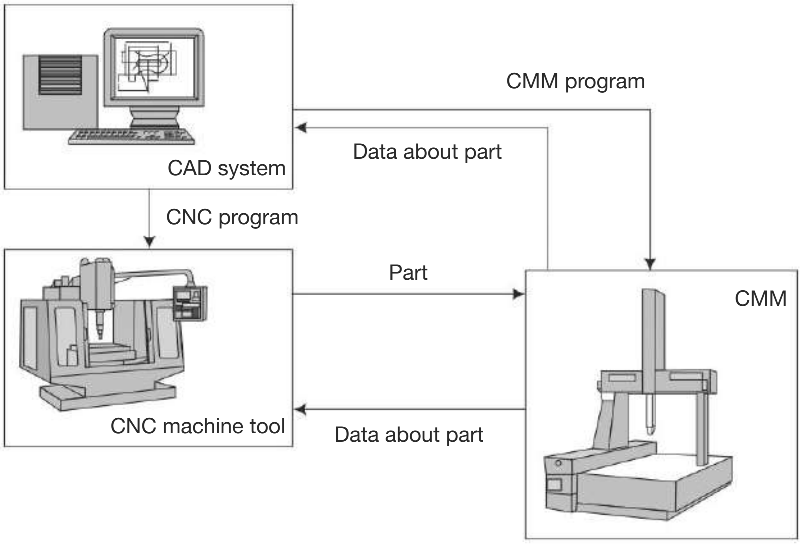


Fig. 5.12 *Introduction of measurement in product development cycle*

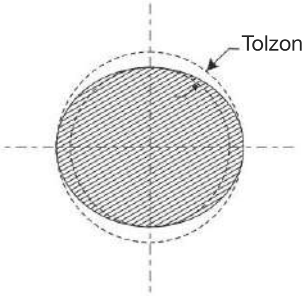


Fig. 5.13 *Meaning of tolerance zone*

Summary

- Standardisation of graphic systems is done at various levels. Standardisation can be at the graphic databases, graphic data handling systems as well as between the various manufacturing applications.
- Graphic kernel system is used to standardise the graphic system calling procedures at the lowest level so that programmers and programs can be easily migrated between different systems.
- The neutral CAD database is an important requirement to help with the transfer of information between various CAD/CAM systems.
- IGES (Initial Graphics Exchange Specification) is used for transferring information between various CAD systems for modelling as well as drafting data.
- STEP (Standard for the Exchange of Product Model Data) is being used extensively in view of its varied and better facilities for exchanging product model data.
- DXF developed by Autodesk is used for lower end drafting and model information exchange.

- DMIS (Dimensional Measurement Interface Specification) is used for transferring information by measuring in a coordinate measuring machine and transfer it to the CAD system for verification as well as reverse engineering.

Questions

1. What is the importance of standards in CAD/CAM?
2. Write briefly on any one of the known graphic standards.
3. What is meant by DMIS? Explain its importance in the manufacturing of mass consumption items.
4. How is IGES different from GKS?
5. Write a short note about DXF standard.
6. Explain the relevance of IGES in the Indian manufacturing scene.

6

INTRODUCTION TO A DRAFTING SYSTEM

Objectives

Drafting is one of the first computer applications used by many a user in view of its low cost and easier adoption. AutoCAD is by far the most widely used CAD software for drafting applications. A brief introduction of AutoCAD for drafting applications is given in this chapter. After completing the study of this chapter the reader should be able to

- Understand the basic structure of a drafting system as used in AutoCAD
- Learn some of the facilities available for the construction of geometric elements in AutoCAD
- Visualise the usefulness of the various editing commands to improve the productivity of the draftsman

6.1 BASIC FACILITIES IN AUTOCAD

The release 2010 screen is shown in Fig. 6.1 which has the familiar Windows look and feel, in terms of the various buttons and an easier interface. AutoCAD revamped the user interface to follow the familiar ribbon interface that has been now used with Microsoft Office applications. Though this is a complete change from the classic user interface, it is far more intuitive and easy to navigate. However, the regular AutoCAD user needs to unlearn the old user interface and learn the new user interface.

Like all the other Windows applications, it has a set of drop-down windows for various menu options. From among them, the relevant ones for the purpose of starting are

New	This allows for starting a new drawing.
Open	This allows for opening an old drawing for editing.
Save	This allows for saving the current drawing.

Save As This allows for saving the current drawing with a new name.

DWG — AutoCAD 2010 drawing file
DWG — AutoCAD 2007 drawing file
DWG — AutoCAD 2004 drawing file
DWG — AutoCAD R14/LT2 drawing file
DWF — AutoCAD drawing web format file
DXF — AutoCAD 2007 drawing interchange file
DXF — AutoCAD 2004 drawing interchange file
DXF — AutoCAD R12/LT2 drawing interchange file

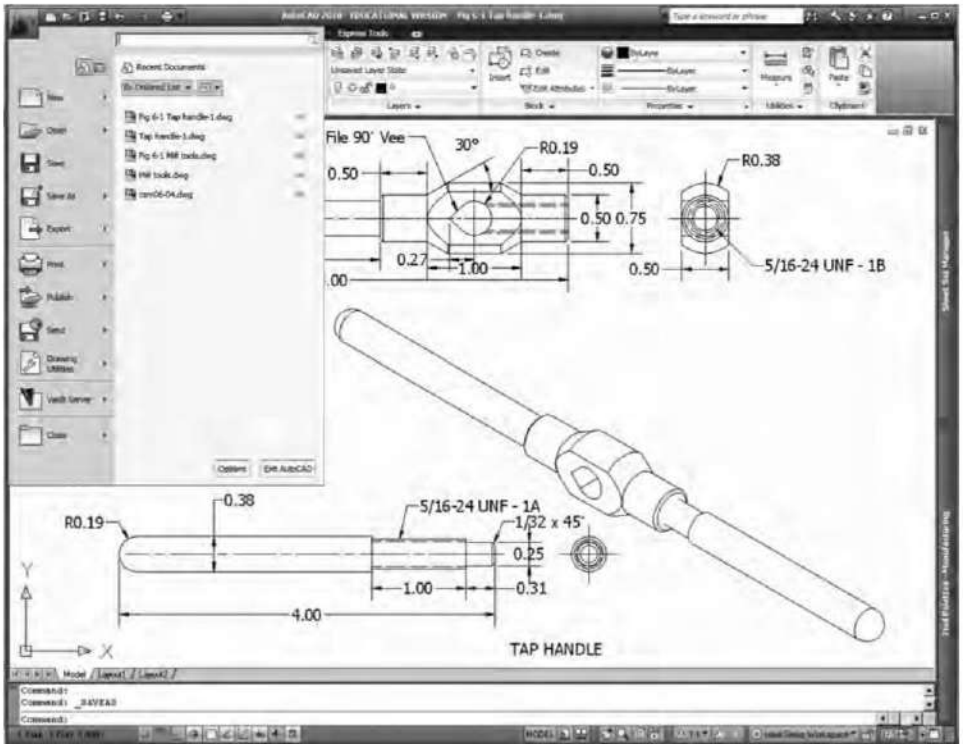
Export This allows for exporting the current drawing into other formats suitable for other programs such as 3D Studio. Some of the formats available are possible:

BMP — Device-independent bitmap file
EPS — Encapsulated PostScript file
SAT — ACIS solid object file
STL — Solid object stereo-lithography file
WMF — Windows Metafile

Print This allows for printing or plotting a drawing.

Publish This allows for opening an old drawing for editing.

Send This allows for sending the current drawing electronically.



Command:
Command: _SAVEAS
Command:

Fig. 6.1 *The AutoCAD 2010 screen appearance*

6.1.1 Screen Display

The total display screen in AutoCAD is divided into a number of areas as shown in Fig. 6.1. The status line is the bottom-most line. At the bottom, a command area is provided which is generally designated for 3 lines. In this portion, the interaction between the user and AutoCAD takes place.

The rest of the screen is designated as the ‘Drawing Area’. The actual drawing to be made is drawn in this drawing area. When the cursor is moved on various regions, its shape changes depending upon the screen area. For example, in the drawing area it is shown as a crosshair to facilitate the drawing function. When the cursor reaches the top line, a window corresponding to the menu item gets highlighted. The cursor cannot be moved by the mouse into the command area. When the user uses the keyboard, it automatically comes to the command area and it is generally shown as an underline ‘_’.

6.1.2 Menu

AutoCAD is a completely menu-driven system. There are many menu commands available. The menu items are made available through a large number of options such as

- Direct command entry
- Through the side-bar menu (in the classic AutoCAD option)
- Through the pop-up windows from the top menu (ribbon) bar, or
- Through the button bars located in any portion of the screen

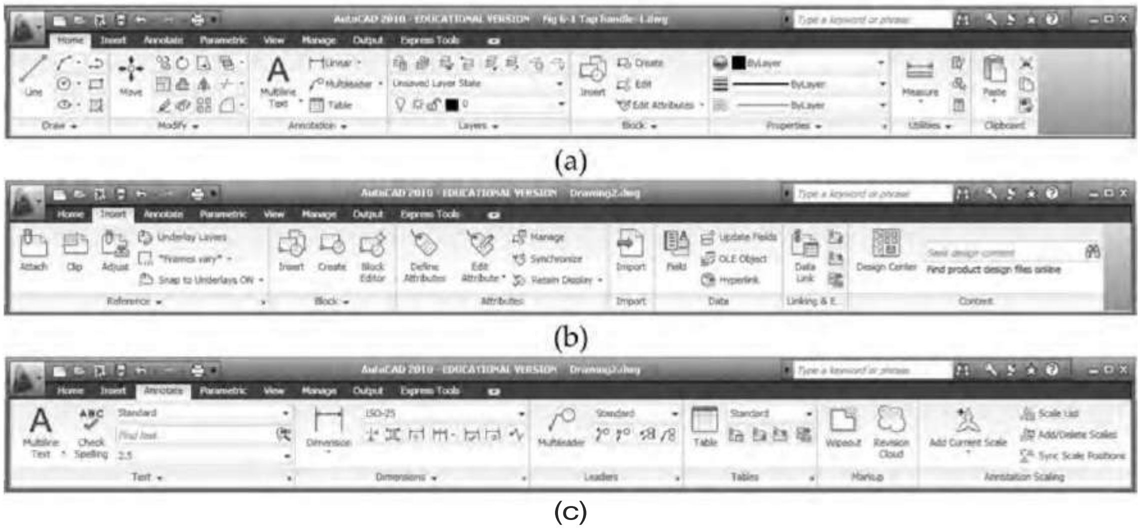


Fig. 6.2 *The ribbon bar in AutoCAD 2010 (a) Home option (b) Insert option (c) Annotate option*

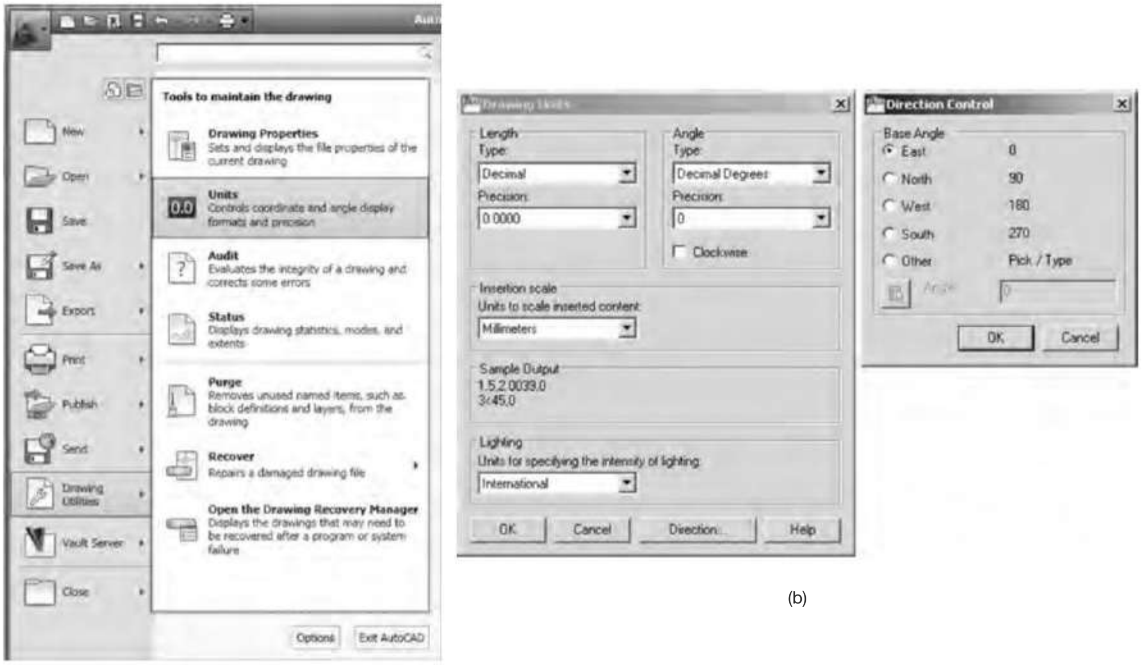
Compared to the earlier menu, the ribbon bar is more versatile and once used to will increase the productivity of the designer. As shown in Fig. 6.2, most of the options that are normally required for a bulk of the drafting processes are directly available from the home option as can be seen.

6.1.3 Planning for a Drawing

While planning a drawing in AutoCAD, one has to carefully organise some of the information. This is carried out in the set-up operations.

Units This lets us set up the units in which AutoCAD would have to work. Internally, AutoCAD works in default coordinates called *drawing units*. It is necessary to define the external representation of these units in recognisable form. This is achieved by the units command. Alternatively, access the 'Units' option from the main menu as shown in Fig. 6.3a. AutoCAD offers 5 different types of units to work with in the drawing. These are

1. Scientific
*2. Decimal
3. Engineering
4. Architectural
5. Fractional



Length
Type: Decimal
Precision: 0.0000

Angle
Type: Decimal Degrees
Precision: 0
☐ Clockwise

Insertion scale
Units to scale inserted content: Millimeters

Sample Output
1.5,2.0039,0
3<45,0

Lighting
Units for specifying the intensity of lighting: International

Base Angle
● East 0
○ North 90
○ West 180
○ South 270
○ Other Pick / Type
Angle 0

(a)

(b)

Fig. 6.3 *The specification of the drawing units in AutoCAD 2010*

Coordinate System The coordinate system used by all the CAD packages is generally the rectangular Cartesian coordinate system which follows the right-hand rule as shown in Fig. 6.4. AutoCAD also uses the rectangular coordinate system designated as *X*, *Y* and *Z* axes. The positive direction of these axes can

---

* signifies that it is the default unit chosen by the system.

be obtained by the application of the right-hand rule. Any point in the 3 dimensional space is therefore designated by the coordinate values of these 3 axes, e.g. (x, y, z).

The coordinates can be input into the system in a number of ways.

- By the direct input of coordinate values in their respective order, x, y and z. If the Z coordinate is not specified, it is considered as the current Z level given through the ELEVATION command.

Command: LINE<CR>

From point: 34.5,12.0<CR>

<CR> here represents the Carriage Return or the one key pressing of the key <ENTER>. Pressing the <Space bar> is also considered by AutoCAD as equivalent to the same.

The above example (Fig. 6.4) shows the coordinates being given in the absolute system.

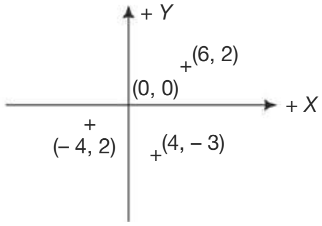


Fig. 6.4 *The coordinate system used in AutoCAD*

Limits It is generally necessary to specify the limits of the drawing that one is about to use. It is like choosing the right size of a drawing sheet for making the drawing. The actual size of the drawing has to be specified using the keyboard. This is achieved by using the LIMITS command as follows.

LIMITS establishes the size of the drawing and the associated drawing guides such as grids, rulers, etc., in the proper format. Though the LIMITS are specified in the beginning, they can be changed at any point of the drawing development, if the size exceeds the planned boundaries. The following are some guidelines to select the LIMITS in AutoCAD:

- Do not use square limits (e.g., 150 × 150). The screen is rectangular in shape and therefore to completely use the screen area for drawing, it is desirable to have rectangular limits such as (400 × 300).
- Start the limits with an idea of the final drawing size in mind.
- Make the lower left corner negative. This provides space for borders since all the drawing starts with (0, 0).
- Let the upper right-hand corner be more than required for additional space needed later.

Though the limits are specified as above, it is still possible to give the coordinate values beyond these limits by moving the cursor. However, if the LIMITS check option is kept on, then AutoCAD does not allow the user to specify any point beyond the LIMITS.

Grid Working on a plain drawing area is difficult since there is no means for the user to understand or correlate the relative positions or straightness of the various objects or entities made in the drawing. The GRID command controls the display of a grid of alignment dots to assist in the placement of objects in the drawing.

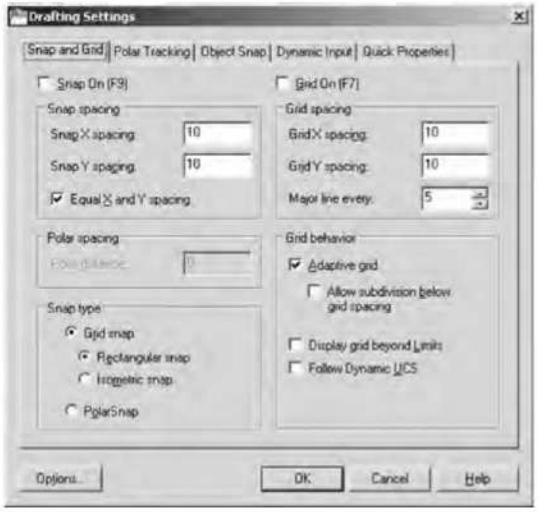
Snap The coordinate system in AutoCAD is a continuous one. This means that when the cursor is moved, it can go through an infinite number of points on the screen. This infinite resolution is not required or necessary for any drawing work. The resolution of the cursor movement can be effectively controlled using the SNAP command.

When the crosshair (cursor) is moved in the drawing area, it moves in increments of the SNAP spacing value specified. This is useful for inputting data through the digitiser or mouse. However, coordinates entered through the keyboard are not affected by SNAP setting in force.

The SNAP mode can be made operational by using the toggle control during the execution of any of the other commands in AutoCAD by using <F9> or ^B (using <CTRL> and <B> keys simultaneously). All these settings can be changed by getting the drafting settings (‘DSETTINGS’ at the command prompt) window as shown in Fig. 6.5.

Ortho The ORTHO command allows the user to control 'orthogonal' drawing mode. While drawing lines, the cursor moves to a point which makes it perpendicular to the point or in the same direction. As a result, all LINES and TRACES drawn while this mode is on are constrained to be horizontal or vertical. However, it will be possible to draw inclined lines by using the SNAP option with ROTATE as shown in Fig. 6.6.

The ORTHO mode can be made operational by using the toggle control during the execution of any of the other commands in AutoCAD by using <F8> or ^O (using <CTRL> and <O> keys simultaneously).



☐ Snap On (F9)

Snap spacing

Snap X spacing: 10

Snap Y spacing: 10

☑ Equal X and Y spacing

Polar spacing

Polar distance: 0

Snap type

● Grid snap

● Rectangular snap

○ Isometric snap

○ PolarSnap

☐ Grid On (F7)

Grid spacing

Grid X spacing: 10

Grid Y spacing: 10

Major line every: 5

Grid behavior

☑ Adaptive grid

☐ Allow subdivision below grid spacing

☐ Display grid beyond Limits

☐ Follow Dynamic UCS

Fig. 6.5 *Drafting settings used in AutoCAD*

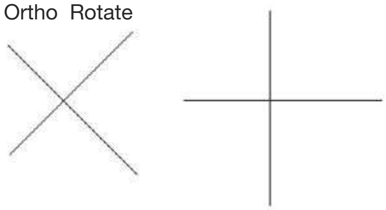


Fig. 6.6 *Drawing inclined orthogonal lines by combining the Ortho and Snap rotate options*

Help AutoCAD provides with complete help at any point of working in the program. Help can be obtained generally or specifically for any of the individual commands. Most of the information that is required by the user is generally provided by the HELP command (Fig. 6.7), which is always instantaneous.

6.1.4 Object Properties

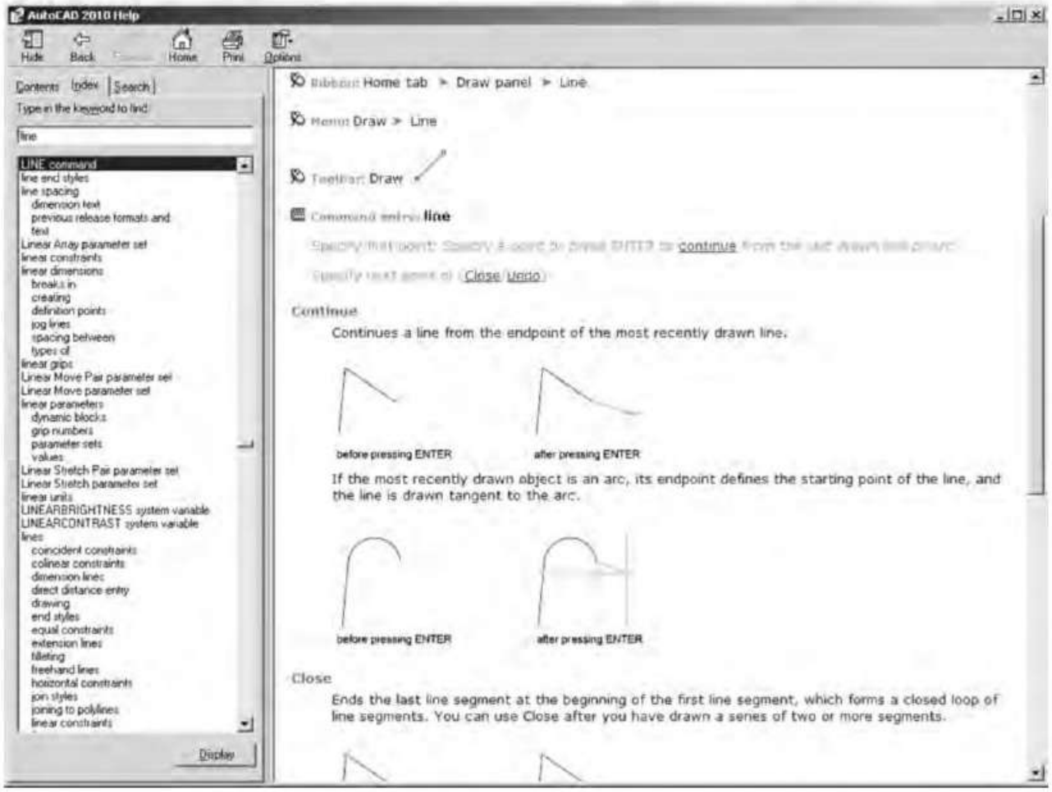
All objects in AutoCAD when created have certain properties such as colour, line type and the layer on which they are residing. The default settings are visible in the object properties button bar as shown in Fig. 6.2a. Descriptions of these properties are as follows.

Line Type AutoCAD allows the user to give various types of lines in a drawing. These dot-dash line types of each entity can be controlled either individually or by layers. To change the line type of existing objects, use the CHANGE command. To control layer-line types, use the LAYER command.

The LINETYPE command sets the line type (Fig. 6.8) for new entities. It can also load line-type definitions from a library file, write new definitions to a library file, and list the line-type definitions in a library file.

6.2 BASIC GEOMETRIC COMMANDS

What follows is a description of the basic commands in AutoCAD through which one can make simple drawings. The necessary introduction required for setting up the AutoCAD drawing was presented previously.



Continue

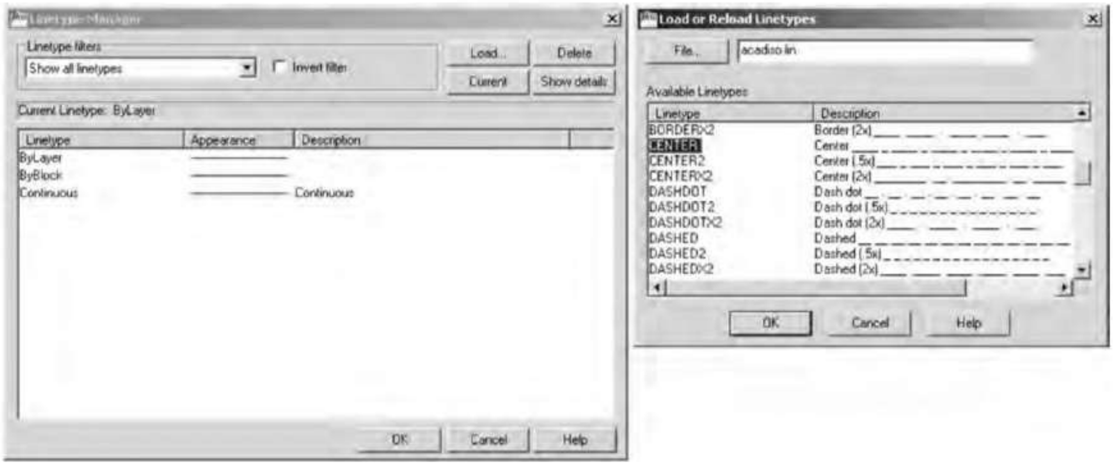
Continues a line from the endpoint of the most recently drawn line.

If the most recently drawn object is an arc, its endpoint defines the starting point of the line, and the line is drawn tangent to the arc.

Close

Ends the last line segment at the beginning of the first line segment, which forms a closed loop of line segments. You can use Close after you have drawn a series of two or more segments.

Fig. 6.7 *The help window showing the line command options*



Current Linetype: ByLayer

Linetype	Appearance	Description
ByLayer		
ByBlock		
Continuous		Continuous

File: acadiso.lin

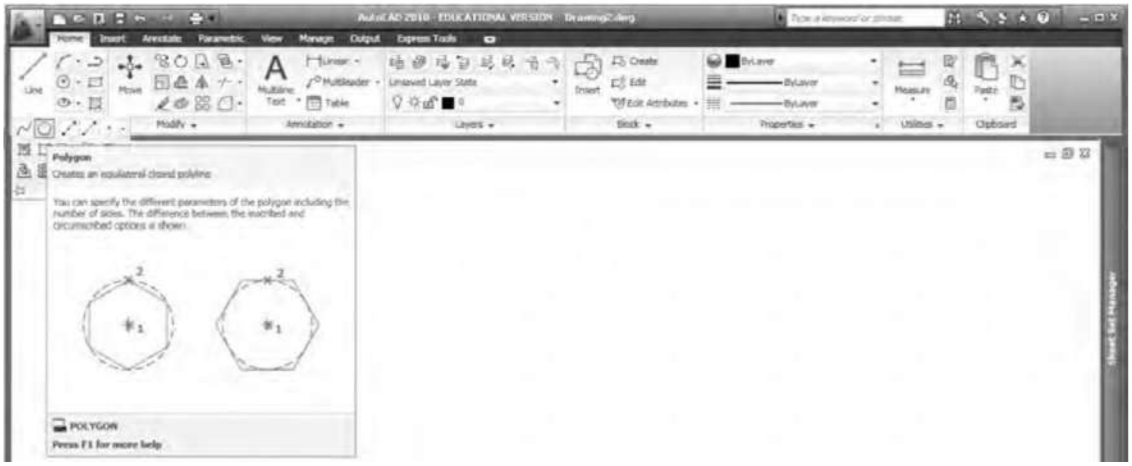
Linetype	Description
BORDERX2	Border (2x)
CENTER	Center
CENTER2	Center (.5x)
CENTERX2	Center (2x)
DASHDOT	Dash dot
DASHDOT2	Dash dot (.5x)
DASHDOTX2	Dash dot (2x)
DASHED	Dashed
DASHED2	Dashed (.5x)
DASHEDX2	Dashed (2x)

Fig. 6.8 *The pop-up window for changing the line type*

Some of the entities that can be used for making an AutoCAD drawing in 2D are as follows:

- Point
- Line
- Arc
- Circle
- Ellipse
- Polygon
- Poly line (pline)
- Doughnut (Donut)
- Sketch
- Text
- Block

In this chapter, we will see some of them and learn how to use them. The pop-up window for creating the geometric entities is shown in Fig. 6.9.



Polygon

Creates an equilateral closed polyline

You can specify the different parameters of the polygon including the number of sides. The difference between the inscribed and circumscribed options is shown.

POLYGON

Press F1 for more help

Fig. 6.9 *Pop-up window for the drawing commands with help*

Before describing the command structure and its variations, let us first mention some of the general conventions used in the AutoCAD dialogue system. Generally, AutoCAD provides a default option as < > in each of the command response. The value or option shown in the angle brackets is the most recently set value. To have the same value or option operation, one has to simply press the <Enter> key. The various options available for each of the commands are shown in full in the command window. However, the user needs to respond only with those letters which are shown in CAPITAL letters, which in many cases is only one letter which will automatically make the AutoCAD choose the right option.

Point This is the most basic command used in AutoCAD. It is used to specify a point or a node in the drawing for any given purpose. For example, it can be used for the centre of a circle or for the starting point of a straight line. It is also used as NODE in object snap options.

The point coordinates can be input into the system in a number of ways; for example, by the direct input of coordinate values in their respective order, x, y and z. If Z coordinate is not specified, it is considered as the current Z level given through the ELEVATION command.

The above example shows the coordinates being given in an absolute system. AutoCAD puts a marker at the specified location.

It is also possible to specify the coordinates in incremental format as the distances from the current cursor position in the drawing area. The distance is specified by using the '@' parameter before the actual value as shown below. When incremental values are used, both X, Y and Z are expected to be in incremental and hence '@' should be written only once as the example shown below:

Command: POINT<CR>

Coordinates of point: @34.5,12.0<CR>

It is also possible to specify the point coordinates using the polar coordinate format. This is an extension of the incremental format shown above. An example below illustrates its use.

Command: POINT<CR>

Command: Point: @15.5<45<CR>

The first value refers to the length of the line or polar radius, while the second quantity refers to the angle at which the line is drawn from the current point.

Line The LINE command allows the user to draw straight lines. The user can specify the desired end points using either 2D or 3D coordinates, or a combination. To erase the latest line segment without exiting the LINE command, enter 'U' (UNDO) when prompted for a 'To' point.

The user can continue the previous line by responding to the From point: prompt with a space or RETURN. If the user is drawing a sequence of lines that will become a closed polygon, the user can reply to the 'To point' prompt with 'C' to draw the last segment (close the polygon). Lines may be constrained to horizontal or vertical by the ORTHO command. The CLOSE option uses the start point of the first line segment in the current LINE command as the next point.

Circle The CIRCLE command is used to draw a full circle. The user can specify the circle in many ways. For specifying a circle, we need at least two values. There are at least 5 ways as follows:

- Centre point and radius
- Centre point and diameter
- Two points on the circumference across the diameter
- Three points on the circumference
- Tangents to two other already drawn entities and radius

The simplest method is by means of a centre point and the radius.

To specify the radius, the user can designate a point to be on the circumference or make use of the 'DRAG' facility in response to the 'Diameter/<Radius>'prompt to specify the circle size visually on screen. If it is more convenient to enter the diameter than the radius, reply to the 'Diameter/<Radius>'prompt with 'D'. This is generally done when the diameter is specified in the drawing.

The circle can also be specified using three points on the circumference (reply '3P' when prompted for the centre point), or by designating two end points of its diameter (reply '2P'). For these methods, the user can 'drag' the last point or specify object snap 'Tangent' points.

In addition, the user can draw a circle by specifying two entities (such as lines, circles, etc.) to which the circle should be tangent, and a radius. Enter 'TTR' for this option.

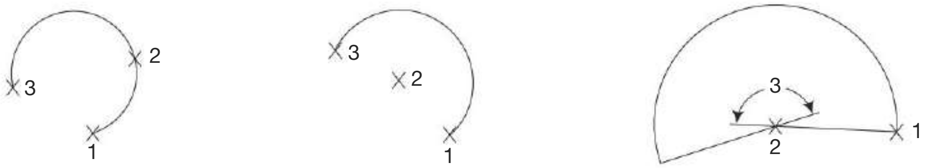
Arc The ARC command draws an arc (segment of a circle) as specified by any of the following methods.

- three points on the arc
- start point, centre, end point
- start point, centre, included angle
- start point, centre, length of chord

- start point, end point, radius
- start point, end point, included angle
- start point, end point, starting direction
- continuation of previous line or arc

To continue the previous line or arc, reply to first prompt with RETURN.

The ARC is always drawn in the counter-clockwise direction. Depending upon the data available, it is necessary to plan carefully the sequence in which the data is to be specified. Drawing a circle is easier than an arc, since the data to be given is more while care has to be taken to see that the data needs to be given in the proper sequence. It is also possible to use the OSNAP options when the various points are to be specified. The following examples in Fig. 6.10 of the ARC options are given for better understanding.



3 Point

Start – centre – end points

Start – centre – included angle

Fig. 6.10 *Arc example 1*

The three points can be specified in any order; however, the arc will be generated in the sequence in which the points are specified.

6.3 LAYERS

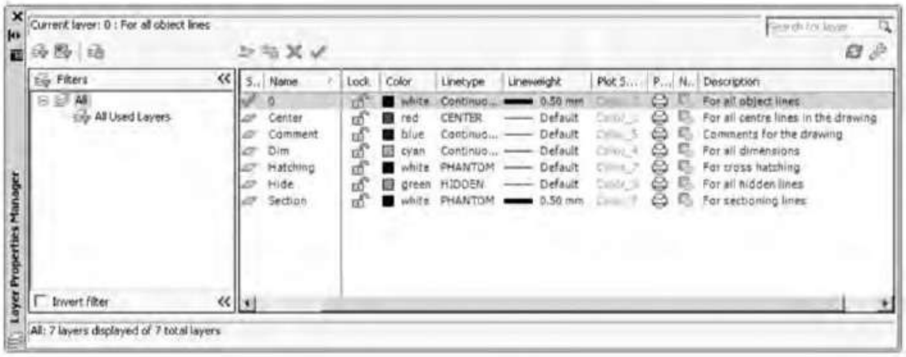
Drawings normally consist of lot of information which is of varying types such as geometric and alphanumeric. The geometric information may be further classified based on the purpose. Similarly, the alphanumeric information also can be classified into various categories. It becomes difficult to see all this information in one frame because of the cluttering effect it produces. Also, it is not necessary always to have all the information.

To deal with this, the layer concept is used in drawings. A layer is basically one which contains some information which can be geometric and alphanumeric. The reason of distributing all the information present in the drawing into various layers is that at any given time, some of the layers can be deleted from view (OFF) or can be made visible (ON). This helps in organising the information in a drawing.

Thus, each layer may be considered as a transparent sheet with information present. At any stage, the unwanted layers may be pulled out leaving only the requisite information visible. A typical pop-up window for altering the layers in a given drawing is shown in Fig. 6.11.

In AutoCAD, a layer has certain properties and certain conventions (rules) are to be followed while using them.

- Each layer has a name which can be up to 31 characters. The default layer name given by AutoCAD is 0. Some typical names could be note, remarks, etc. Alternatively, only numbers could be used for the layer name.
- A layer could be ON or OFF. When a layer is ON, the information present in it is visible on the screen (or in a plot). When OFF, though the information is not lost from the drawing, it is not visible in the drawing. More than one layer can be ON at any given time.



Current layer: 0 : For all object lines

S...	Name	Lock	Color	Linetype	Lineweight	Plot S...	P...	N...	Description
✓	0		■ white	Continuo...	0.50 mm				For all object lines
	Center		■ red	CENTER	Default				For all centre lines in the drawing
	Comment		■ blue	Continuo...	Default				Comments for the drawing
	Dim		■ cyan	Continuo...	Default				For all dimensions
	Hatching		■ white	PHANTOM	Default				For cross hatching
	Hide		■ green	HIDDEN	Default				For all hidden lines
	Section		■ white	PHANTOM	0.50 mm				For sectioning lines

All: 7 layers displayed of 7 total layers

Fig. 6.11 *The pop-up window for modifying the layers*

- A layer is either 'Current' or 'Inactive'. All the information being entered goes into the current layer. By virtue of this, only one layer can be Current at any given time. If the current layer is ON then the information being entered is visible on the screen. However, it is possible to have a layer current and off, though it is not a good practice.
- Each layer has a colour associated with it. All the information entered into that layer gets that colour. It does not mean that more than one colour can be present in a single layer. The colour of individual information can be altered by using the CHANGE command or by using the COLOR command.

Similarly, each layer has a default line type associated, which can be any one of the valid line types. It is also possible, as in the case of colour, to change the line type of individual elements in a layer.

6.4 DISPLAY CONTROL COMMANDS

ZOOM ZOOM is used to change the scale of display (Fig. 6.12). This can be used to magnify part of the drawing to any higher scale for looking closely at some fine details in the drawing. This is often quite useful during the construction stage since the size of the display screen is limited compared to the size of drawing and the details present there.

Fig. 6.12 *The pop-up window for the various display and zoom options*

There are a large number of options available within ZOOM (Table 6.1).

Table 6.1 *Options available with Zoom*

Scale<X>	A numeric zoom factor. A value less than 1 zooms out (reduction), and greater than 1 zooms in (magnification)
All	Zooms out to original drawing limits
Dynamic	Graphically selects any portion of the drawing as the next screen view, but also sees what part of the file is generated and therefore appears fast.
Center	Picks a centre point and a picture top and bottom by selecting two end points of a height. Alternatively, the height can be given as a value.
Extents	Shows everything in the file ignoring the limits. Similar to ALL except limits.
Left	Picks a lower left corner and a height of how much drawing information the use wants to display to fill up the screen.
Previous	Restores the last ZOOM setting.
Window	Picks two points that define a rectangular window to describe what part of the drawing file will be seen on the screen.

Choosing the DYNAMIC option displays all the drawing up to limits in a small window so that the entire drawing is visible in the display screen. The current visible window is shown as a rectangle linked to the cursor. Pressing the left button of the mouse and moving the mouse makes the window smaller or larger than the previous display. Once the window rectangle size is satisfactory, another pressing of the mouse left button fixes the size of the rectangle. The mouse can now be moved to any position to place the display window on the drawing. Then pressing the right mouse button correspondingly shows the image in the window in the full size of the screen. This is a very convenient option and requires normally a small number of steps to come up with the required image to be shown to the requisite scale.

PAN The PAN command allows the user to move the display window in any direction without changing the display magnification. This means the display being seen is through a window in an opaque sheet covering the drawing limits. The window can be moved to any location within the display limits, although no dynamic movement is possible. This shows details that are currently off the screen.

Object Snap The SNAP is useful for drawing a new object into the drawing by itself. However, it is often desired to have drawings made in relation to already existing objects. It may also be possible to have this relation be a very specific one.

In such situations, the **O**bject**SNAP** facility available in AutoCAD is quite useful. Sometimes, it may be required to start a line from an unknown precise tangent point on a circle. All that the user may know is a specific area where the tangent may be lying. Then by selecting the OSNAP option (Fig. 6.14), the system is able to automatically calculate the tangent point in the region selected. The various OSNAP options available are the following (Table 6.2):

Table 6.2 *Some of the options available with Zoom*

CENter	Centre of arc or circle
ENDpoint	Closest end point of line/arc or closest corner of trace/solid/3D face
INSertion	Insertion point of text/block/shape/attribute
INTersection	Intersection of lines/arcs/circles or corner of trace/solid/3D face

Contd..

Contd..

MIDpoint	Midpoint of line/arc or midpoint of an edge of trace/solid/3D face
NEArest	Nearest point on line/arc/circle/point
NODe	Nearest point entity (or dimension definition point)
NONe	None (off)
PERpendicular	Perpendicular to line/arc/circle
QUAdrant	Quadrant point of arc or circle
QUIck	Quick mode (first find, not closest)
TANgent	Tangent to arc or circle
Apparent Intersection	Apparent Intersection includes both Apparent Intersection and Extended Apparent Intersection. Apparent Intersection allows snapping to the apparent intersection of any two entities (e.g., arc, circle, ellipse, line, poly line, ray or spline) that do not physically intersect in 3D space, but appear to intersect on screen. Extended Apparent Intersection snaps to the imaginary intersection of two objects that would appear to intersect if the objects were extended along their natural paths.

When more than one option is selected, AutoCAD applies the selected snap modes to return a point closest to the centre of the aperture box. Use commas to separate multiple modes when entering from the command window. These modes can also be entered whenever a point is requested to over-ride the running object-snap modes.

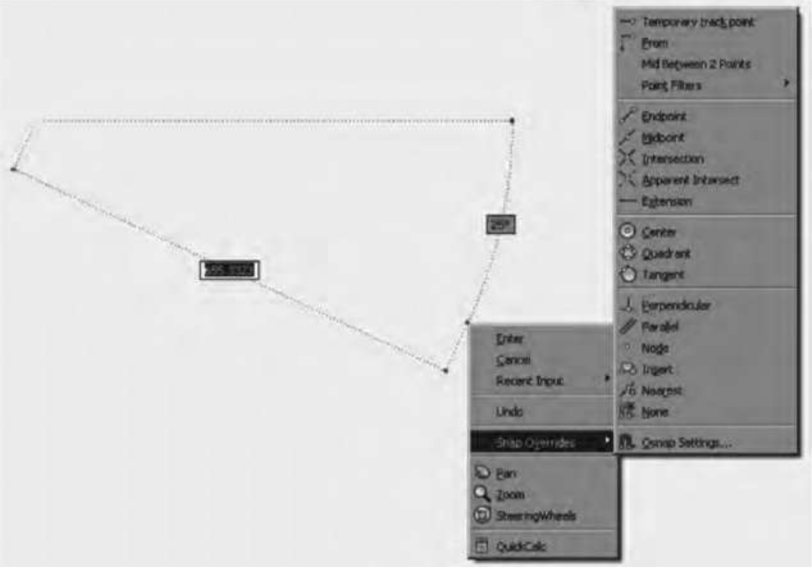


Fig. 6.13 *Object snap options by right clicking the mouse button*

Text Handling AutoCAD provides a large range of text-entering capabilities including various fonts, and other text-handling features (Fig. 6.14).

In each of the fonts (styles), it is possible to have lettering with the various attributes which are specified during the selection of the style as shown in the dialogue box (Fig. 6.15). The selection results are also shown

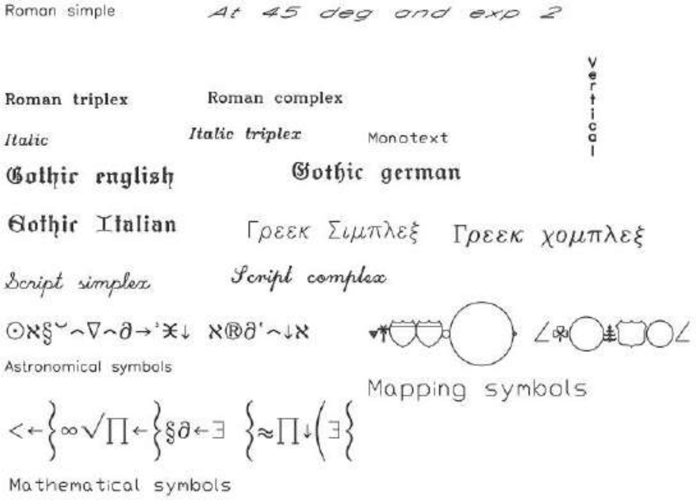
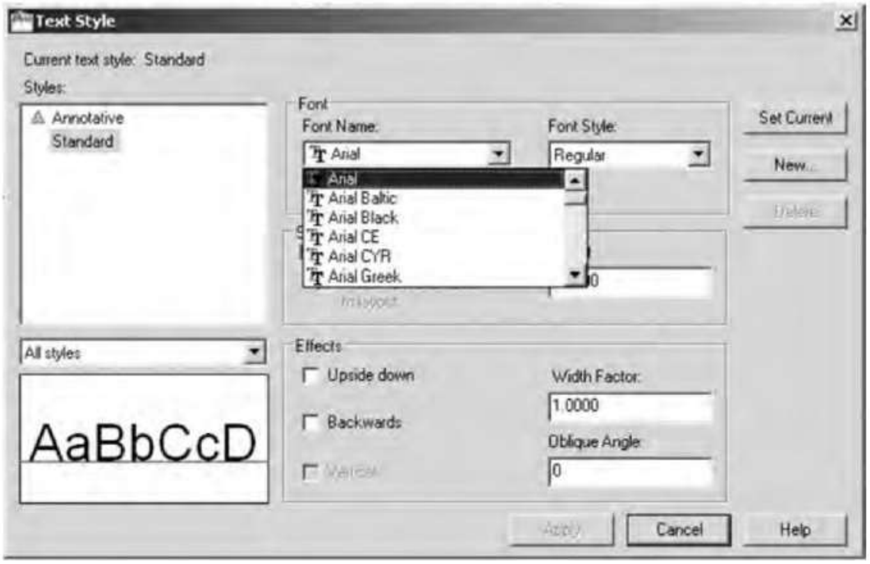


Fig. 6.14 *Sample fonts available in AutoCAD*

in the dialogue box. It is also possible to combine the attributes to get complex attributes. For example, the text, when originally stored, has a certain aspect ratio (height-to-width ratio) built in depending upon the text height chosen. It is possible to alter this by changing the width factor > 1 which makes the letters elongated and < 1 which makes them compressed. When the text is to be fitted in a given area, this is the technique used. Similarly, the text can be slanted by any angle.

The user can also combine any or all of these attributes to get combined effects. For example, the text can be expanded and written backwards. Some examples of combinations are shown in Fig. 6.15.



Current text style: Standard

Font Name: Arial

Arial
Arial Baltic
Arial Black
Arial CE
Arial CYR
Arial Greek

Font Style: Regular

☐ Upside down

☐ Backwards

☐ Vertical

Width Factor: 1.0000

Oblique Angle: 0

Fig. 6.15 *The font specifications that are given through the pop-up window*

6.5 EDITING A DRAWING

Editing capabilities are the most useful part of AutoCAD to exploit the productivity potential, making use of the already existing objects in the drawing. A few of the commonly used editing facilities are listed in Table 6.3.

Table 6.3 *Commonly used editing facilities*

Array	Places multiple copies of objects with a single command
Break	Cuts existing objects and/or erases portions of objects
Change	Changes spatial properties of some objects like location, text height, circle radius or line end-point location
Copy	Makes copies of objects
Erase	Allows to select objects in the drawing file and erase them
Mirror	Creates a mirror image of objects
Move	Picks up existing objects and puts them down in another location of the drawing file
Rotate	Turns existing objects to any angular specifications
Scale	Scales objects up or down to the user's specification

To use any of the editing functions, it is necessary to make a selection of the objects (entities) in the drawing on which the editing function is to be applied. For this purpose, many facilities (Fig. 6.16) are built into AutoCAD.

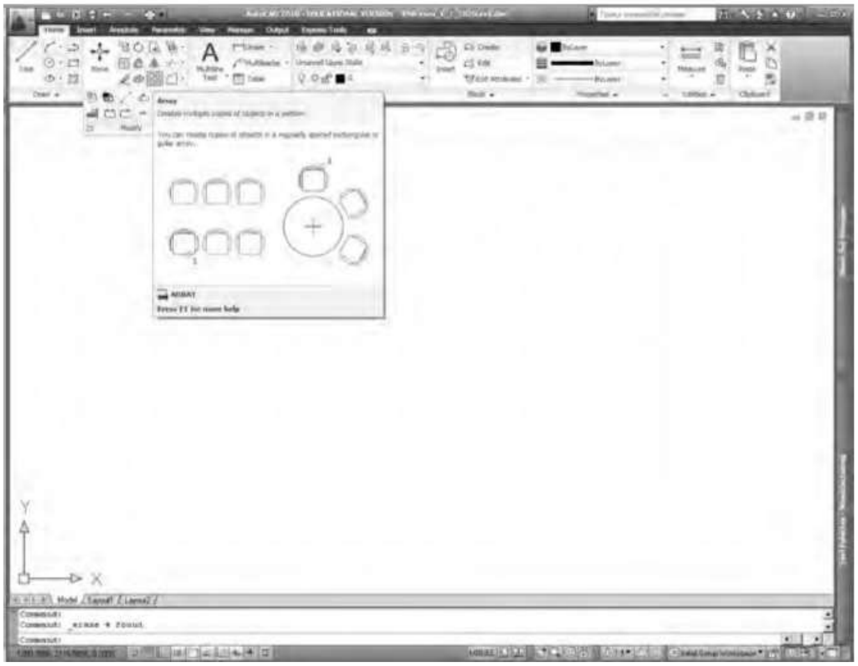


Fig. 6.16 *Some of the editing options available in AutoCAD*

One of the important things for editing is the selection of the objects or entities that were already drawn in the drawing. As any of the editing commands are issued, AutoCAD responds first with the object selection option. There are various options available for the same as shown in Table 6.4.

Table 6.4 *Options available for selecting objects*

Object	The default selects a set by picking individual objects.
Window	Selects a set by picking all objects that are completely inside a window drawn by the cursor control device.
Last	Uses only the last object created as a selection set.
Crossing	Works like a window, except it also includes any object which is partially within the window.
Remove	Deselecting from the selected object set. This is useful for removing any objects that are accidentally selected by any of the object-selection methods.
Add	This is the normal setting at the time of object selection. Switches from Remove back to normal. It is to be used for adding when the Remove option is specified.
Multiple	Allows multiple objects in close proximity, and/or speeds up selection by allowing multiple selections without highlighting or prompting.
Previous	Adds (or removes) the entire previous selection list to the current selection list.
Undo	Undoes or reverses the last selection operation. Each U undoes one selection operation.
Select All	Selects all the objects present in the drawing.
Window Polygon	This is similar to the window option, except that the rectangular window is replaced by a polygon of as many vertices as required with the edges not crossing each other.
Crossing Polygon	This is similar to the window polygon option, except that all objects crossing the polygon are also selected.
Select fence	This is similar to the crossing polygon option, except that the polygon here is not closed. Hence, the fence can cross itself, if necessary.
Select filters	Uses a special type as a filter (e.g., object properties) for selecting the objects. For example, all lines or objects with the same colour, etc. To use this option, it is necessary to create a filter list first by using the option FILTER.

6.5.1 Basic Editing Commands

MOVE The MOVE command is used to move one or more existing drawing entities (Fig. 6.17) from one location in the drawing to another.

The user can ‘drag’ the object into position on the screen. To do this, designate a reference point on the object in response to the ‘Base point...’ prompt, and then reply ‘DRAG’ to the ‘Second point:’ prompt. The selected objects will follow the movements of the screen crosshairs. Move the objects into position and then press the pointer’s ‘pick’ (left mouse) button.

Displacement can be directly specified in place of the set of points to designate the actual displacement.

COPY The COPY command is used to duplicate one or more existing drawing entities at another location (or locations) without erasing the original.

The user can ‘drag’ the object into position on the screen. To do this, designate a reference point on the object in response to the ‘Base point...’ prompt, and then reply ‘DRAG’ to the ‘Second point:’ prompt. The selected objects will follow the movements of the screen crosshairs. Move the objects into position and then press the pointer’s ‘pick’ (left mouse) button (Fig. 6.18).

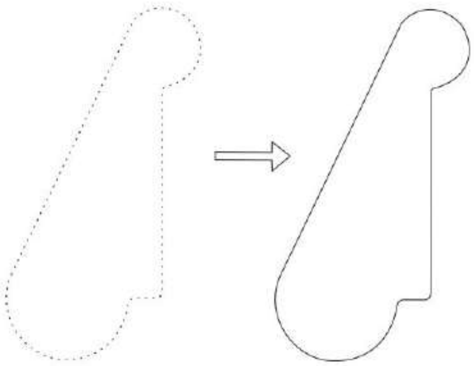


Fig. 6.17 *Moving objects in the drawing*

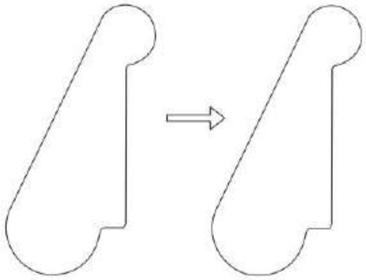


Fig. 6.18 *Copying an object in the drawing*

To make multiple copies, respond to the 'Base point' prompt with 'M'. The 'Base point' prompt then reappears, followed by repeated 'Second point' prompts. When the user made all the copies (Fig. 6.18) that are needed, gives a null response to the 'Second point' prompt to come out of the COPY command.

CHAMFER The CHAMFER command (Fig. 6.19) creates a bevel between two intersecting lines (or two adjacent line segments of a poly line) at a given distance from their intersection (Fig. 6.20). It can also trim the lines from the bevel edge and connect the trimmed ends with a new line if the TRIMMODE variable is set to 1. If it is set to 0, it leaves the selected edges intact. Different trim distances can be set for the two lines, and are retained with the drawing. If the specified lines do not intersect, CHAMFER will extend them until they do and then proceed as above. Chamfers can be applied to an entire poly line, chamfering all the intersections. A chamfer can only be applied between line segments and not on any other objects.

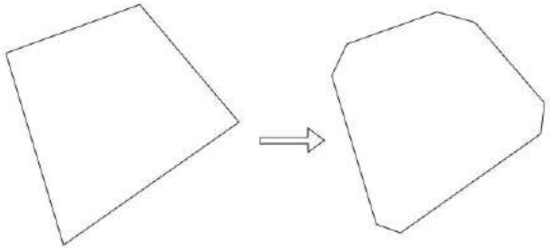


Fig. 6.19 *Example of chamfer as used in drawings*

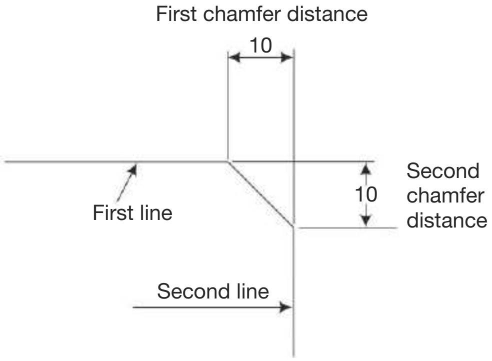


Fig. 6.20 *The meaning of chamfer distances in AutoCAD*

FILLET The FILLET command connects two lines, arcs, or circles with a smooth arc of specified radius. It adjusts the lengths of the original lines or arcs so they end exactly on the fillet arc (Fig. 6.21).

The fillet value specified remains in force until it is altered by another value. If the fillet radius is zero then the two lines will meet exactly at a point which is normally used to make a sharp corner.

Filleting can also be done to two circles, a line and a circle, a line and an arc, and a circle and an arc.

AutoCAD chooses the fillet whose end points are closest to the points by which the user selects the objects to be filleted. Lines and arcs are trimmed (depending upon the TRIMMODE setting) after filleting while circles are not trimmed. An entire poly line can also be filleted with a single command.

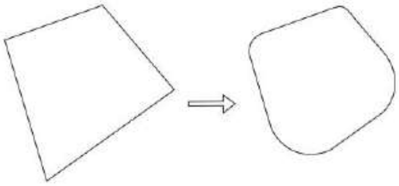


Fig. 6.21 *Example of fillet as used in drawings*

OFFSET The OFFSET command constructs an entity parallel to another entity at either a specified distance or through a specified point (Fig. 6.22). The user can OFFSET a line, arc, circle, or poly line. Offset lines are parallel, while the offset circles and arcs make concentric circles and arcs respectively.

To offset from a wide poly line, measure the offset distance from the centre-line of the poly line. Once the object is selected, it is highlighted on the screen. Depending on whether the user specified an offset distance or selected 'through point' in the original prompt, the user will receive one of the following prompts:

Side to offset:

Through point:

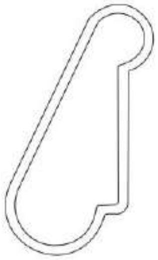


Fig. 6.22 *Example of offset as used in drawings*

The offset is then calculated and drawn. The selected object will be de-highlighted and the 'Select object to offset' prompt is re-issued. <RETURN> exits the command.

ARRAY The ARRAY command makes multiple copies of selected objects in a rectangular or circular (polar) pattern.

For a rectangular array, the user is asked for the number of columns and rows and the spacing between them. The array is built along a base line defined by the current snap rotation angle set by the 'SNAP Rotate' command.

AutoCAD constructs the array along the base line defined by the SNAP function. Normally, the SNAP is horizontal (Fig. 6.23). However, it is possible to alter the orientation by using the ROTATE option in SNAP. In such cases, the rectangular array is rotated by the SNAP rotate angle. However, the object itself is not rotated.

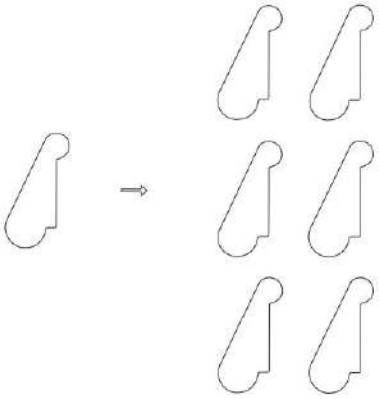


Fig. 6.23 *Rectangular array of objects*

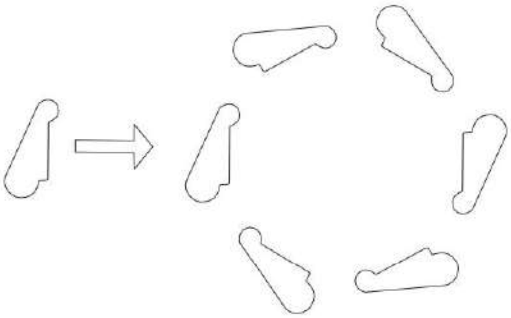


Fig. 6.24 *Polar array of objects rotated while copying*

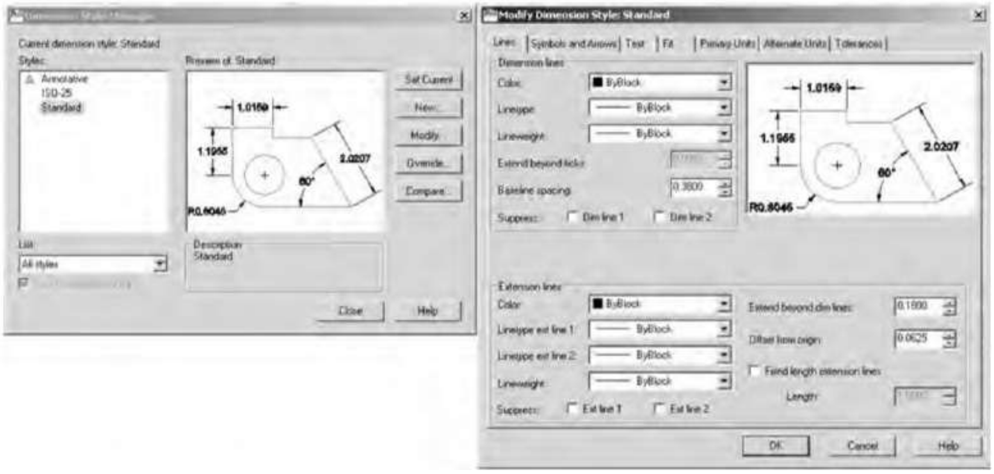
For a polar or circular array (Fig. 6.24), the user must first supply a centre point. Following this, the user must supply two of the following three parameters:

- the number of items in the array
- the number of degrees to fill
- the angle between items in the array

Optionally, the user can rotate the items as the array is drawn.

6.6 DIMENSIONING

After creating the various views of the model or after preparing the drawing, it is necessary to add dimensions at the appropriate places. AutoCAD provides semi-automatic dimensioning capability with a way of associating the dimensions with the entities. As a result, once dimensioning is created, there is no need to redo it after modifications to the drawing. The following screen (Fig. 6.25) shows the typical appearance of the dimension menu.



Current dimension style: Standard

Description: Standard

List: All styles

Modify Dimension Style: Standard

Dimension lines

Color: ByBlock

Linetype: ByBlock

Lineweight: ByBlock

Baseline spacing: 0.3800

Suppress: ☐ Dim line 1 ☐ Dim line 2

Extension lines

Color: ByBlock

Linetype ext line 1: ByBlock

Linetype ext line 2: ByBlock

Lineweight: ByBlock

Suppress: ☐ Ext line 1 ☐ Ext line 2

Extend beyond dim lines: 0.1800

Offset from origin: 0.0625

☐ Fixed length extension lines

Fig. 6.25 *The Dimension style option*

While dimensioning, the information to be specified is as follows:

- where is the dimension
- where the dimension text should go
- how big and what style the text will be
- whether tolerance range is to be included
- how big and what the arrows will look like

To set these values, a number of variables are available in AutoCAD whose values need to be set in the prototype drawing. These variables actually control the way the dimensions appear in the drawings. AutoCAD gives great control over the way dimensions may appear in the drawings. It is therefore necessary that users should be familiar with these in order to customise the dimensioning to the best of the methods used in the design office.

Each of these can be further specified based on dimension families or for all of them. The dimension families are specified as follows:

- Linear
- Diameter
- Radial
- Angular
- Ordinate
- Leader

The procedure to be followed in dimensioning in AutoCAD is as follows:

- Set up the basic parameters for dimensioning. They are
- Arrowhead type
- Arrowhead size
- Extension-line offset
- Placement of dimension text

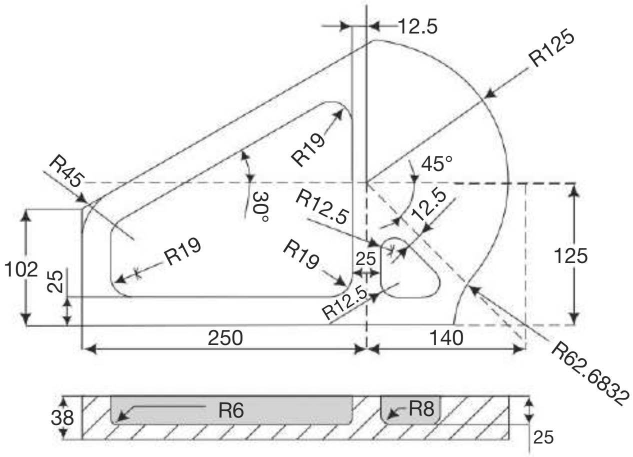


Fig. 6.26 *Example drawing made in AutoCAD*

These need not be changed for every drawing but could be incorporated in the prototype drawing itself.

- Identify what the user wants to measure

Pick end points, lines, arcs or circles, or other points of existing drawing entities using OSNAP if necessary.

- Specify where the dimension line and text are to be located.
- Approve AutoCAD's measurements as dimension text or type in your own text.

An example made in AutoCAD is shown in Fig. 6.26.

Summary

- AutoCAD is indeed a versatile CAD software which is widely used on PC platforms. There are a number of facilities available in AutoCAD that are quite intuitive and useful for quick development of drawings.
- The user interface of AutoCAD is akin to other Windows software, and allows for easier understanding and integration with other Windows-based software.
- It is important to plan properly a drawing before committing to actually develop it in AutoCAD. To that extent, there are a number of facilities available in AutoCAD such as layers, units, etc.
- A number of basic geometric entity drawing commands are available in AutoCAD such as line, circle, polygon, etc., that help in easier construction of a drawing. In each of these commands, a variety of options are available that can help in directly transferring the dimensions directly.
- Great productivity in developing a drawing is achieved by exploiting the symmetry present in a drawing. To that extent, a number of editing commands are available in AutoCAD, which are easy to understand and apply. Examples of editing commands are copy, move, array, trim, break, offset, etc.

Questions

1. What are the facilities that are useful for creating geometric entities in a drafting system?
2. What are the facilities that are useful for editing geometric entities in a drafting system?
3. What types of typical dimensioning facilities are available in a drafting system?
4. What are the various display control commands available in a drafting system?
5. Give details of a few editing commands used in a drafting system.

7

INTRODUCTION TO MODELLING SYSTEMS

Objectives

Modelling systems have increasingly become common among the industries in view of their affordability and ability to run on the available low-cost desktop computers. In this chapter, a few of the common aspects of these modelling systems are studied. After completing the study of this chapter, the reader should be able to

- Learn the basics of an affordable and easy-to-use parametric modelling system running on a desktop computer and the example chosen is the Autodesk Inventor

7.1 INTRODUCTION

The previous chapter gave the details of the drafting system, which is basically a 2D data representation. However, the case of modelling a part in 3D requires greater imagination. As discussed earlier, the facilities required for 3D modelling are far greater compared to drafting. There are a large number of modelling systems that are generally used by the industry. However, the following are some of the higher-end CAD/CAM systems.

- Unigraphics
- Pro Engineer
- CATIA

Though these have fairly comprehensive facilities for modelling and assembly of complex parts, these are expensive and can only be used by large corporations. To cater to the needs of small and medium enterprises, some CAD systems are being developed, and the following are some of these systems which are currently in the market that are popular.

- Solidworks
- Autodesk Inventor
- Iron CAD

Most of these systems provide comprehensive modelling facilities. These normally are powerful CAD systems with hybrid modelling facilities. The modelling methods that are embedded in them are

- Solid modelling
- Surface modelling
- Wireframe modelling
- Feature-based modelling

These systems seamlessly integrate constraint-based feature modelling and explicit geometric modelling. The standard design features include several varieties of holes, slots, pockets, pads, bosses, as well as a full set of cylinders, blocks, cones, spheres, tubes, rods, blends, chamfers and more. Freeform modelling forms the basis for a unique merging of both solid- and surface-modelling techniques.

7.2 CONSTRAINT-BASED MODELLING

A brief view of the constraint-based modelling process is given below. We will now take up an example of modelling a component in Autodesk Inventor release 2010. The process will be similar in other modelling systems that follow feature-based modelling. The only difference may be in terms of the command and menu structure. The opening display of the Inventor is shown in Fig. 7.1 with a screen structure very similar to other Windows application programs.

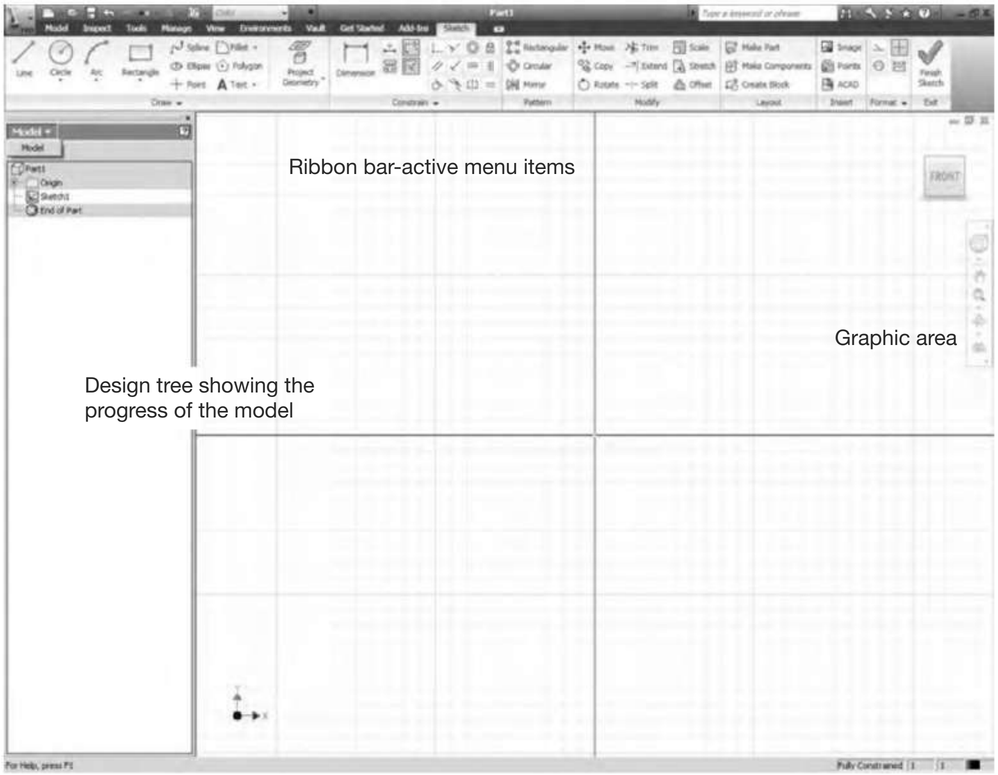


Fig. 7.1 *The opening screen of Inventor release 2010*

The main approach used in the modelling is to identify the various features in the model and then arrange them in the order in which they will be modelled. Once the ordering is done, the user starts with a sketch plane, which is two-dimensional as shown in Fig. 7.1. The sketch is then swept or extruded along with the necessary Boolean operations as we have discussed earlier. We will develop the total modelling procedure for the component shown in Fig. 7.2 to understand the concepts of feature-based modelling.

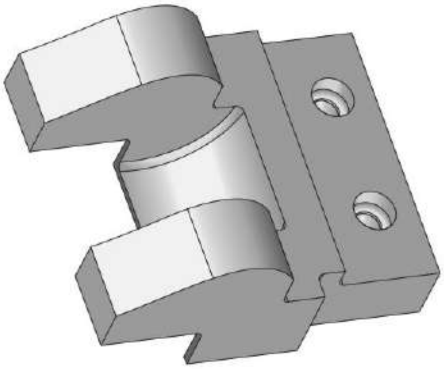
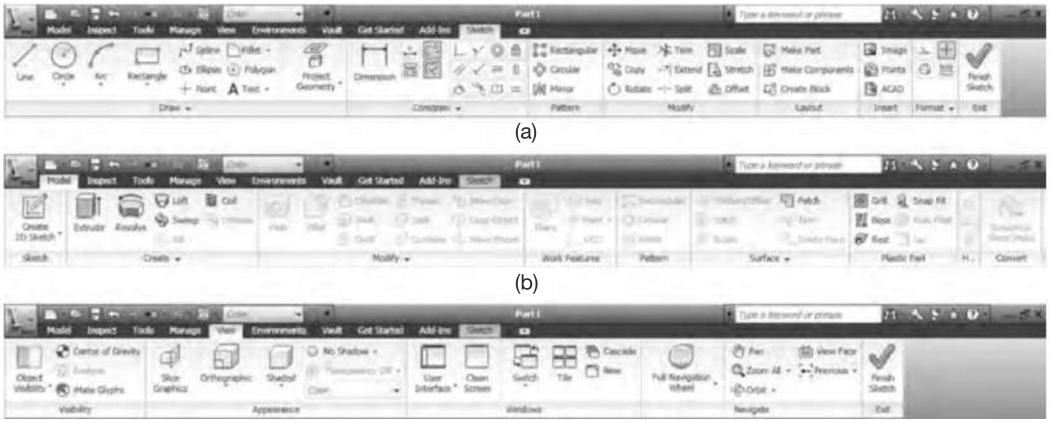


Fig. 7.2 *Example part that will be modelled*

The first step in the modelling process is to select a sketch plane to generate the base feature. The approximate shape of the profile can be generated using the various 2D tools available as shown in the left-hand pane in the screen such as line, circle, arc, etc., similar to any two-dimensional drawing. The various facilities available for modelling are shown in Fig. 7.3. A brief description of these facilities is given below:

Line/Spline This is a line tool. It can also be used for making a spline through a set of points. The type of lines that can be drawn are normal lines, construction lines, which are not part of the geometry, or centre lines which are used as axes for creating the rotational symmetric objects.



(a)

(b)

(c)

Fig. 7.3 *Various tool ribbons showing the facilities available: (a) sketch, (b) model, and (c) view options*

When the line command is in progress, pressing the right mouse button brings the menu shown in Fig. 7.4, which allows for picking the right point (midpoint, centre point or intersection point) from the objects that are already present in the sketch.

Circle/Ellipse It creates a circle with the centre and radius. Other options available are ellipse and tangent circle.

Arc It generates an arc through three points. Other options available are the centre, start and end points, and tangent.

Rectangle It generates a rectangle through two corner points. Another option available is to specify the two sides of the rectangle.

Fillet/Chamfer Normally fillets and chamfers are better given as features since they can be easily edited. However, when needed they can be used at the sketch level also. Its operation is similar to other CAD programs.

Point, Hole Center This is used to specify the location for the hole feature.

Polygon Creates a polygon with a maximum of 120 edges.

Mirror Creates a mirror of the selected geometry elements in the sketch. A pop-up window opens giving all the necessary instructions for selecting the objects and mirror line.

Rectangular Pattern It creates a rectangular pattern of the selected entities. A pop-up window opens giving all the necessary instructions for selecting the objects and other parameters. It is possible to suppress some of the instances that are being patterned.

Circular Pattern It creates a circular (polar) pattern of the selected entities. A pop-up window opens giving all the necessary instructions for selecting the objects and other parameters. It is possible to suppress some of the instances that are being patterned.

Fig. 7.4 *Right mouse button menu in line command*

Offset It is used to offset the geometry elements.

General Dimension It is used for dimensioning of a sketch. Inventor identifies the object being dimensioned and automatically decides the type of dimension to be used. Pressing the right mouse button inside the menu brings up a submenu for the type of dimensions as shown in Fig. 7.5. This gives the actual value of the dimension as sketched. The user can modify the value of the dimension as required. When the dimension is changed, the sketch is automatically modified to reflect the change.

Auto Dimension This command automatically identifies the number of dimensions and constraints needed to fully constrain the sketch. The user can edit the dimensions as required.

Extend This command helps to extend a given object. The operation is different to AutoCAD in the sense that there is no need to specify the boundary. With the mouse, the user can drag the end of the object to where it is required in the sketch plane.

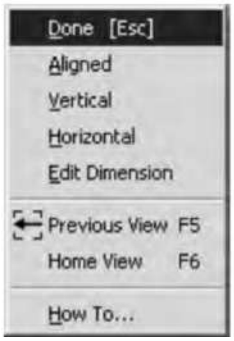


Fig. 7.5 *Right mouse button menu in General Dimension command*

Trim This command helps to trim the objects as required. Inventor automatically selects any of the cutting edges that are crossing the object to be trimmed.

Move This command is used to move or copy an object. This command opens a dialog box through which the necessary options are given.

Rotate This command is used to rotate an object in the sketch plane. This command opens a dialog box through which the necessary options are given.

Constraints This command allows adding constraints to the geometry. The available constraints are Perpendicular, Parallel, Tangent, Coincident, Concentric, Collinear, Horizontal, Vertical, Equal, Fixed, and Symmetric.

Show/Delete Constraints This command shows the constraints that are present on the individual entities. If necessary, pressing the right mouse button brings a pop-up menu to allow for the deletion of any constraint.

Project Geometry It creates additional geometry elements in the current sketch plane by projecting the geometry from other planes that were present in the model or in other sketch planes.

Insert AutoCAD File This is used to bring any AutoCAD drawing as a sketch.

Edit Coordinate System When we start developing, the default coordinate system is located which is shown on the graphic screen at the lower left-hand corner. The location of that can be changed using this option. Always look for the messages shown in the message box to help you through the tasks.

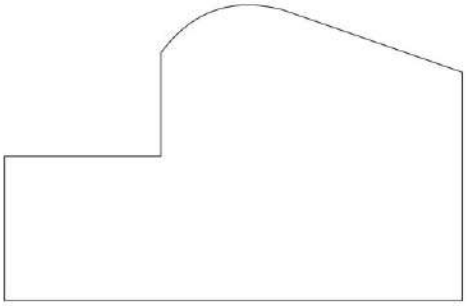


Fig. 7.6 *Sketch for the base feature for the example shown in Fig. 7.2*

All the tools as shown above are used to quickly sketch the approximate shape of the object in the sketch plane. At this stage there is no need to consider the exact dimensions and the constraints. These can be added later. The sketch generated is shown in Fig. 7.6.

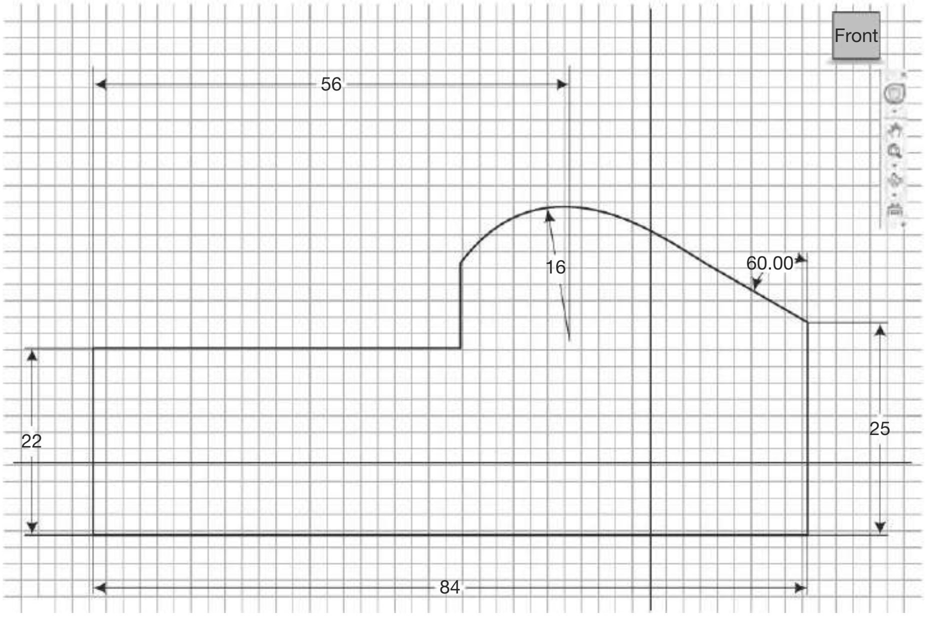


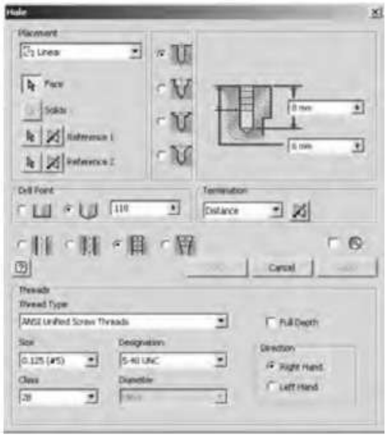
Fig. 7.7 *Sketch is now dimensioned with the actual values*

Using the general-dimension tool, the user selects the various dimensions that are required to completely define the sketch. A large number of design rules are embedded in the system, which are invoked by the system during the entire stage of the sketching process and suitable feedback is provided to the user. At this stage, it is necessary to provide other constraints such as parallel or perpendicular lines, or certain angles, etc. The dimension values can be given direct values or they can also be designed as variables whose values can be specified at a later stage. Also, a dimension can be given in relation to other dimensions as well. There are a large number of possibilities depending upon the design requirement. The completely dimensioned and constrained sketch for our model is shown in Fig. 7.7.

We are now ready to generate the first feature, which will act as the base feature for building all others to finally reach the component that we want to model. There are a number of feature-creation facilities available in Inventor as shown in Fig. 7.3b.

Extrude This command allows generating a solid by extruding a given closed profile. This command is used to generate a linear or tapered extrusion in a direction perpendicular to the sketch profile similar to the examples shown in Fig. 4.7 and 4.8. It opens a dialog window through which all the necessary options can be given. The generated solid from this command can be automatically Boolean operated on any solid volume already present in the following manner.

Join Adds the feature created in this command to that of the other volume present.



Placement: Linear

Drill Point: 118

Termination: Distance

Thread Type: ANSI Unified Screw Threads

Size: 0.125 (#5)

Designation: 5-40 UNC

Class: 2B

Direction: ● Right Hand ○ Left Hand

Fig. 7.8 *A hole Feature dialog box in Inventor 2010*

Cut It removes the volume created in this command from the other volume already present.

Intersect Creates a new feature from the shared volume of the extruded feature and another feature already present. Material that is not included in the feature from between the two features will be removed.

Surface Creates a 3D surface.

Revolve This command is used to create axi-symmetric objects by revolving the selected geometry about an axis, and the resultant feature looks similar to Fig. 4.12. The axis can be separately defined using the centre line option and used. The profile and axis must be coplanar. It is possible to create a complete revolve or partial, by specifying the necessary angles. The required options can be given through a pop-up window. The generated solid from this command will be automatically Boolean operated on any solid volume already present similar to the extrusion option.

Hole This command creates a parametric hole feature for simple holes, counter-bored holes or countersunk holes. The holes can be either through or blind. The hole can also be a tapped hole. The various parameters required for the hole can be specified through the pop-up dialog box, which shows all the options available as in Fig. 7.8. In the case of threads option, all the necessary standard thread data is incorporated into Inventor and can be obtained by selecting the thread option. In the case of a blind hole, the details of the bottom shape of the hole can be specified using the options.

Shell This option removes the material from the interior of the part such that a hollow part is prepared. The faces of the solid from which the material is to be removed is to be selected by the user. The thickness of the part after the removal of the material can be from the outer surface, inner surface or on either side by selecting the appropriate button in the pop-up window. The use of this feature is shown later in the example.

Rib This command creates an extrusion of geometric element(s) by creating a feature connecting the two faces of the part. It is used for creating the ribs and webs in castings.

Loft It creates a solid similar to extrusion, except that it blends a number of sketches by creating the volume along all these sketches with uniform blending. The sketches can be created on any planes that are not perpendicular. However, generally planes parallel to each other are used for creating the sketches. A typical example of a loft solid is shown in Fig. 7.9.

Sweep This command creates a feature by moving a sketch (closed or open) along a planar path, which is not necessarily perpendicular to the sketch plane. The profile sketch plane and plane of the path through which it will be swept should not be intersecting. A typical example of a swept solid is shown in Fig. 7.10.

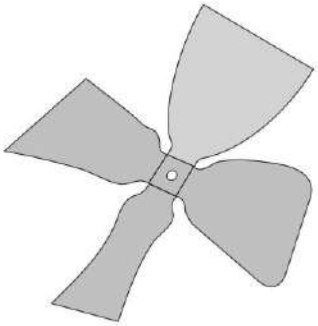


Fig. 7.9 *A loft solid created in Inventor 2010*

Fig. 7.10 *A solid created using the SWEEP command in Inventor 2010*

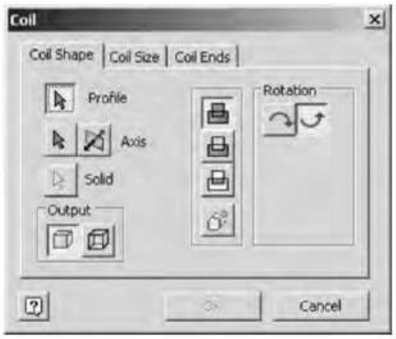


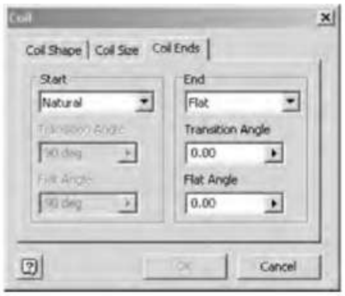
Fig. 7.11 *The pop-up window for the COIL option in Inventor 2010*

Coil This tool helps create the helical coils such as helical springs. The user has to select a profile for the cross-section of the coil and the axis along which the helical coil will be generated. Both the profile and the axis should be in the same sketch, unless the axis is a work axis. The various options required for the shape of the coil are specified through the pop-up dialog as shown in Fig. 7.11. The coil size option specifies the manner in which the helical geometry is specified as shown in Fig. 7.12. Finally, the shape of the coil ends are specified as shown in Fig. 7.13.

Thread This is a tool that simulates a thread by actually mapping a bitmap of thread around a cylindrical surface. The information about thread parameters are given using the pop-up window shown in Fig. 7.14. This information is used only for drafting purposes.

Type: Pitch and Revolution
Pitch: 10 mm
Height: 15 mm
Revolution: 1.00
Taper: 0.00

Fig. 7.12 *The pop-up window for the COIL to specify the helical geometry in Inventor 2010*



Start: Natural
Transition Angle: 90 deg
Flat Angle: 90 deg
End: Flat
Transition Angle: 0.00
Flat Angle: 0.00

Fig. 7.13 *The pop-up window for the COIL to specify the shape of ends in Inventor 2010*

Thread Type: ANSI Metric M Profile
Size: 15
Designation: M15x1.5
Class: 6g
● Right hand
○ Left hand

Fig. 7.14 *The pop-up window for the THREAD option in Inventor 2010*

Fillet It creates a fillet between any two planes. It is generally suggested that the fillet be used after completing most of the design work, so that it is easier to edit and regenerate the model. Inventor automatically selects all the interconnecting edges for filleting. It is possible to create a constant radius or variable radius fillet. All the necessary options are specified using a dialog window that opens when the Fillet option is selected.

Chamfer This creates a chamfer of specified parameters along an edge. All the necessary options are specified using a dialog window that opens when the Chamfer option is selected.

Face Draft This is a taper applied only to a particular face unlike the taper option in Extrusion, which generates a taper along all the faces.

Split This tool is used to create two halves of a part at any specified plane such as a simple parting plane for a die-casting die. This tool actually deletes the split part. If both the split parts are required, then the user has to save the part with the split plane, and then carry out the split operations separately to create the two halves in two different files.

Rectangular Pattern This is used to create a rectangular pattern of a given feature. All the necessary options are specified using a dialog window that opens when the Rectangular Pattern option is selected. It is possible to suppress some of the instances as required.

Circular Pattern This is used to create a circular pattern of a given feature. All the necessary options are specified using a dialog window that opens when the Circular Pattern option is selected. It is possible to suppress some of the instances as required.

Mirror Feature This is used to mirror features in the model. All the necessary options are specified using a dialog window that opens when the Mirror Feature option is selected.

Work Plane Use this option to create a plane using a number of options available for creating new sketches. A number of options could be used for creating a plane such as any geometry from the unconsumed sketches, normal to geometry, parallel to geometry or at an angle to geometry.

Work Axis This creates an axis that could be used for other functions in defining geometry or features.

The base feature is obtained by extruding or sweeping the previous sketch in a perpendicular direction by a distance of 70 mm. The result is shown in Fig. 7.15, which is the base feature.

Next we would like to generate the slot in the side of the base feature. To do this, first identify the sketch plane as the side plane and then sketch a line on this face parallel to the vertical line as shown in Fig. 7.17. The sketch plane is identified by means of grid line display as shown in Fig. 7.16. Identify the dimension as parallel distance between the two lines as 32 mm. This helps in generating the slot by the sweeping process as was done earlier. However, this time after extruding we would be interested in subtracting that volume from the base feature to generate the slot. The slot of 6-mm depth is then generated as shown in Fig. 7.17.

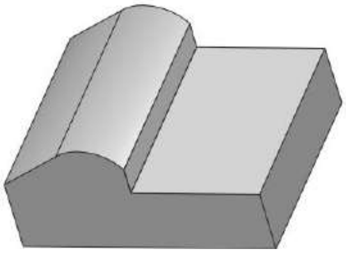


Fig. 7.15 *The base feature generated by sweeping the sketch shown in Fig. 7.7*

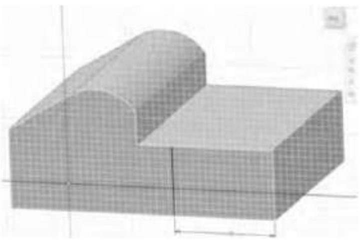


Fig. 7.16 *New sketch plane for generating the slot; parallel line to the vertical corner edge can be seen in the sketch*

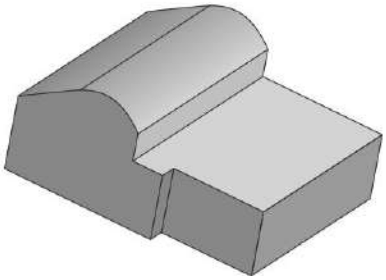


Fig. 7.17 *The sketch is swept with 6-mm depth and then subtracted from the base feature to generate the slot*

Next, we need to generate the slot in the bottom face. Rotate the current model to expose the bottom face and then select the bottom face to work for the next part of the sketch. Sketch a straight line parallel to the bottom edge and dimension it to 25 mm from the edge as shown in Fig. 7.18. Now, sweep this rectangular geometry formed by the sketched line and the other edges of the base feature to get the shape of the slot. That slot is subtracted (extruded cut) from the base feature to get the shape of the part at this stage as shown in Fig. 7.19.

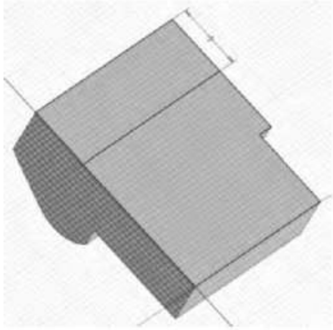


Fig. 7.18 *New sketch plane for generating the slot in the bottom face*

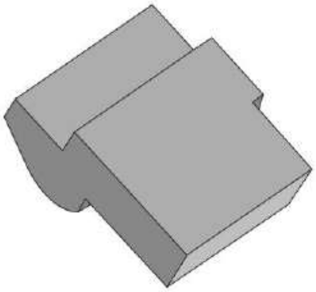


Fig. 7.19 *The sketch is swept with 13 mm depth and then subtracted from the base feature to generate the slot in the bottom face*

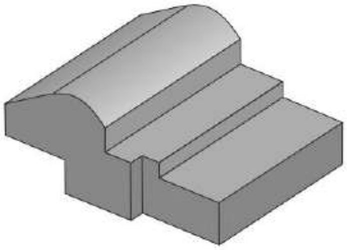
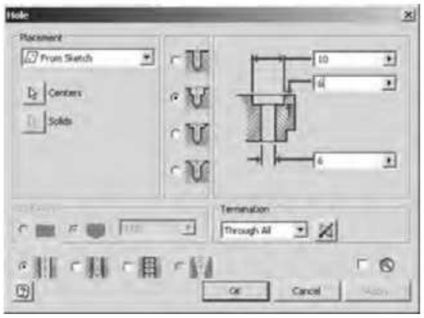


Fig. 7.20 *The sketch is swept with 6-mm depth and then subtracted from the base feature to generate the slot in the top face*

Next, we need to generate the slot in the top face. Rotate the current model to expose the top face and then select it to work for the next part of the sketch. Sketch a straight line parallel to the bottom edge and dimension it to 25 mm from the horizontal edge. Now, sweep this rectangular geometry formed by the sketched line and the other edges of the base feature to get the shape of the slot. That slot is subtracted (extruded cut) from the base feature to get the shape of the part at this stage as shown in Fig. 7.20.

The next feature to be added is the two counter-bored holes in the slot that was just modelled. Select the slot face for adding the two holes by making it the work plane. Sketch the two hole positions at distances of 16 and 46 mm from the right-hand side edge. Now insert the hole feature with the dimensions and structure as shown in Fig. 7.21. The generated part is shown in Fig. 7.22.



Hole

Placement: From Sketch

10

6

6

Termination: Through All

Fig. 7.21 *The counter-bored hole features to be modelled with their dimensions*

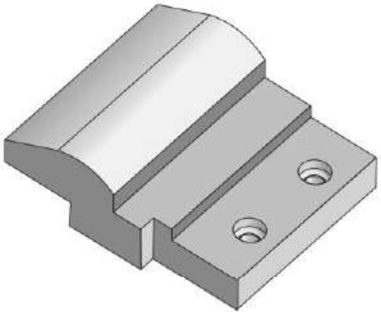


Fig. 7.22 *Model of the part after adding the two counter-bored holes*

The next step involves a more difficult modelling operation. It requires a working plane parallel to the side face and located at the middle of the part as shown in Fig. 7.23. This can be done by touching the mouse pointer on the side face and then the middle of the bottom edge of the part. Select this work plane to form the sketching plane for the next part. Since we want to work in the middle plane, the part is now sliced at the midplane and the display is shown as in Fig. 7.24 using the option Slice Graphics. The slicing is only for the display and not on the model. Now project the contour of the part on to the sketch plane so that they will be available for forming the sketch for extrusion using the option Project Geometry. The projected lines can be seen in the sketch plane in Fig. 7.24. Now draw a circle concentric to the arc in the sketch plane with a radius of 21 mm. Trim parts of the circle with the projected edges such that we get a closed geometry as shown in Fig. 7.25.

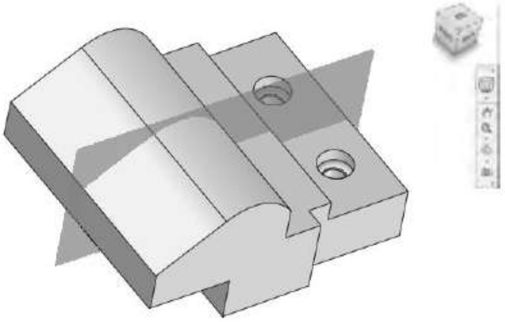


Fig. 7.23 *A new work plane defined that is parallel to the side face and at the middle of the part*

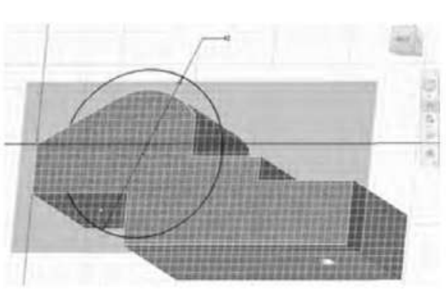


Fig. 7.24 *The new sketch plane and sliced part for construction purpose only. The edges are projected on to the sketch plane to be used for extrusion*

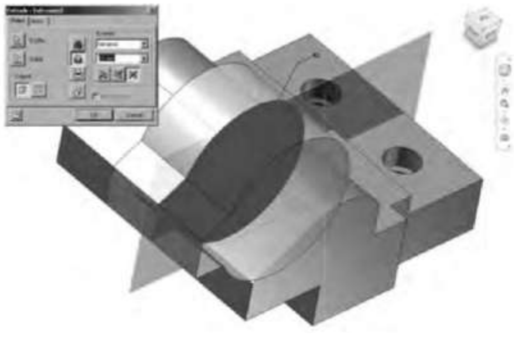


Fig. 7.25 *The topology of the sketch for extrusion to form the cut in the middle of the part*

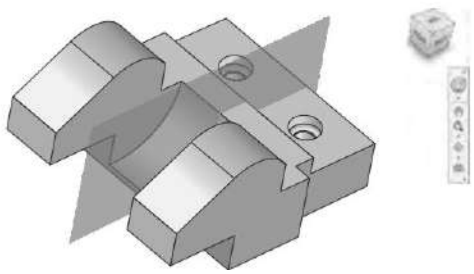


Fig. 7.26 *Model of the part after subtracting the midplane extruded volume*

Now extrude this geometry to a depth of 33 mm and cut from the part. However, this extrusion needs to take place not in a single direction but equally on both sides of the sketch plane as shown in Fig. 7.25. The resultant part is shown in Fig. 7.27.

We have completed the major modelling part and it is visible in the display. The next step is to shell the part. Shells are parametric features used generally in cast and moulded parts with uniform wall thickness.

Fig. 7.27 *Model of the part applying the shell feature*

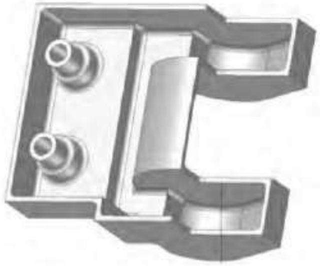
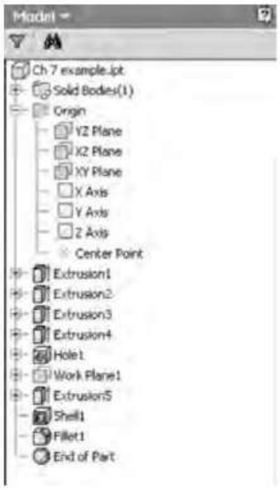


Fig. 7.28 *The final model generated showing the bottom side view of the part*

Fig. 7.29 *The final model generated showing the top side view of the part along with all the hidden line displays*

Material from the interior part is removed, leaving a hollow cavity. Any changes carried out later for the dimensions of either the part or the shell automatically resize both. Features added to a part after the shell operation is applied will not be shelled. So care needs to be taken to see that the shell is applied after completing all the modelling of the part. To apply the shell, specify the part thickness and the faces from which the material is to be removed (in this case it is all the bottom faces and the shell thickness is 2 mm). The resultant shape of the part is shown in Fig. 7.27.

The last operation that we will be carrying out will be to fillet all the edges so that the sharp corners are removed. A uniform fillet radius of 1.6 mm is applied to all the edges which are automatically identified by the system. The resultant part is shown in Fig. 7.28. The top side view of the part showing all the hidden details is shown in Fig. 7.29. The modelling process that we have undertaken so far is shown in Fig. 7.30 in the form of a model tree.



- Ch 7 example.ipt
  - Solid Bodies(1)
  - Origin
    - YZ Plane
    - XZ Plane
    - XY Plane
    - X Axis
    - Y Axis
    - Z Axis
    - Center Point
  - Extrusion1
  - Extrusion2
  - Extrusion3
  - Extrusion4
  - Hole1
  - Work Plane1
  - Extrusion5
  - Shell1
  - Fillet1
  - End of Part

Fig. 7.30 *The modelling process undertaken to generate the final part is shown with all the steps*

It can be seen that feature-based modelling is a very powerful technique and can be used to first-time model as well as modify with relative ease, thereby saving a lot of effort and time in the total design process. As a result, practically all the modern CAD systems make use of this approach and extend the same methodology for all the downstream applications as well. In this course we do not have sufficient time to go further into the subject. The student can try to have hands-on experience on any one of the modelling systems to gain further insight into the subject.

Summary

- The modelling system discussed in this chapter has comprehensive facilities for practically most of the modelling tasks that are required for manufacturing industries. It covers the full spectrum of applications for medium scale to large scale manufacturers.
- Inventor can be used for applications involving large assemblies and complex modelling requirements.

- The systems are very user-friendly.
- The modelling facilities provided in Inventor are a comprehensive set covering wireframe modelling, surface modelling as well as solid modelling. All these modelling methods are seamlessly integrated.
- It also has the parametric modelling capability, which is becoming increasingly important with most of the current designers.

Questions

1. Briefly explain the modelling using a CAD system.
2. What are the facilities available for sketching in a 3D modelling system?
3. What are the various construction facilities available for modelling in a 3D modelling system?

8

FINITE ELEMENT ANALYSIS

Objectives

Use of finite element analysis has becoming increasingly popular with many of the industrial designers in view of its affordability. A large variety of systems are available in the market that not only provide the functionality but make it easier for users in its usage. After completing the study of this chapter, the reader should be able to

- Study the application of finite-element-analysis methods
- Understand the basic concept of finite element modelling from the conventional analytical principles of elasticity procedures
- Introduction to the various software modules and methods required to carry out the Finite Element Modelling (FEM) and analysis
- Learn the various types of analysis that are possible and how they are used
- Understand the process of using FEM to model trusses
- Learn the process of using FEM to model the beams
- Understand the process of using FEM to model the plane stress parts

8.1 INTRODUCTION

Conventional analytical methods at solving for stresses and strains become very complex and almost impossible when the part geometry is intricate. In such cases, Finite Element Modelling (FEM) becomes a very convenient means to carry out the analysis. The Finite Element Analysis (FEA) is a very powerful analysis tool, which can be applied to a range of engineering problems.

The finite-element-modelling process allows for discretising the intricate geometries into small fundamental volumes called *finite elements*. It is then possible to write the governing equations and material properties for these elements. These equations are then assembled by taking proper care of the constraints and loading, which results in a set of equations. These equations when solved give the results that describe the behaviour of the original complex body being analysed.

Application of FEM is not limited to mechanical systems alone but to a range of engineering problems such as

- Stress analysis
- Dynamic analysis
- Deformation studies
- Fluid-flow analysis
- Heat-flow analysis
- Seepage analysis
- Magnetic-flux studies
- Acoustic analysis

A large number of commercial software for the application of finite elements is available. Generally, to solve a complex problem, a very powerful computer is needed. However, with the developments in Intel Pentium processors, it is possible to carry out such analyses in most of the present-day desktop computers. With the FEM software, it is possible to try a number of alternative designs before actually going for a prototype manufacture. For example, in the case of building an injection mould, it is possible to see how the plastic material flows into the mould cavity and how it gets solidified. After solidification, it is also possible to see the shrinkages and weld marks. Based on this knowledge it is possible to modify the mould to get the right component first time.

Some definitions that are used in finite element analysis are given below:

Mesh Subdivided part geometry is called a mesh. The process of subdivision is called meshing.

Element In finite element analysis, part geometry is divided into small volumes to solve it easily. Each such volume is called an element.

Node Each element has a set of points called nodal points or nodes for short. Nodes are usually located at the corners or endpoints of elements. In the higher-order elements, nodes are also placed on sides or faces, as well as possibly the interior of the element.

Degrees of Freedom Degrees of Freedom (DOF) specify the *state* of the element. Normally, each node has six degrees of freedom in static analysis. These are three linear displacements along the rectangular coordinate axes and the other three are the rotary motion about these axes.

Nodal Forces These are the forces that are applied on the finite element model. These are generally the concentrated forces at the nodes.

Boundary Conditions These specify the current state of some nodes in the finite element mesh. This is a way to specify how some of the nodes in the model are constrained.

Dimensionality Elements that are used in FEM can have intrinsic dimensionality of one, two or three space dimensions. This dimensionality can be expanded for higher dimensions by kinematic transformation. For example, a one-dimensional element such as a bar or a beam may be used to build a model in 2D or 3D space.

8.2 FEM SOFTWARE

In the early days of finite-element-method development, the total process of solving a particular problem using the finite element method had to be done by individuals. That means the steps involved were as follows:

- Converting the geometry into the discretised elements and calculating for each of the elements the various properties such as geometry, material properties, constraints and loading. This formed the input for the analysis.
- Assembling the global equations from the formulation specified in the previous stage and then solving the governing equations and obtaining the results.
- Interpreting the results in the specific form depending upon the type of problem solved.

Carrying out of all these processes manually to a great extent and then use the home-grown computer programs mostly written in FORTRAN and getting a large volume of results was a tedious task. It was further compounded by the interpretation of the results which required the sifting of a large amount of data and converting it into tables, graphs or contour plots—a nightmarish exercise.

All this changed by the availability of commercial comprehensive Finite Element Analysis (FEA) software that could run on desktop computers and made the job of an engineer relatively simple in applying FEA. This was further helped by the rapid development in the desktop computer capability which allowed practically the most comprehensive FEA to be carried on the desktop computer, whereas earlier these used to be done on the supercomputers or mainframe computers that were available only with large corporations.

The commercial FEA solver generally has the same three-stage approach as shown in Fig. 8.1.

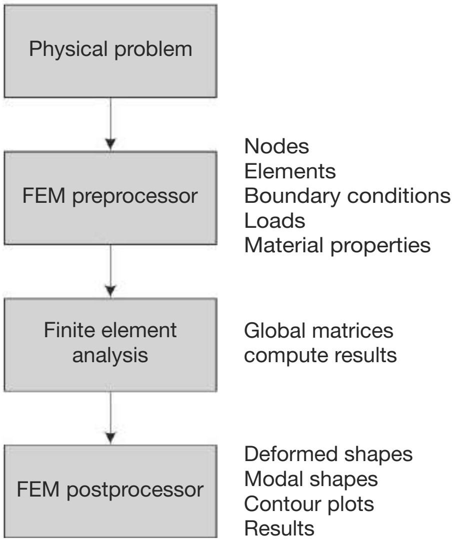


Fig. 8.1 *Steps involved in the use of finite element method for solving a physical problem*

8.2.1 Preprocessor

The preprocessor of FEM software allows for the following functions:

- Modelling of the geometry
- Defining the element types to be used
- Generating the finite element mesh by making a suitable approximation to the geometry
- Calculating the nodes and elemental properties
- Allowing for the specification of the support condition and loading conditions for the individual element position
- Allowing the material properties to be specified

Generally, the modelling function provided in the FEM software is relatively cumbersome compared to the more user friendly facilities found in the major CAD packages. As a result, there is a possibility that a CAD system can be used to input the geometry which can then be interfaced with the FEM software in a number of ways as shown in Fig. 8.2.

One of the possibilities is the direct linkage in which the geometric model from the CAD system goes into the preprocessing part of the finite element software. For this purpose, generally some neutral data format is used. The most common one is the DXF or IGES format for many of the low-end packages. Another possibility is that of a preprocessor present within the CAD software, which converts the geometric model

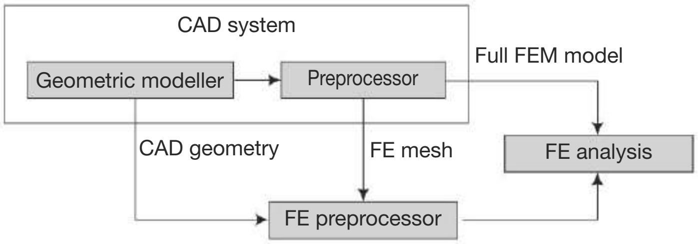


Fig. 8.2 *Integration of CAD systems with the finite element methods*

into the finite element mesh and then transmits that data into the preprocessor of the finite element software which then can assemble the rest of the options such as the specification of the boundary conditions. This requires that the element types available in the FE software be same as those that are available in the CAD software. The types of elements to be used for generating the mesh depend upon the actual geometry and the type of problem being considered.

Still another possibility is that the CAD package has a complete specialised preprocesor built in which can do the complete mesh generation along with the specification of all the boundary conditions as well. Then it can directly link with the solver component of the FE software. A typical geometry and the corresponding mesh generated in shown in Figs. 8.3 and 8.4 respectively.

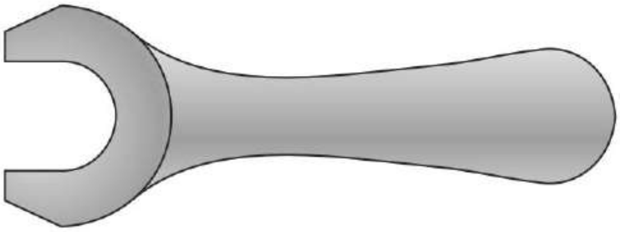


Fig. 8.3 *A spanner modelled directly in CAD software*

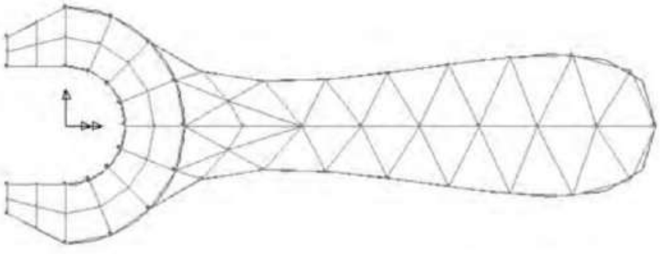


Fig. 8.4 *Spanner as in Fig. 8.3 with the generated mesh using FEA software*

Once having completed the conversion of the geometry into the finite element form, it is to be solved using the solver available within the FE software. The results will be generated after all the equations as assembled are solved.

The types of elements to be used for generating the mesh depend upon the actual geometry and the type of problem being considered. Some typical elements generally found in most of the FE software are shown in Fig. 8.5.

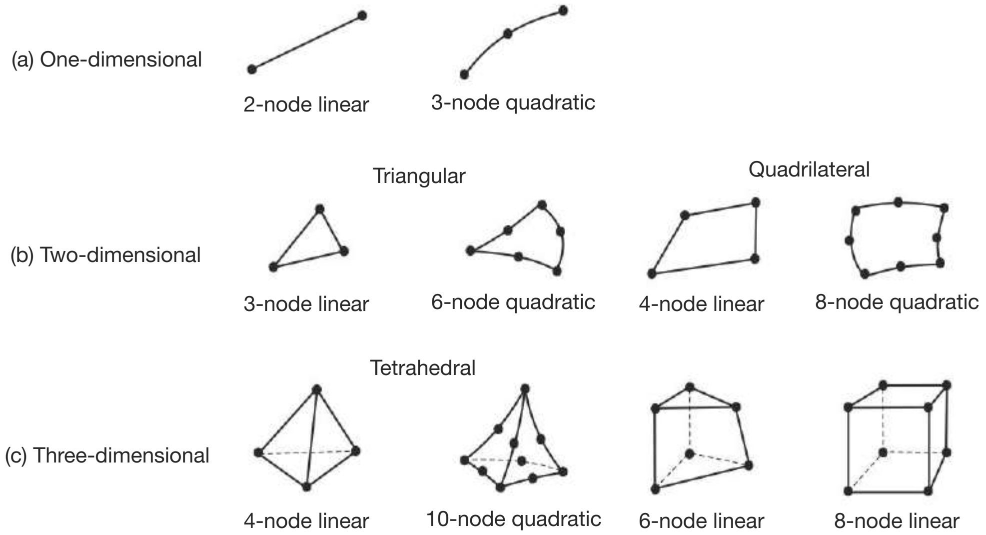


Fig. 8.5 *Typical elements available in the commercial finite-element-analysis software*

8.2.2 Postprocessing

Postprocessing involves the ability to go through a large amount of data generated during the solving process and convert it into an easily understood form for the design purpose. For this purpose, many facilities are available within the FE software. Alternatively, some specialised postprocessor software may also be used which are part of the CAD suite.

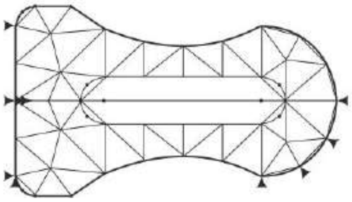


Fig. 8.6 *Cam plate showing the geometry, the finite element mesh and the boundary conditions used*

An example generated using FEA software is shown in Fig. 8.6. It shows a cam plate with the geometry and the finite element mesh generated. It is supported from the\ left side as shown by the areas in Fig. 8.6. The load acting is on the bottom-right edge of the clamp. This model is now ready for running the analysis.

After a successful running of the analysis, the results generated can be postprocessed to various forms depending upon the requirements envisaged in the design process. For example, the von Mises stress-contour plot for the cam plate is shown in Fig. 8.7. It can be noticed that the display consists of full legends in terms of the values of stresses and displacements. It is also possible to slice the component and get such contour plots at those slices.

8.2.3 History

The numerical procedure to obtain the solution of complex problems by breaking them down into small segments was credited to Courant in early 1940s. However, the name 'Finite Element' was first used by

Clough in 1960. In the 1960s, the finite element method was applied to many engineering problems. The main problem in applying the finite element method is the availability of sufficient computing resources based on the capability of the computers of that era. In 1970s, the finite element method could only be run on large mainframe computers. With the availability of personal computers in the 1980s, the use of the finite element method moved to desktops. Though initially only preprocessing and postprocessing was done on personal computers with the actual solving done on mainframe computers, with the availability of Intel Pentium processors, the solving part also moved to personal computers in 1990s. With the current level of processing power on desktop computers, it is practically possible to run really large problems with great ease. Also, the user interface of most of the FEA software has made it easier for practicing engineers to easily access the enormous capability of the method for all types of analysis.

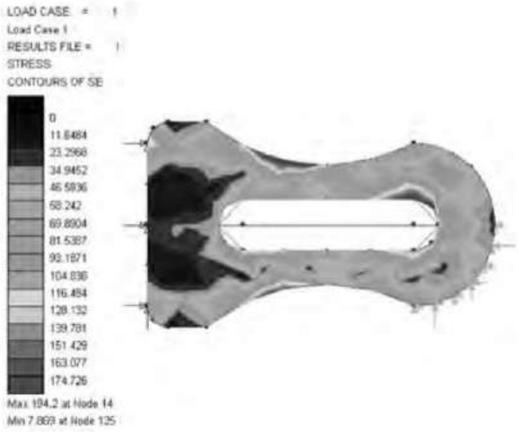


Fig. 8.7 *Stress contours of the cam plate as in Fig. 8.6*

8.2.4 Advantages

There are many advantages of using Finite Element Analysis (FEA) during the design stage.

- Parts with irregular geometries are difficult to be analysed by the use of conventional strength of material approaches. In FEA, any complex geometry can be analysed with ease.
- Parts made from different materials can be analysed using FEA.
- It is possible to analyse parts that have complex loading patterns with multiple types of forces acting on the geometry with large number of supports.
- The FEA procedure provides results throughout the part (all points).
- It is easy to change the model in FEA and generate a number of options with 'what-if' scenarios. This helps in developing faster prototypes and speeds up time to market by shortening the design cycle.
- Testing of products can be done using FEA without expensive destructive testing.

It is also important to recognise the limitations of FEA. Commercial FEA has excellent capabilities and it is possible that an inexperienced user can deliver incorrect answers without fully understanding the implications. Using FEA is not simply understanding the user interface, but understanding the underlying engineering in the process. Hence, it is important that the user is very familiar with engineering aspects of the problem and has considerable experience in using it for design.

8.3 STIFFNESS MATRICES

In order to understand the concept of finite element modelling, we will considerable a one-dimensional problem which is easier to visualise. The same concepts can be extended to two and three dimensional problems similarly. The type of analysis that will be considered is the linear static analysis with the following assumptions:

- Small changes in stiffness
- No changes in loading direction

- Material remains in the linear elastic range
- Small deformation and strain

Consider a taper beam as shown in Fig. 8.8 with unidirectional force acting in the direction of X.

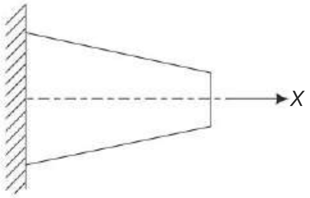


Fig. 8.8 *A one-dimensional body with a force in the X-direction*

The total force T acting on the body consists of three components—the body force f, the traction force T and the point load P. The body force acts on every point of the body, e.g., the body weight. The traction force is that acting on the surface such as friction.

Since the cross-section is non-uniform, in order to discretise, we may convert that into a stepped shaft of different diameters each of which are uniform in size. Each of these sections is then converted into a rectangular shaft maintaining the total area as the same. These then are called finite elements and are numbered with circles as 1, 2, 3 and 4, as shown in Fig. 8.9. The nodes are numbered 1 to 5 and are used as the points of application of the point forces.

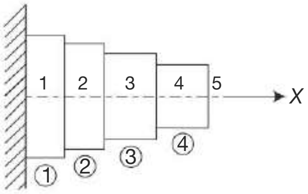


Fig. 8.9 *Equivalent stepped shaft after discretising with no volume change*

8.3.1 Spring Element

Though the structural parts are generally three-dimensional, it is easier first to analyse a one-dimensional approach. It is then possible to extend this to two-dimensional and three-dimensional models. To develop a one-dimensional model, it is easier to consider a spring element. A spring element obeys Hooke's law (deflection linearly proportional to applied force) in one dimension. A structural part is shown in Fig. 8.10 with four spring elements. Each spring element has two nodes at the endpoints of the spring.

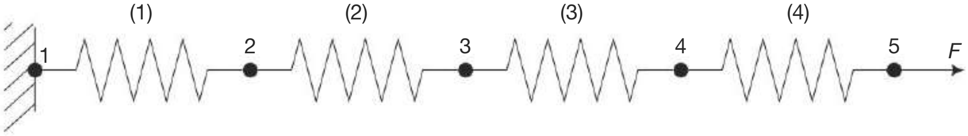


Fig. 8.10 *Four spring elements modelling a structural part shown in Fig 8.9*

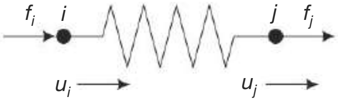


Fig. 8.11 *Isolated spring element from the structural part between nodes i and j*

Isolate a spring between two nodes i and j as shown in Fig. 8.11 with the respective displacements u_i and u_j, and forces f_i and f_j.

Net deflection δ is

$$\delta = u_j - u_i \tag{8.1}$$

Defining the spring constant k as the force per unit deflection, the resultant axial force in the spring is

$$f = k\,\delta = k\,(u_j - u_i) \tag{8.2}$$

For equilibrium,

$$f_i + f_j = 0 \tag{8.3}$$

Hence, it is possible to write the equations for deflection at each of the nodes as

$$k\,(u_j - u_i) = f_i$$

$$k\,(u_i - u_j) = f_j \tag{8.4}$$

Expanding the above equations

$$k\,u_j - k\,u_i = f_i$$

$$k\,u_i - k\,u_j = f_j \tag{8.5}$$

These can be written in matrix form as

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \begin{Bmatrix} -f_i \\ -f_j \end{Bmatrix} \tag{8.6}$$

This can be written more compactly as

$$[K]\,\{U\} = \{f\} \tag{8.7}$$

[K] is called the stiffness matrix of the element under consideration.

8.3.2 Assembling the Stiffness Matrix

If two springs are assembled together, the node numbers, forces acting and the displacements in these two spring elements are shown in Fig. 8.12. Using the results obtained earlier (Eq. 8.6), the combined stiffness matrix can be obtained for this system. Element (spring) 1 has nodes 1 and 2, while the element 2 has nodes 2 and 3, with the node 2 being common between the two elements.

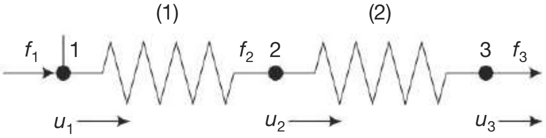


Fig. 8.12 *System of two springs showing various forces and displacements*

Considering that for the equilibrium, the sum of the internal forces should be equal to the external forces applied at each node, it is possible to write the equations for deflection at each of the nodes as

$$\begin{aligned} k_1\,(u_1 - u_2) &= f_1 \\ -\,k_1\,(u_1 - u_2) + k_2\,(u_2 - u_3) &= f_2 \\ -\,k_2\,(u_2 - u_3) &= f_3 \end{aligned} \tag{8.8}$$

These equations can then be written in matrix form as

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -\,k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} \tag{8.9}$$

This can be written more compactly as

$$[K]\,\{U\} = \{f\} \tag{8.10}$$

where the stiffness matrix for two springs [K] is given by

$$[K] = \begin{bmatrix} k_1 & -k_1 & 0 \\ -\,k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \tag{8.11}$$

This can be generalised for the four-element case shown in Fig. 8.10 as

$$\begin{bmatrix} k_1 & -k_1 & 0 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 & 0 \\ 0 & 0 & -k_3 & k_3+k_4 & -k_4 \\ 0 & 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{Bmatrix} \tag{8.12}$$

8.3.3 Boundary Conditions

The stiffness matrix as derived will be singular and, therefore, cannot be solved. In order to solve the above set of equations, it is necessary to specify enough boundary conditions or constraints. Boundary conditions are of two types: homogenous type and non-homogenous type. Each of the nodes in the structural system has six degrees of freedom, three translational along the three axes (x, y and z) of the rectangular coordinate system used and three rotary axes about each of the rectangular axes. In the homogenous boundary conditions, all the degrees of freedom of the node are completely fixed, whereas in the non-homogenous type, some displacements in specified axes will be allowed. The application of the boundary conditions can be explained with the help of an example.

Example 8.1 Consider the assemblage of three springs as shown in Fig. 8.13. Calculate the displacement of the nodal points 2 and 3.

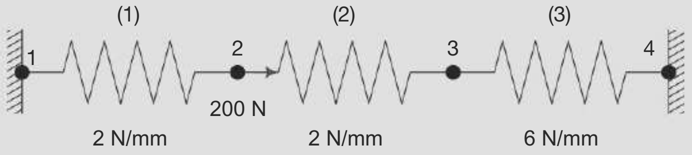


Fig. 8.13 *System of two springs showing various forces and displacements*

Solution The assemblage of the equation for the three-spring system is

$$\begin{bmatrix} 2 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 8 & -6 \\ 0 & 0 & -6 & 6 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}$$

Applying the homogenous boundary conditions as fixed ends will have no displacements, and the internal forces are zero. We get

$$\begin{bmatrix} 2 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 8 & -6 \\ 0 & 0 & -6 & 6 \end{bmatrix} \begin{Bmatrix} 0 \\ u_2 \\ u_3 \\ 0 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ 200 \\ 0 \\ f_4 \end{Bmatrix}$$

Eliminating the columns and rows corresponding to the boundary conditions from the above equation, we get

$$\begin{bmatrix} 4 & -2 \\ -2 & 8 \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} 200 \\ 0 \end{Bmatrix}$$

$$4\,u_2 - 2\,u_3 = 200$$

$$-\,2\,u_2 + 8\,u_3 = 0$$

Solving the above, we get, $u_2 = 57.1428$ mm, and $u_3 = 14.2857$ mm

To obtain the unknown forces,

$$\begin{bmatrix} 2 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 8 & -6 \\ 0 & 0 & -6 & 6 \end{bmatrix} \begin{Bmatrix} 0 \\ 57.143 \\ 14.286 \\ 0 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ 200 \\ 0 \\ f_4 \end{Bmatrix}$$

Solving this we get, $f_1 = -114.286$ N, and $f_4 = -85.716$ N

The equilibrium of the system can be checked by taking

$$f_1 + f_2 + f_3 + f_4 = -114.286 + 200 + 0 - 85.716 = 0$$

8.3.4 Elastic Bar Element

While the spring element provides an easy understanding of the concept of the stiffness matrix, its direct application to solve problems is limited. A more useful formulation would be an elastic linear bar element. Figure 8.14 shows an elastic bar element that obeys Hooke's law. Forces are only applied at the end, and forces are only applied axially. It is known from strength-of-material formulations that for an elastic bar of uniform cross-section, the deflection, δ is given by

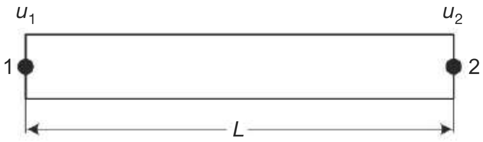


Fig. 8.14 *Elastic bar element*

$$\delta = \frac{PL}{AE} \qquad (8.13)$$

where, P = applied axial load on the bar,

L = length of the bar

A = uniform area of cross-section of the bar

E = Elastic modulus of the bar material

From the above equation, the equivalent spring constant of the elastic bar can be written as

$$k = \frac{P}{\delta} = \frac{AE}{L} \qquad (8.14)$$

Analogous to the spring element discussed earlier, it is now possible to write the stiffness matrix for an assemblage of bar elements. The following example applies the bar element.

Example 8.2 A tapering round bar is fixed at one end and a tensile load of 1000 N is applied at the other end as shown in Fig. 8.15. Take elastic modulus, $E = 2 \times 10^5$ MPa. Find the global stiffness matrix and displacements considering it as 4 elements.

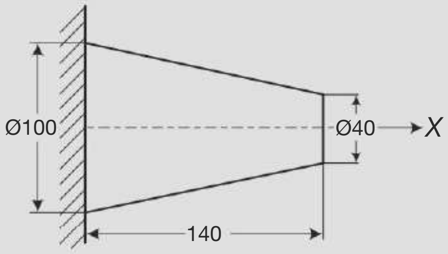


Fig. 8.15 *Tapered round bar*

Solution The tapered bar is divided equally into four elements as shown in Fig. 8.16 and its elemental properties are calculated below. Equivalent spring representation of the beam is shown in Fig. 8.17.

Calculate the areas of all the four elements

$$A_1 = \frac{\pi \times 100^2}{4} = 7853.98 \text{ mm}^2$$

$$A_2 = \frac{\pi \times 80^2}{4} = 5026.55 \text{ mm}^2$$

$$A_3 = \frac{\pi \times 60^2}{4} = 2827.43 \text{ mm}^2$$

$$A_4 = \frac{\pi \times 40^2}{4} = 1256.64 \text{ mm}^2$$

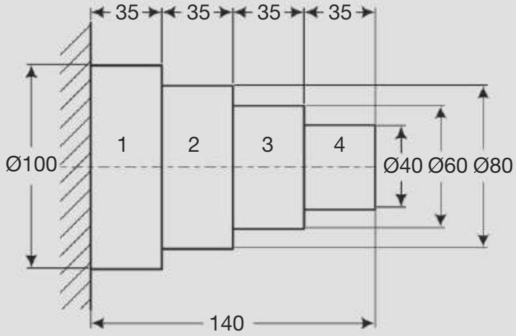


Fig. 8.16 *Tapered rod divided into four elements*

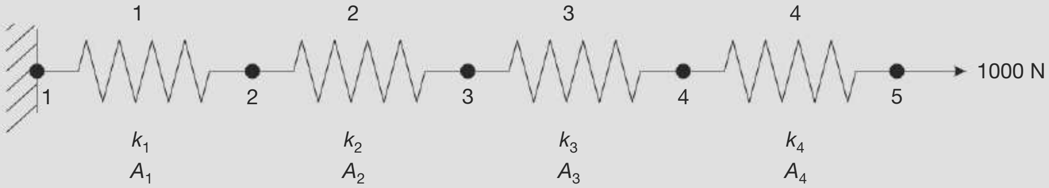


Fig. 8.17 *Equivalent spring formulation of the tapered bar*

Calculate stiffnesses of all the four elements.

Stiffness, $$k = \frac{A \times E}{L}$$

$$k_1 = \frac{A_1 \times E}{L_1} = \frac{7853.98 \times 2 \times 10^5}{35} = 448.8 \times 10^5 \text{ N/mm}$$

$$k_2 = \frac{A_2 \times E}{L_2} = \frac{5026.55 \times 2 \times 10^5}{35} = 287.23 \times 10^5 \text{ N/mm}$$

$$k_3 = \frac{A_3 \times E}{L_3} = \frac{2827.43 \times 2 \times 10^5}{35} = 161.57 \times 10^5 \text{ N/mm}$$

$$k_4 = \frac{A_4 \times E}{L_4} = \frac{1256.64 \times 2 \times 10^5}{35} = 71.808 \times 10^5 \text{ N/mm}$$

Equilibrium equations for each node

$$\begin{bmatrix} k_1 & -k_1 & 0 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 & 0 \\ 0 & -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & 0 & -k_3 & k_3 + k_4 & -k_4 \\ 0 & 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{Bmatrix} = \begin{Bmatrix} -R \\ 0 \\ 0 \\ 0 \\ P \end{Bmatrix}$$

where R is the reaction at the fixed end. Substitute the boundary conditions, and crossing out the first row and column for the fixed condition.

$$\begin{bmatrix} k_1 & -k_1 & 0 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 & 0 \\ 0 & 0 & -k_3 & k_3+k_4 & -k_4 \\ 0 & 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{Bmatrix} 0 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{Bmatrix} = \begin{Bmatrix} R \\ 0 \\ 0 \\ 0 \\ 1000 \end{Bmatrix}$$

From the reduced matrix form the equations as follows:

$$(k_1 + k_2)\, u_2 - k_2\, u_3 = 0$$
$$-k_2\, u_2 + u_3\, (k_2 + k_3) - k_3\, u_4 = 0$$
$$-k_3\, u_3 + u_4\, (k_3 + k_4) - k_4\, u_5 = 0$$
$$k_4\, u_5 - k_4\, u_4 = 1000$$

Substituting the stiffness values as calculated above

$$(448.8 \times 10^5 + 287.23 \times 10^5)\, u_2 - 287.23 \times 10^5\, u_3 = 0$$
$$-\,287.23 \times 10^5\, u_2 + u_3\, (287.23 \times 10^5 + 161.57 \times 10^5) - 161.57 \times 10^5\, u_4 = 0$$
$$-\,161.57 \times 10^5\, u_3 + u_4\, (161.57 \times 10^5 + 71.808 \times 10^5) - 71.808 \times 10^5\, u_5 = 0$$
$$71.808 \times 10^5\, u_5 - 71.808 \times 10^5\, u_4 = 1000$$
$$736.03\, u_2 - 287.23\, u_3 = 0$$

Solving the above simultaneous equations, we get

$$u_1 = 0 \text{ in (boundary condition)}$$
$$u_2 = 2.2272 \times 10^{-5} \text{ mm}$$
$$u_3 = 5.7079 \times 10^{-5} \text{ mm}$$
$$u_4 = 11.896 \times 10^{-5} \text{ mm}$$
$$u_5 = 25.822 \times 10^{-5} \text{ mm}$$
$$R = k_1\, u_2 = 999.573 \text{ N}$$

8.4 TRUSS AND BEAM ANALYSIS

Trusses are the simplest elements and provide an exact solution for the deflections. Truss members are used for analysing the *planar trusses* as well as *space trusses*. A truss element is a straight *bar* of an arbitrary cross-section, which can deform only in its axis direction when it is subjected to axial forces. Truss elements are also termed *bar elements* and described earlier. In planar trusses, there are two components in the x and y directions for the displacement as well as forces at a node. For space trusses, there will be three components in the x, y and z directions for both displacement and forces at a node. In trusses, the truss or bar members are joined together by pins or hinges (not by welding), so that there are only forces (not moments) transmitted between bars. It is assumed that the element has a uniform cross-section.

To develop a relationship between the forces and displacements at each end of a single truss element arbitrarily oriented, consider such an element in the x–y plane as shown in Fig. 8.18, attached to nodes numbered i and j and inclined at an angle θ from the horizontal. The elongation in the truss element can be written in terms of the differences in the displacements of its end points:

$$\delta = (u_j \cos\theta + v_j \sin\theta) - (u_i \cos\theta + v_i \sin\theta) \tag{8.15}$$

where u and v are the horizontal and vertical components of the deflections, respectively. The same can be written in matrix form as

$$\delta = [-c \quad -s \quad c \quad s]\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} \tag{8.16}$$

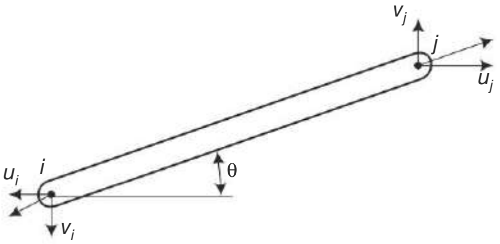


Fig. 8.18 *Truss element*

where $c = \cos\,\theta$ and $s = \sin\,\theta$. The horizontal and vertical nodal forces are shown in Fig. 8.19; these can be written in terms of the total axial force (Eq. 8.13) as

$$\begin{Bmatrix} f_{xi} \\ f_{yi} \\ f_{xj} \\ f_{yj} \end{Bmatrix} = \begin{Bmatrix} -c \\ -s \\ c \\ s \end{Bmatrix} P = \begin{Bmatrix} -c \\ -s \\ c \\ s \end{Bmatrix} \frac{AE}{L}[-c \quad -s \quad c \quad s]\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} \tag{8.17}$$

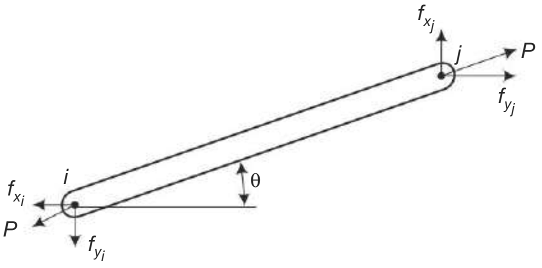


Fig. 8.19 *Truss element with force components*

After carrying the matrix multiplication, we get

$$\begin{Bmatrix} f_{xi} \\ f_{yi} \\ f_{xj} \\ f_{yj} \end{Bmatrix} = \frac{AE}{L}\begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} \tag{`8.18}$$

This is the stiffness matrix for a single element. A truss consisting of a number of elements will then need to be assembled to get the global stiffness matrix. To maintain static equilibrium, the sum of all the elemental forces should be equal to the external forces.

$$f_i^{\text{ext}} = \sum_{elem} f_i^{elem} = \left(\sum_{elem} k_{ij}^{elem}\, u_j\right) = K_{ij}\, u_j \tag{8.19}$$

8.4.1 Analysis of Trusses

To use the truss elements the following guidelines are used:

- It has uniform cross-section throughout its length.
- The length of the element is much greater than the width or depth (approximately 8–10 times).
- It is connected to the rest of the model with hinges that do not transfer moments.
- The external applied forces are only at joints.

The steps in using commercial FEA software to carry out the analysis are the following:

- Build the geometric model selecting a suitable element for the meshing purpose.
- Define the units and type of analysis to be used:
    - Linear static
    - Non-linear static

  - Thermal steady state
  - Fluid flow
- Define the type of element to be used and its geometric parameters.
- Specify the material properties.
- Define the boundary conditions and loads acting at the appropriate nodes.
- Solve and analyse the results.

In the following example, the use of Algor for solving truss problems is given.

Example 8.3 Figure 8.20 shows a steel bridge constructed using all steel members with a cross-sectional area of 3250 mm^2. The maximum load applied by the moving traffic on the bridge is shown in the figure. Analyse the bridge structure. Determine all the nodal displacements and elemental stresses in the structure. What is the effect if the roller end is replaced by a fixed end?

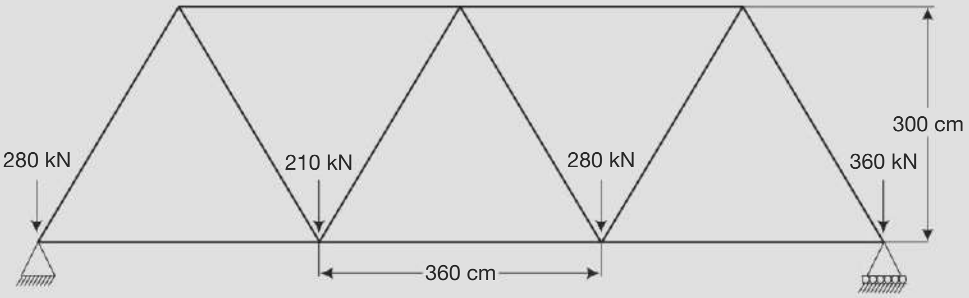


Fig. 8.20 *Example of truss of a bridge*

Algor is a comprehensive finite-element-analysis program with a very easy graphical user interface that is very intuitive. It has linkage to most of the well-known CAD systems as well as other FEA programs, so that the geometry and meshed models could be easily transported between the packages. The opening screen of Algor is shown in Fig. 8.21. Most of the commonly used commands are available in the form of buttons in the top part of the screen while the windows style menu commands are also available. The model tree shows the details of the model that is being built. In the graphic window, the current scales of the geometric model as well as the coordinate system are shown.

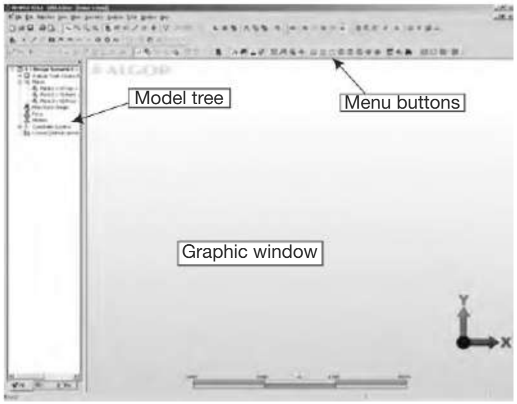


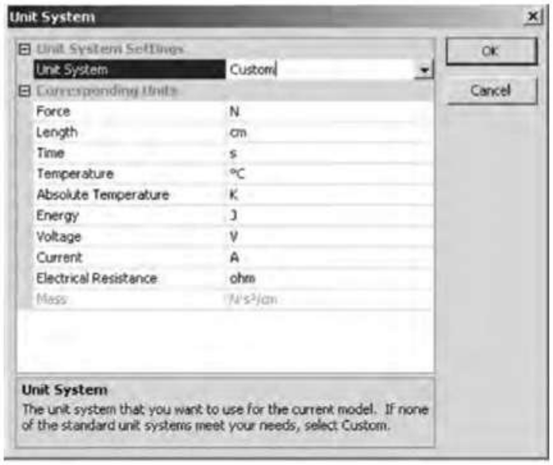
Fig. 8.21 *Opening screen of Algor*

The geometry of the part to be modelled can be either imported from other CAD programs such as Pro Engineer, Inventor, AutoCAD or Solid works, or can be directly entered within Algor. Though it has sufficient geometric capability for building models, the geometric part is not efficient as other CAD programs. Hence, only simple models will have their geometry entered in Algor, while other CAD programs will be more useful. Before entering the geometry, the model units are to be entered. The pop-up window for entering units is shown in

Fig. 8.22. Any type of units can be used for defining as seen in Fig. 8.22.

For simple geometry entry, the user can give the coordinates for endpoints of lines in the case of a truss model. Since a truss element can take loads only at the endpoints, it is sufficient to have nodal points at the end of each element. After the geometry is entered, the user needs to specify the type of element to be used (in this case, truss) after which the cross-sectional area of each of the truss elements is to be specified. Typical pop-up screen for this purpose is shown in Fig. 8.23.

After the geometry entry the material properties are to be entered. Algor has a comprehensive library of materials with all their properties as required for the analysis. The user has to simply select the material from the library as shown in Fig. 8.24. In case the material is not available in the library, the properties can be entered as custom material. This completes the description of the geometric part and the material properties.



Unit System Settings

Unit System	Custom

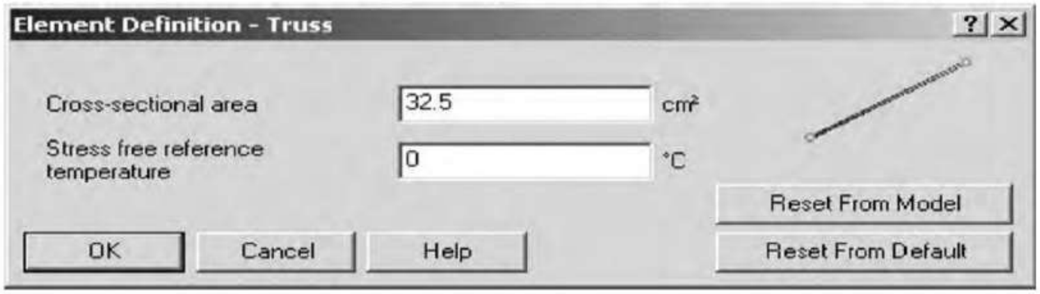
Corresponding Units

Force	N
Length	cm
Time	s
Temperature	°C
Absolute Temperature	K
Energy	J
Voltage	V
Current	A
Electrical Resistance	ohm
Mass	N*s²/cm

Unit System

The unit system that you want to use for the current model. If none of the standard unit systems meet your needs, select Custom.

Fig. 8.22 *Specifying units in Algor*



Cross-sectional area: 32.5 cm²

Stress free reference temperature: 0 °C

Fig. 8.23 *Specifying the geometric properties of the truss element in Algor*

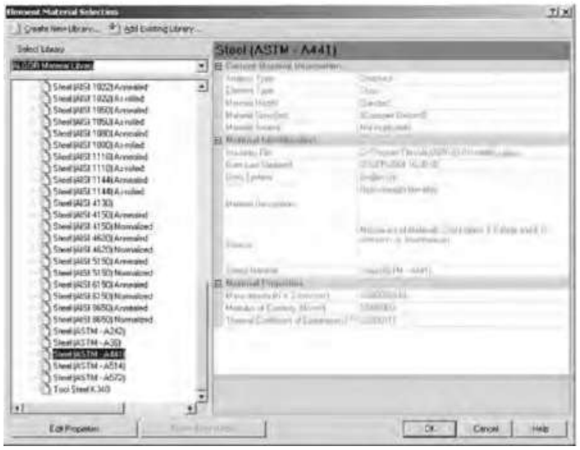
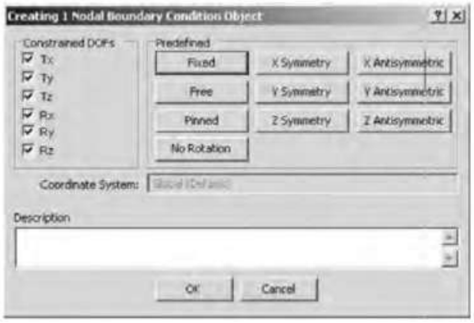


Fig. 8.24 *Material specification from library in Algor*



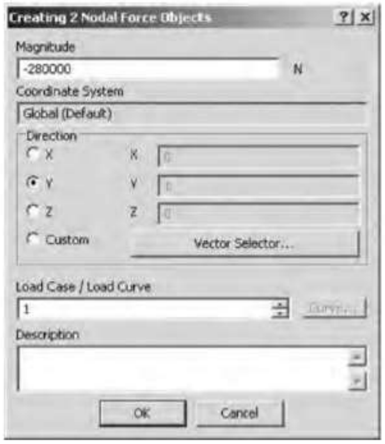
Constrained DOFs

☑ Tx
☑ Ty
☑ Tz
☑ Rx
☑ Ry
☑ Rz

Coordinate System: Global (Default)

Fig. 8.25 *Specifying boundary conditions in Algor*

Next is to enter the boundary conditions as shown in Fig. 8.25. Here the left end is given as fixed, thus all the six degrees of freedom are fixed. As the right end is a roller contact, the translation along the *X*-axis will be released. After this, the acting forces at the various nodes as specified in the problem need to be entered through the pop-up window shown in Fig. 8.26. Care has to be exercised with reference to the units being entered in this window.


Creating 2 Nodal Force Objects

Magnitude: -280000 N

Coordinate System: Global (Default)

Direction: ○ X ● Y ○ Z ○ Custom

X: 0

Y: 0

Z: 0

Load Case / Load Curve: 1

Description:

Fig. 8.26 *Specifying the acting force in Algor*

The model with all the boundary conditions and loading is shown in Fig. 8.27. It is then analysed. Various results for the analysis can be displayed such as displacement, axial stress or axial forces in the elements. An example is given in Fig. 8.28 for axial stress in the various elements. In all these figures. the magnitude of the result is seen as a continuous colour with the colour legend shown at the top right-hand corner of the figure. It is also possible to read the various values of the results from this figure. For this purpose, first select all the nodes from the model in Fig. 8.29 and then inquire the results. Complete results from the analysis are given in Table 8.1. The element and node numbers are identified in Fig. 8.30.

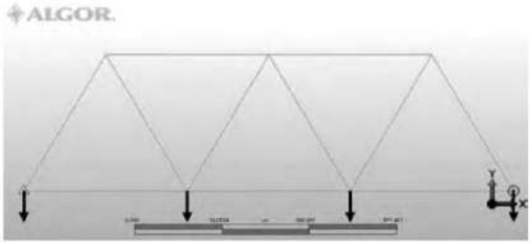


Fig. 8.27 *Truss model with the boundary conditions and loading*

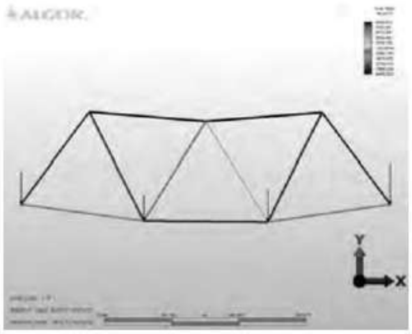


Fig. 8.28 *Displaced model of truss with boundary conditions shown*

Fig. 8.29 *Display of results at various nodes in the model*

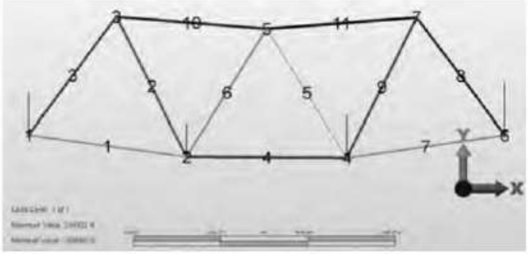


Fig. 8.30 *Element and node numbers identified in the model*

Table 8.1 *Complete results from the truss analysis for Example 8.3*

Node	*One fixed and other roller ends, Displacement, mm*	*Both ends fixed Displacement, mm*
1	0	0
2	6.843	5.504
3	4.844	3.088
4	7.608	5.919
5	7.677	5.868
6	3.257	0
7	3.865	3.341

Element	*Axial Force, kN*	*Axial Stress, MPa*
1	140.000	43.07
2	272.111	83.72
3	–272.111	–83.72
4	294.000	90.46
5	27.211	8.37
6	–27.211	–8.37
7	154.000	47.38
8	–299.322	–92.09
9	299.322	92.09
10	–280.000	–86.15
11	–308.000	–94.76

8.5 BEAM ANALYSIS

A beam is considered as a long and slender structural member subjected to transverse loading and experiences significant bending effects. Beams are generally found in building frames, transmission towers and bridges. A beam differs from a truss since it resists moments (twisting and bending) at the connections.

A beam element is used under the following conditions:

- The length of the element is much greater than the width or depth.
- The element has constant cross-section throughout its length.
- The element must be able to transfer moments.
- The element must be able to handle a load distributed across its length.

In Algor three orthogonal forces (one axial and two shear) and three orthogonal moments (one torsion and two bending) are calculated at the end points of all elements.

Generally, strength-of-materials procedures are easy to work with simple beam configurations. However, when cross-sectional changes occur or different loading conditions are present then the analysis becomes difficult and FEA becomes necessary. Cross-sectional properties are used to define the stiffness in all the degrees of freedom of beam elements. Different types of loads specified in beams are concentrated loads, moments, distributed loads uniform along the entire length and linearly variable along the length. When meshing the geometry, nodes should be placed according to the requirement. Nodes should be provided at all supports, where concentrated loads are present and where results are required. This is required because FEA calculates the values only at the nodal points.

An example is taken here to understand the procedure of beam analysis below:

Example 8.4 Figure 8.31 shows a common street-light arrangement with the light fixture weighing 70 N. Using the Algor FEM software, analyse the street-light structure. Examine the nodal displacements, and deformed shape of the structure. If the rod *AC* is removed from the structure, what happens to the displacement?

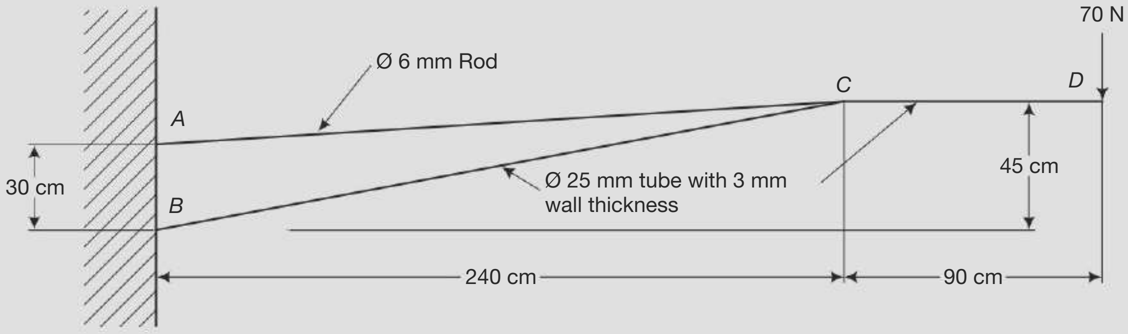


Fig. 8.31 *Example of a beam model for lamp post*

Taking this as a simple cantilever, with a 25-mm rod that is 330 cm long, with a concentrated load of 70 N acting at the free end, calculate the maximum deflection to get an idea of the expected result.

Solution Maximum deflection, $\delta = \frac{FL^3}{3EI} = \frac{70 \times 3300^3 \times 64}{3 \times 2 \times 10^5 \times \pi \times 25^4} = 218.65$ mm

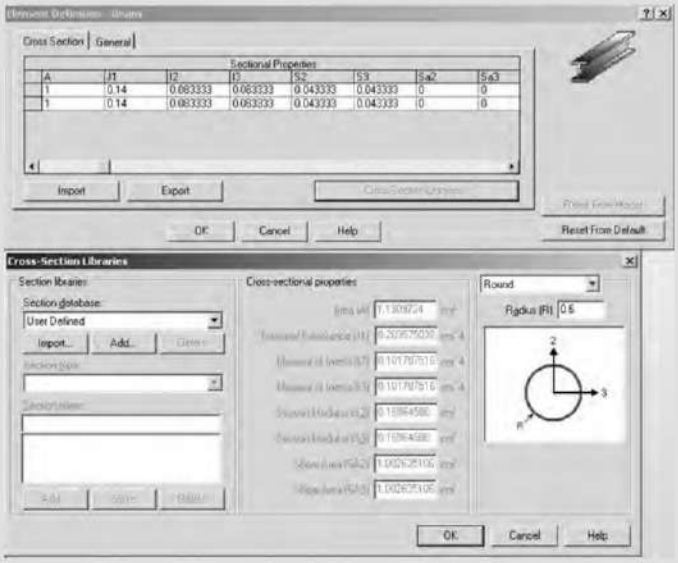
Similar to the truss, the geometry can be entered by specifying the coordinates for endpoints of lines. However, care has to be taken to add more nodes by dividing the lines to let Algor calculate the displacements where required. The cross-sectional area of each of the truss elements is to be specified. A typical pop-up screen for this purpose is shown in Fig. 8.32. The beams can have a wide variety of cross-sections possible, and the structural rigidity is affected by the geometry. Hence, Algor provides a wide variety of cross-section libraries of standard shapes and custom shape geometries that can be defined by the user. The required geometric properties such as area, moment of inertia, section modulus are all automatically calculated from the geometric data entered. The finite element model with all the constraints and loading is shown in Fig. 8.33.

The results of the analysis are shown in Fig. 8.34 with the node numbers and the actual displacement values are shown in Table 8.2.

Table 8.2 *Complete results from the beam analysis for Example 8.4*

Node	*Original geometry displacement, mm*	*With the rod removed displacement, mm*	*Shear force, N*	*Bending moment N mm*
1	–3.354	–200.698	0	0
2	–6.804	–226.357	0	0
3	–10.893	–252.654	0	0
4	–15.461	–279.431	0	0
5	–20.349	–306.528	0	0
6, Free end	–25.396	–333.784	70	0
7, Support	0	0	–123.95	26937.36
8, Support	0	–	53.95	136.05

It can be seen from Table 8.2 the effect of the rod is in reducing the deflection of the light fixture. Also, it can be seen that with the rod removed the displacement value is close to the analytical result calculated earlier (218.65 mm).



Cross Section | General

Sectional Properties							
A	J1	I2	I3	S2	S3	Sa2	Sa3
1	0.14	0.083333	0.083333	0.043333	0.043333	0	0
1	0.14	0.083333	0.083333	0.043333	0.043333	0	0

Cross-Section Libraries

Section database: User Defined

Area (A) 1.1309724

Round

Radius (R) 0.6

Fig. 8.32 *Pop-up window for entering beam cross-section data*

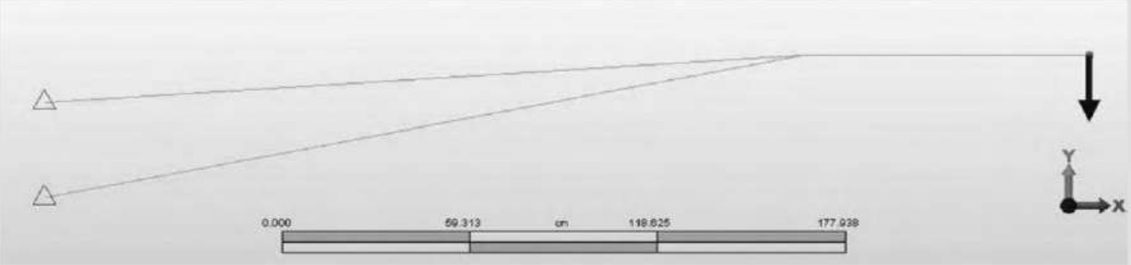


Fig. 8.33 *Example of a beam model for lamp post*

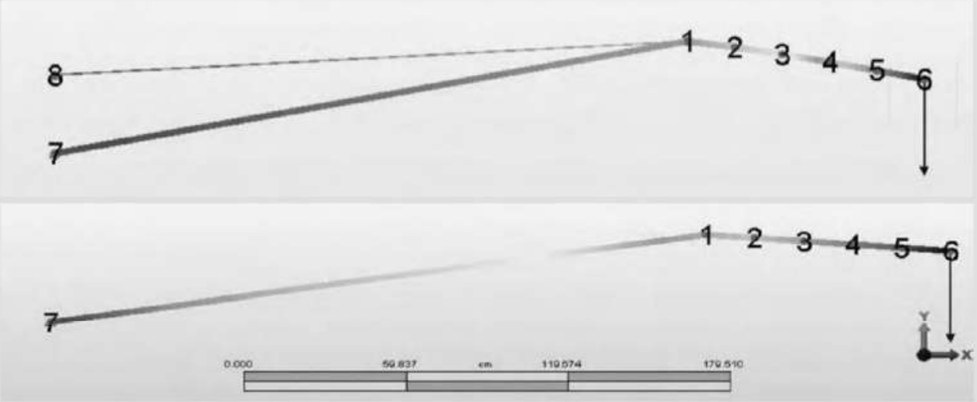


Fig. 8.34 *Beam model with node numbers for the two cases in Example 8.4*

8.6 PLANE STRESS/STRAIN ANALYSIS

These are the simplest type of problems.

- Plane stress is defined to be a state of stress in which the normal stress and the shear stresses directed perpendicular to the plane are assumed to be zero.
- Plane strain is defined to be a state of strain in which the strain normal to the plane and the shear strains are assumed to be zero.

Algor utilises the *YZ* plane for all the two-dimensional problems of plane stress, plane strain and axi-symmetric elements. Plane stress elements are used to analyse parts that are thin (having a thickness that is less than the in-plane *YZ* dimension) and the applied load act only in the *YZ* plane. A 2D solid element can have a triangular, rectangular or quadrilateral shape with straight or curved edges. All loading is expected to be in the plane. At any point, there are two components for the displacement as well as forces in 2D elements. 2D elements, by definition, cannot have rotational degrees of freedom (DOFs) or translation in the *X* direction. A few precautions to be noted while meshing the plane stress/plane strain problems are

- All elements must be bounded by line segments
- Arcs and splines must be broken into line segments
- Adjacent elements must share a single line
- Elements should only be quadrilateral or triangular with good aspect ratio and a small amount of skewness

An example has been worked out to demonstrate the procedure.

Example 8.5 Determine the safe load that can be applied to the plate (AISI 1020 steel, as rolled) shown in Fig. 8.35 without causing the tensile stress at any point to exceed 100 MPa using Algor FEM software. In building the Algor model, utilise the symmetry of the part geometry. Calculate the stress from the analytical procedure using the stress-concentration factors given in handbooks.

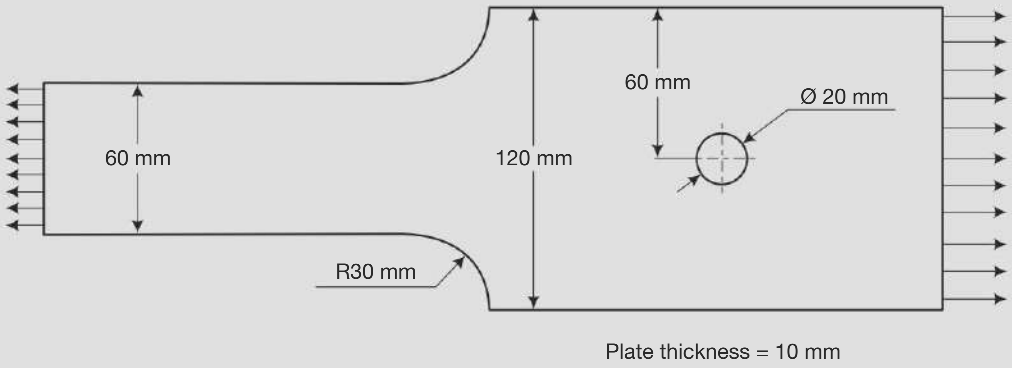


Fig. 8.35 *Part for Example 8.5*

In view of the symmetry present in the part, we can take the symmetrical half of the part and in view of that the shape of the part along with the boundary conditions for the two sides of the part are shown in Fig. 8.36. The boundary conditions for the symmetrical part will be similar to roller ends (translation in only direction allowed while all other degrees of freedom are fixed) as shown in Fig. 8.36.

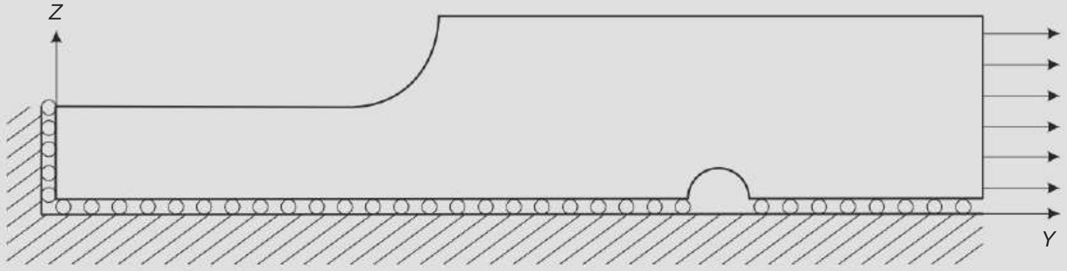
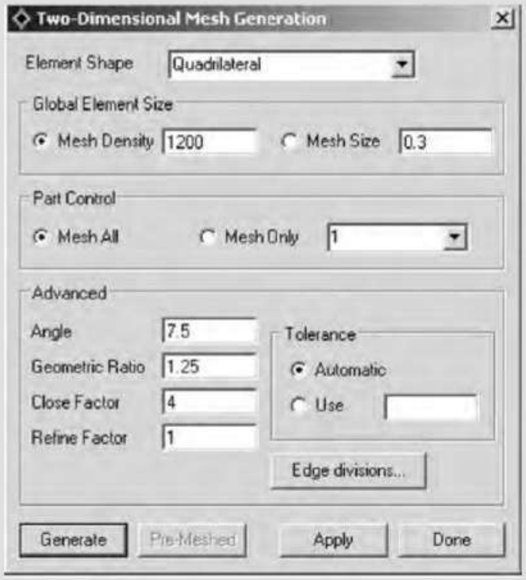


Fig. 8.36 *Symmetrical half of the part for Example 8.5*

The 2D geometry of the plate is modelled in a CAD package and imported as a wireframe model into Algor and meshed. The meshing pop-up window is shown in Fig. 8.37. The choice of 1200 quadrilateral elements was made for part with automatic meshing. The generated mesh can be seen in Fig. 8.38. As can be seen, the automatic mesh generation program is able to generate finer mesh near the hole surfaces in view of the possibility of high stress gradient.



Two-Dimensional Mesh Generation

Element Shape: Quadrilateral

Global Element Size
● Mesh Density 1200
○ Mesh Size 0.3

Part Control
● Mesh All
○ Mesh Only 1

Advanced
Angle 7.5
Geometric Ratio 1.25
Close Factor 4
Refine Factor 1

Tolerance
● Automatic
○ Use

Fig. 8.37 *Pop-up window for generating 2D mesh from wireframe geometry model*

In order to estimate the safe load to be applied on the plate, we may start with any arbitrary load on the plate and find out the maximum stress in the plate. Then the safe load can be calculated by extending it to 100 MPa. It can be seen from Fig. 8.38 that there are 15 nodes on the face where the tensile force is to be applied. Assuming each node has a force of 100 N with the extreme nodes being 50 N for the sake of uniform application of load, we have a total of 1400 N applied. Next the boundary conditions and loading are applied as required, and the analysis carried out. The result of the analysis is shown in Fig. 8.39.

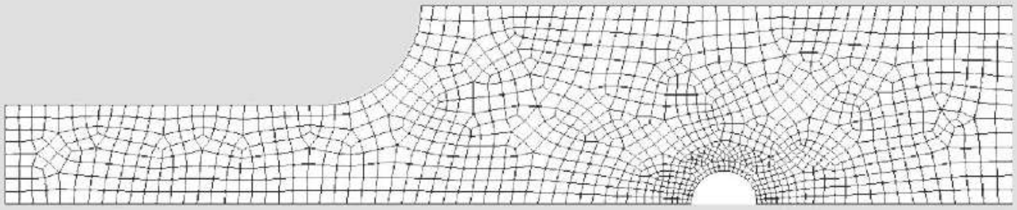


Fig. 8.38 *Meshed model for Example 8.4*

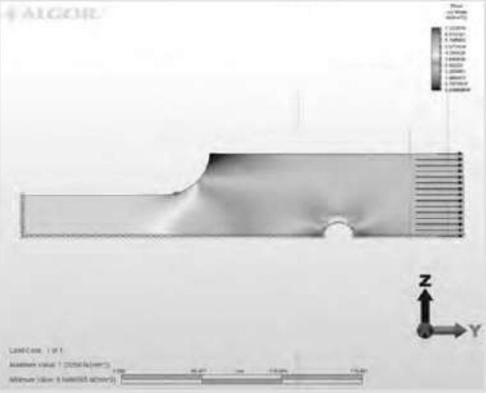


Fig. 8.39 *von Mises stress contours in the plate for Example 8.5*

The maximum stress in the plate for 1400 N load = 7.23258 MPa

Hence, load for 100 MPa stress = $\frac{1400 \times 100}{7.23258}$ = 19356.853 N

Since we have taken only half the plate for analysis, the total load required to get a maximum of 100 MPa in the plate = 19356.853 × 2 = 38714 N.

The analytical solution utilising stress-concentration factor for the above case can be calculated as follows:

Stress concentration factor for the above case, $K = 2.5$ at the circular hole, where the maximum stress is present.

$$\text{Maximum stress, } \sigma_{\max} = \frac{KP}{\text{Area}} = \frac{2.5 \times P}{100 \times 10} \leq 100 \text{ MPa}$$

Re-arranging the above equation, we get

$$P \leq \frac{100 \times 100 \times 10}{2.5} \leq 40000 \text{ N}$$

The error in the value calculated from the FEA is about 3.2%. To get a better approximation, the mesh may have to be refined further.

Summary

- Finite element analysis as a procedure is now well established and is used to a large scale in the manufacturing industries.
- Finite element analysis can be used for practically all types of analysis problems.
- Finite element model discretises the solid body into small and finite volumes called finite elements, where the elasticity principles can be easily applied.
- Finite element analysis procedure requires a preprocessing operation that converts the CAD model into discretised form called the mesh, the solver to evaluate the various equations formed from the mesh and postprocessing module that allows for the interpretation of the results obtained from the solver.
- A variety of elements are available to cater to the type of analysis as well as the geometries encountered in the analysis.
- Postprocessing of the results is an important part, since it helps in clearly displaying the information in a way that meaningful conclusions can be drawn about the design.

Questions

1. Explain what you understand by the finite element method.
2. Define the terms: nodal point, element and degrees of freedom.
3. Explain why the computer is necessary in the use of the finite element method.
4. What advantage would a company derive by performing a finite element analysis of an existing part, which can be strain gauged and tested in a lab?
5. What are the various types of analyses that are possible using FEA?
6. What are the conditions under which linear static analysis is carried?
7. Briefly explain the steps to be followed in manually carrying out the finite element solution to a physical problem.
8. What are the steps to be carried out for solving a physical problem with the help of FEM software?
9. Explain the functions served by a preprocessor in FEM.
10. What are the guidelines for using a truss element?
11. What are the guidelines for using a beam element?
12. What are the assumptions made while using a beam element?
13. When will you use a plane stress (2D) elements?
14. Explain the functions served by a postprocessor in FEM.
15. What different methods are available for preprocessing when using a CAD system in conjunction with FEM software?

Problems

1. Figure 8.40 shows a truss constructed using all steel members with a cross-sectional area of 5000 mm^2. The maximum load applied on the truss is shown in the figure. Using the Algor FEM software, analyse the truss. Determine all the nodal displacements and elemental stresses in the structure. Based on the results, discuss the reduction in the displacement of the roller end by replacing with a fixed support.

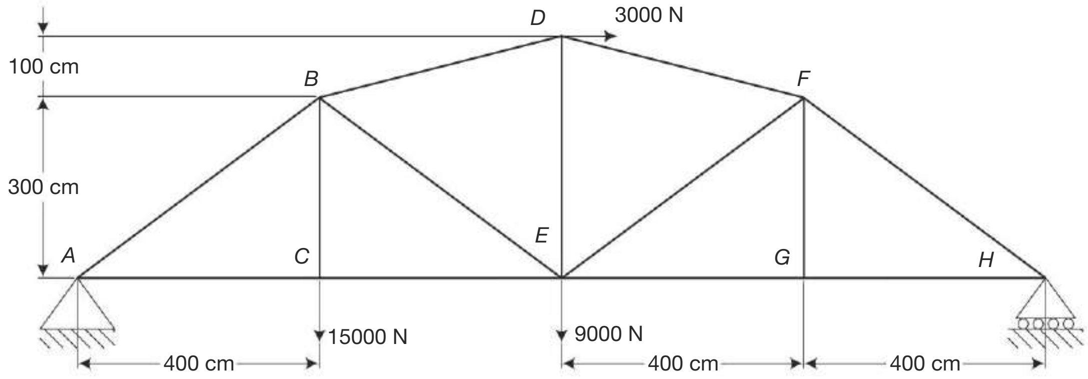


Fig. 8.40 *Truss for Problem 1*

2. Figure 8.41 shows a steel frame to support a load of 9 kN. Using the Algor FEM software, analyse the frame. All the members of the frame are hollow circular cross-sections as shown. The frame is made of Steel ASTM A36. Selecting the thickness of the tube, determine all the nodal displacements and elemental stresses in the structure.
3. Determine the static load that may be applied to the 10-mm thick plate (AISI 1020 steel, as rolled) shown in Fig. 8.42 without causing the tensile stress at any point to exceed 160 MPa, using Algor FEM software. In building the Algor model, utilise the symmetry of the part geometry. Calculate the stress from the analytical procedure using the stress-concentration factors given in handbooks. Use at least 3 different mesh sizes (coarse to fine, e.g., 400, 800, and 1600 nodes) in Algor and show that the results have converged. Based on the results, discuss the deviation from the analytical result, mesh type and the effect of mesh fineness to that of the accuracy.

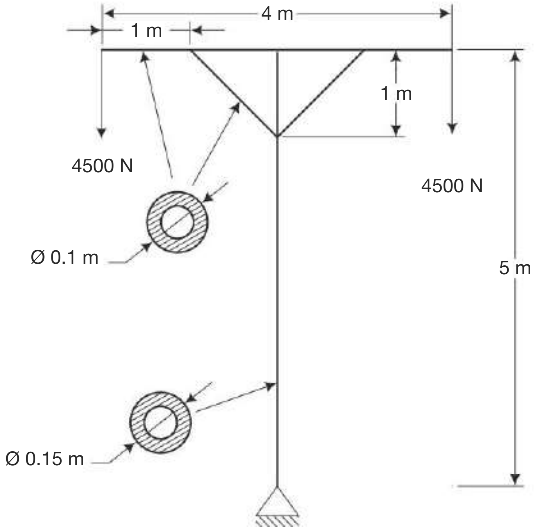


Fig. 8.41 *Beam for Problem 2*

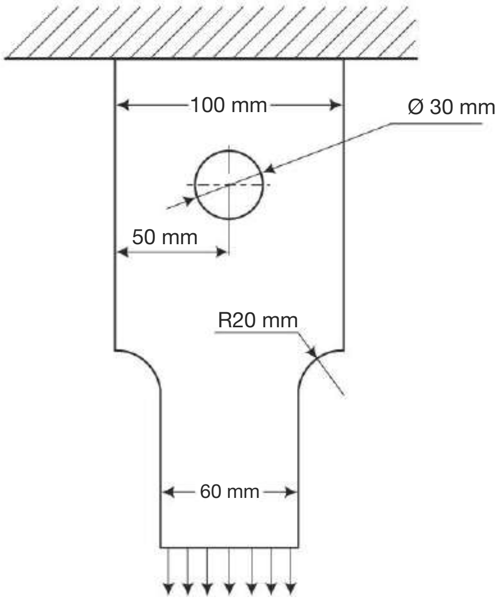


Fig. 8.42 *Plate for Problem 3*

Part – III

MANUFACTURING ASPECTS OF INDUSTRIAL PRODUCTS

9

INTRODUCTION TO COMPUTER NUMERICAL CONTROL

Objectives

Computer numerical control has revolutionised the manufacturing automation during the last 40 years. The movement of the tool and the operation of the various components of a machine tool through a program brings in the kind of flexibility to manufacturing which is not possible by means of hard automation. After completing the study of this chapter, the reader should be able to

- Learn the historical development of numerical control technology
- Understand the principle of numerical control
- Explore the various modes of operation of numerical control
- Know about the elements that are required for the successful operation of numerical control
- Discuss the application of numerical control in the industry
- Understand the advantages and disadvantages of adopting numerical control
- Look at a few of the practical NC machines

9.1 INTRODUCTION

Competition between manufacturing firms is increasingly dictated by quality, cost, variety and servicing. Each one of these attributes of a successful product can only be produced by achieving the highest possible efficiency in manufacturing.

The variety being demanded in view of the varying tastes of the consumer calls for very small batch sizes. Small batch sizes are not able to take advantage of mass-production techniques such as special-purpose machines or transfer lines. Hence, the need for flexible automation where you get the benefits of rigid automation but also be able to vary the products manufactured thus bringing in the flexibility. Numerical control fits the bill perfectly, and we can predict that future manufacturing would increasingly be dependent on Numerical Control or NC to be short.

Numerical Control (NC), or control by numbers, is the concept which has revolutionised the manufacturing scene that is partially due to the rapid advancement in microelectronics that has taken place since late 1960's. The key factor responsible for the popularity of numerical control is the *flexibility* it offers in manufacturing.

Towards the end of the Second World War, there was increased activity in aerospace manufacturing in USA. John Parsons of Parsons Corporation, who was one of the subcontractors to USAF (United States Air Force), was toying with the idea of utilising the digital computers, to reduce the drudgery of computation, which were just then becoming popular. Machining (milling) of complex curvature is a highly skilled job. He proposed that the coordinate points of a complex three-dimensional profile may be utilised for controlling the milling-machine table so that accurate jobs could be produced. The USAF accepted his proposal and a contract was awarded to him to develop such a machine. The project was then awarded to the Servomechanism Laboratory of Massachusetts Institute of Technology in 1951, who finally demonstrated a working milling machine in 1952.

This was a 28-in., Cincinnati Hyrdro-Tel vertical-spindle contour milling machine which was extensively modified. All the motion elements in the milling machine were removed and were replaced by three variable-speed hydraulic transmissions and connected to the three lead screws of the table. The resolution of the machine was 0.0005-in. A feedback control system involving synchros was provided to make sure the machine was moving to the correct positions as programmed.

The first control system was developed using electronic valves. Bendix Corporation produced the first commercial production-based NC unit in 1954 after purchasing the patent rights from MIT. In 1960, the first controller with transistor technology was introduced. These systems were able to control machines with three, four, and five axes and had new features such as circular and parabolic interpolation, cutter compensation and dial input. Integrated circuits (ICs) appeared on the scene in 1967. These permitted a 90% reduction in the number of components, as well as an 80% reduction in wiring. These systems were much more reliable.

Though the concept was demonstrated, the actual availability of such a machine for the aerospace industry was around 1955 after a very large number of refinements to the basic controller demonstrated in 1952. Later on, machine-tool builders serving a variety of applications introduced several commercial NC units into the market. Since then, rapid strides have taken place in NC technology parallel with the developments in electronics and microelectronics.

9.2 NUMERICAL CONTROL

Numerical control of machine tools may be defined as a method of automation in which various functions of machine tools are controlled by letters, numbers and symbols. Basically, an NC machine runs on a program fed to it. The program consists of precise instructions about the methodology of manufacture as well as the movements, for example, what tool to be used, at what speed, at what feed and to move from which point to which point in what path. Since the program is the controlling point for product manufacture, the machine becomes versatile and can be used for any part. All the functions of an NC machine tool are therefore controlled electronically, hydraulically or pneumatically.

In NC machine tools, one or more of the following functions may be automatic:

(a) starting and stopping of machine-tool spindle
(b) controlling the spindle speed
(c) positioning the tool tip at desired locations and guiding it along desired paths by automatic control of the motion of slides

(d) controlling the rate of movement of the tool tip (i.e., feed rate)

(e) changing of tools in the spindle

The principle of operation of an NC machine tool is shown in Fig. 9.1. The basic information that has to be input into the system consists of the part geometry, cutting-process parameters followed by the cutting tools used. This part program is then entered into the controller of the machine, which in turn runs the machine tool to make the part. Each of the machine axes is connected to a servomotor which works under the control of the Machine Control Unit (MCU) as shown in Fig. 9.2. The movement of the cutting tool with respect to the workpiece is given in terms of the coordinates, which are used to control the motion of the servomotor which drives the individual axes.

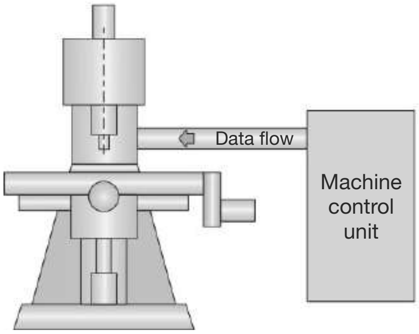


Fig. 9.1 *Principle of operation of an NC machine tool*

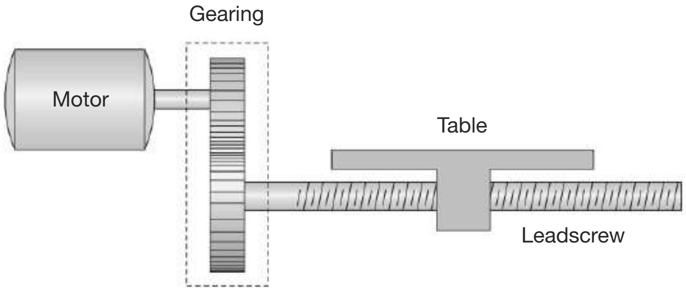


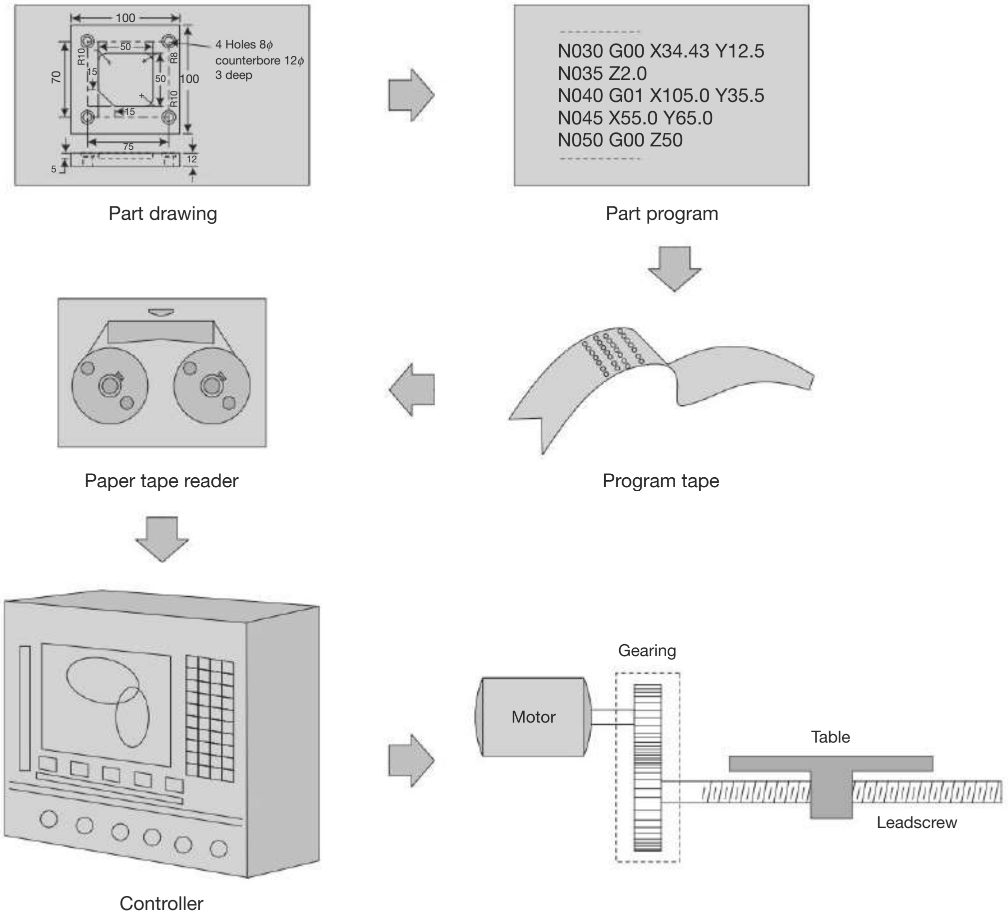
Fig. 9.2 *Principle of operation of the control of axis motion in an NC machine tool*

The general structure of the operation of a typical numerical control system is shown in Fig. 9.3. The part program consists of instructions written in the numerical codes that constitute the basic operations to be carried out in machining of the part. These instructions are then entered into an input medium such as a standard 1-inch paper tape. The program is then read by the paper-tape reader. The controller translates these numerical codes into the machine actuation details, which are then used to control the individual machine functions such as the movement of the axes.

The system shown in Fig. 9.2 is working in an open-loop control, where the checking for the actual position reached is not carried out. Most of the NC machine tools are controlled with a feedback control system wherein the feedback information is provided to the machine control unit as shown in Fig. 9.4 to ensure that the programmed instructions are accurately carried out. The feedback provided consists of the positional as well as velocity data. The feedback for the actual motion achieved is obtained generally with the help of an encoder as shown in Fig. 9.5.

The axis-motion control system operates in a feedback loop with suitable transducers such as linear scales and/or rotary encoders to get the appropriate position or velocity feedback as shown in Fig. 9.5. Most of these systems have a very high response with good resolution of the order of 1 μm (micron) or less.

The NC machines developed in the early days had the total control system developed using the hardware. So the control system is actually implemented as hardware logic using a variety of SSI and MSI integrated circuits. This is sometimes called *hardwired numerical control*. These are characterised by a part program input media such as magnetic or paper tape. Typically, these had very little part program memory, often only a single block.



```
N030 G00 X34.43 Y12.5
N035 Z2.0
N040 G01 X105.0 Y35.5
N045 X55.0 Y65.0
N050 G00 Z50
```

Fig. 9.3 *Elements of NC machine-tool operation*

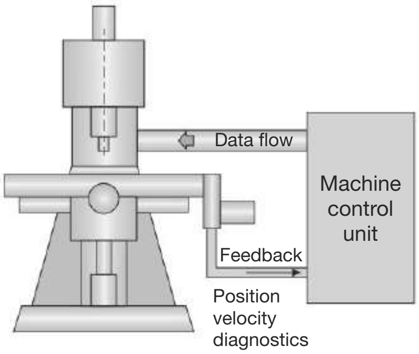


Fig. 9.4 *The data processing in a CNC machine tool in closed-loop control*

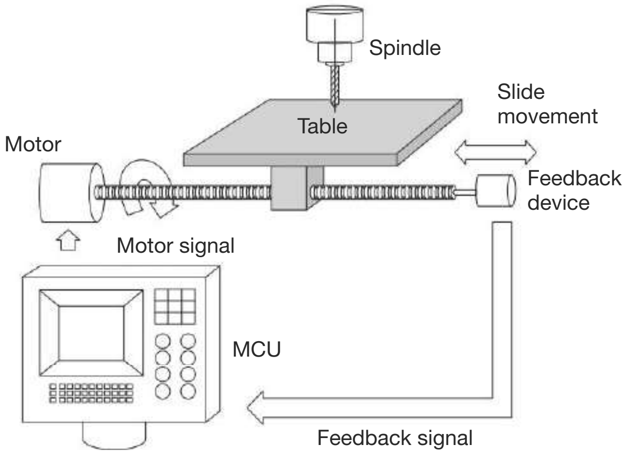


Fig. 9.5 *The data processing in a CNC machine tool in closed-loop control*

With the availability of microprocessors in the mid-70s, controller technology has made a tremendous progress. The new control systems are termed as Computer Numerical Control (CNC) which are characterised by the availability of an embedded computer and enhanced memory in the controller as shown in Fig. 9.6. These also may be termed *softwired numerical control*.

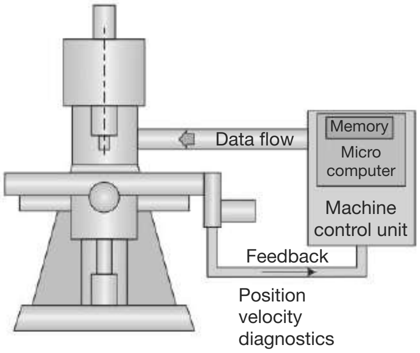


Fig. 9.6 *Typical CNC machine-tool operation*

There are many advantages to be derived from the use of CNC compared to NC. Some of them are

- Part-program storage memory
- Part-program editing
- Part-program downloading and uploading
- Part-program simulation using tool path
- Tool-offset data and tool-life management
- Additional part-programming facilities
- Macros and subroutines
- Background tape preparation
- Drip feeding of part programs for large-sized part programs
- Local storage such as attached hard disks
- Additional support software for diagnostics and maintenance
- Using standard operating systems such as Windows 95 for easier interfacing with other components of manufacturing systems

In reality, the controls with the machine tools nowadays are all CNC and the old NC control do not exist anymore. As a result, the terms NC and CNC become almost synonymous.

9.3 NUMERICAL CONTROL MODES

The controllers have a number of modes in which to operate. There could be 4 possible modes in which the controller can function as shown in Fig. 9.7 in relation to a machining centre. The first shows a typical drilling machine operation, termed *point-to-point mode*. In this, the control has the capability to operate all the 3 axes, but not necessarily simultaneously. As a result, it would be possible to move the tool to any point (in X and Y-axes) in the fastest possible speed and carry out the machining operation in one axis (Z-axis) at that point. This is useful for drilling and punching machines. The second type is an improvement over this in which in addition to the point-to-point mode, the machine tool has the capability to carry out a continuous motion in each of the axis direction. This helps in obtaining the milling in a straight line along any of the axes.

In the third type is shown a control system, which improves the previous type by adding the simultaneous motion capability in any 2 axes. This is what is required in most of the cases. Any 3D profiles to be machined can be completed using the concept of 2.5D mode, in view of the limitation of the machine.

The last one is the highest form of control that is generally found in most of the current-day control systems. This gives the capability of simultaneous 3 or more axes motion. This would be useful for machining most of the complex 3D profiles encountered in industrial practice such as aerospace components, moulds and dies.

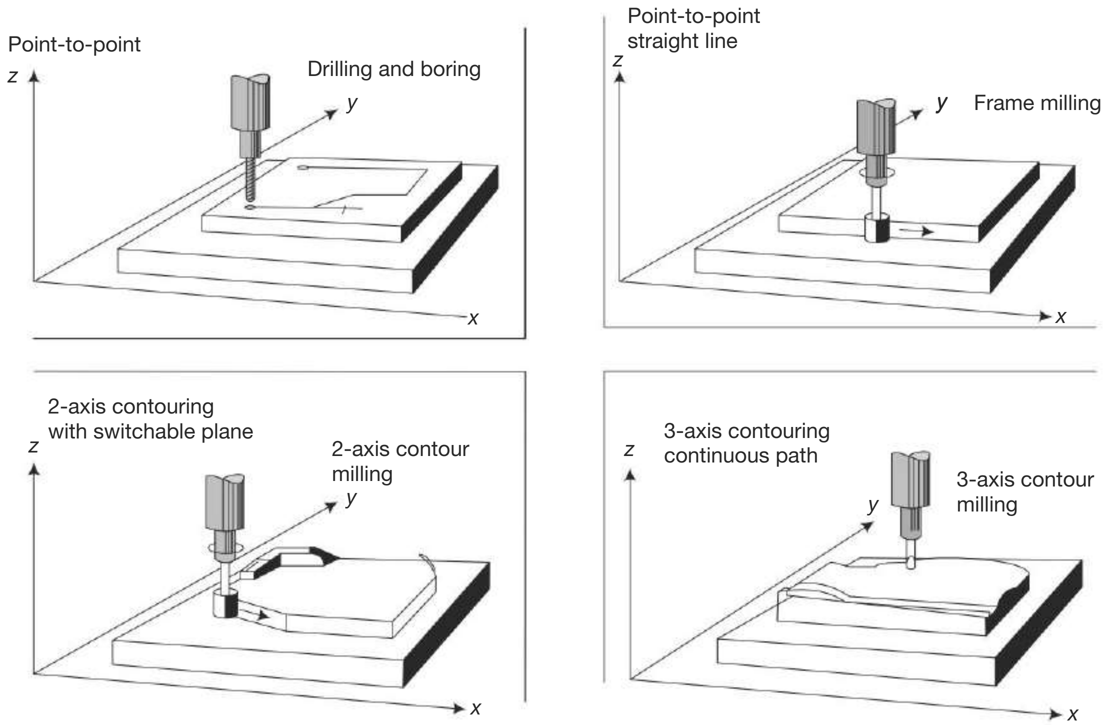


Fig. 9.7 *Types of control systems possible in CNC operation*

9.4 NUMERICAL CONTROL ELEMENTS

9.4.1 Machine Control Unit

Every NC machine tool is fitted with a Machine Control Unit (MCU) which performs the various controlling functions under the program control. The MCU may be generally housed in a separate cabinet-like body or may be mounted on the machine itself. When separately mounted, it may sometimes be like a pendant which could swing around for convenient handling by the operator. Appearance-wise it looks like a computer with a display panel generally of small size (9 inches), and a number of buttons to control the machine tool along with a keyboard. This control unit controls the motion of the cutting tool, spindle speeds, feed rate, tool changes, cutting-fluid application and several other functions of the machine tool. A typical machine control unit is shown in Fig. 9.8.

9.4.2 Part Program

Part program is a very important software element in the NC manufacturing system. It is a detailed plan of manufacturing instructions required for machining the part as per the drawing. It is similar to a computer program containing a number of lines/statements/instructions (called NC blocks) following a specified format. The format is standardised by ISO which is followed by many a controller manufacturers with minor variations. Some typical NC blocks written in the word address format as per ISO are shown below:

```
N30 G00 X120.0 Y 45.0 Z-85.0
N40 G90
N50 G03 X200.0 Y200.0 I-100.0 J0 F200
N60 G01 X120.0 Y110.0
```

The program can also be written in higher-level languages such as APT, UNIAPT, COMPACT II, etc. These programs have to be converted into the earlier mentioned machine-tool-level program with the help of processors and post-processors. It is similar to the practice by which computer programs written in high-level languages such as FORTRAN are converted into the relevant computer machine language with the aid of a suitable compiler. This is termed as computer-aided part programming and discussed later.

The programs can also be developed directly using CAD/CAM systems such as Unigraphics, Pro Engineer, Euclid, and SDRC I-DEAS or CAM systems such as MasterCAM, SmartCAM, SurfCAM, Duct, etc. These also require a post-processor like the earlier discussed computer-aided part-programming systems.

Fig. 9.8 *Machine tool control unit (Heidenhain TNC410) for machining centres*

9.4.3 Program-Coding Systems

The human readable characters as seen in the manuscript of the part program cannot be directly entered into the machine control unit. They need to be converted into some form of code and entered. The numbering system that we use for normal algebra function utilises a base-10 decimal system. However, computers utilise a binary system, which is of base 2. A binary system has only two states, 0 and 1. Sometimes other forms such as octal (base 8) and hexadecimal (base 16) systems are widely used in computer applications.

It is a very straightforward manipulation to convert one system of numbering system to the other. For example, the number 365 in base 10 format can be represented as

$$365 = 3 \times 10^2 + 6 \times 10^1 + 5 \times 10^0$$

If the same number is to be represented in binary format, we need far more number of digits.

$$365 = 101101101 = 1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

A pure binary system is of course rarely used for NC input. Instead a more sophisticated BCD (Binary Coded Decimal) system is usually employed. It is an encoding for decimal numbers in which each digit is represented by its own binary sequence. To encode decimal number in BCD format we may use the common encoding, where each decimal digit is stored in a four-bit nibble.

Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

The number 365 in BCD format is

$$365 = 0011\ 0110\ 0101$$

Its main advantage is the easy conversion to decimal digits, and thus it allows for faster decimal calculations.

9.4.4 Part-Program Input

Part program after preparing, needs to be entered into the machine-tool control unit for the purpose of execution. For this purpose a variety of methods are used:

- Paper tape (now obsolete)
- Manual Data Input (MDI)
- Direct Numerical Control (DNC)

The punched paper tape is a 1-inch wide tape with 8 tracks of holes that are used for characters along with a set of sprocket holes. The punched tape as per the standard code represents the program characters on its eight tracks. It can be observed that BCD coding specifies the numerals 0 through 9 on the first four tracks while other characters are represented by using additional tracks 5 to 8 to get a wider range. A row of feeding holes is present between the third and fourth tracks. It extends through the length of the tape and is used to engage with the teeth of a sprocket which transports the tape. These feed holes have no relationship with the holes which have been punched according to the code.

Although the tape-punching equipment is reliable, there is always a possibility that some holes may not be detected or some holes may be spuriously added. To help detect this error, the standards prescribe either odd number of holes or even number of holes for each character. Thus, the EIA (Electronic Industry Association) code specifies an odd number of holes, while the ISO code and ASCII codes prescribe even number of holes. For example, with EIA coding using odd parity, when in a particular row even number of holes are present, e.g., for number 3 (0011), then the punching system adds a hole in the parity track to make the total number of holes in that row to be odd. The codes thus use one of the tracks (for example, the fifth track in the EIA code for odd parity) to add a hole to bring the BCD based holes to the desired parity.

In the case of manual data input, the machine control unit (Fig. 9.8) provides a keyboard to directly enter the program into the main memory of the MCU. Since most of the present-day MCUs have large memory, it is possible to store a number of programs in the memory as well as edit them as required. Also, most of the modern-day MCUs are provided with a USB port or Ethernet port as part of the communication facility. The part programs can be entered in ASCII mode using any simple text editor in a personal computer and then transferred to MCU through the USB port or over the LAN connection.

Direct Numerical Control or DNC refers to a system where a Personal Computer (PC) is connected to the MCU through a serial port (RS 232C). More details of DNC are given in Chapter 20.

9.4.5 NC Tooling

The operator gathers, or is supplied with, the relevant tooling for the part to be machined. A distinctive deviation of the NC tooling from the conventional one is that each cutting tool is set in a different adapter (Fig. 9.9). The configuration suggested by ISO is now generally followed. A power-operated drawbar may be employed to pull the tooling at the retention knob. This helps eliminate any clearance between the mating surfaces of the spindle and tooling shank. It is not uncommon to set apart an allocation of 20 to 30% of total budget for tooling during the buying of new NC machine tools.

A preset tool has adjustable locating faces as shown in Fig. 9.10. It enables the dimensions between the tool-cutting edges and location faces to be preset to a close tolerance using a presetting device. The preset tool usually needs to be removed from the machine for adjustments required during batch production. The tools may be stored on a drum, which is operationally an integral part of the machine itself. In the latter case, the tools are automatically replaced or changed in the spindle.

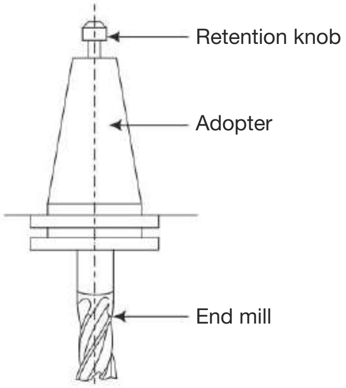


Fig. 9.9 *Typical spindle tooling holding an end mill*

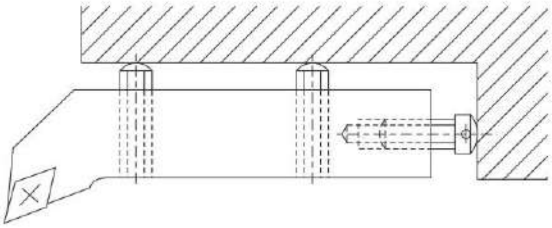


Fig. 9.10 *Typical preset tooling used in CNC turning machines*

These inform the operator about the deviation the tool tip of the actually supplied tool has with the one taken into account by the part programmer. The programmer gets the information from the tool files that are updated periodically. In spite of the 'updating', the position of the tool tip when supplied to the operator may be different (from what is mentioned in the tool file) because of wear and tear, resharpening or setting of a new cutting tool due to breakage.

9.5 NC MACHINE TOOLS

The basic objective behind the development of NC machine tools continues to remain the reduction of cost of production and improvement in product quality. The major emphasis is directed towards the avoidance of non-productive time which is mainly due to the number of set-ups, set-up time, workpiece-handling time, tool change time and lead time.

9.5.1 NC Machine-Tool Applications

NC machines have been found quite suitable in industries such as the following:

1. For the parts having complex contours that cannot be manufactured by conventional machine tools.
2. For small lot production, often for even single (one-off) job production, such as for prototyping, tool manufacturing, etc.
3. For jobs requiring very high accuracy and repeatability.
4. For jobs requiring many set-ups and/or the set-ups that are very expensive.
5. The parts that are subjected to frequent design changes and consequently require more expensive manufacturing methods.
6. The inspection cost is a significant portion of the total manufacturing cost.

One or more of the above considerations would justify the processing of a part by an NC machine tool.

9.5.2 Advantages of NC

NC is superior to conventional manufacturing in a number of ways. The superiority comes because of the programmability. These are as follows:

1. Parts can be produced in less time and, therefore, are likely to be less expensive. The idle (non-cutting) time is reduced to absolute minimum. This, of course, depends on the way the part program for the part is written. The endeavour of the machine-tool builder is to provide facility whereby the non-cutting time can be brought to the barest minimum possible. It is possible to reduce the non-productive time in NC machine tools in the following ways:
   - by reducing the number of set-ups,
   - by reducing set-up time,
   - by reducing workpiece-handling time, and
   - by reducing tool-changing time.

   These make NC machines highly productive.
2. Parts can be produced more accurately even for smaller batches. In the conventional machine tools, precision is largely determined by human skill. NC machines, because of automation and the absence of inter-related human factors, provide much higher precision and thereby promise a product of consistent quality for the whole of its batch.
3. The operator involvement in part manufacture is reduced to a minimum and as a result, less scrap is generated due to operator errors. No operator skill is needed except in setting up of the tools and the work. Even here the set-up has been simplified to a very great extent.
4. Since the part program takes care of the geometry generated, the need for expensive jigs and fixtures is reduced or eliminated, depending upon the part geometry. Even when the fixture is to be used, it is very simple compared to a conventional machine tool. It is far easier to make and store part programs (tapes).
5. Inspection time is reduced, since all the parts in a batch are identical provided proper care is taken about the tool compensations and tool wear in part-program preparation and operation. With the use of inspection probes in the case of some advanced CNC controllers, the measurement function also becomes part of the program.
6. The need for certain types of form tools is completely eliminated in NC machines. This is because the profile to be generated can be programmed, even if it involves 3 dimensions.
7. Lead times needed before the job can be put on the machine tool and can be reduced to a great extent depending upon the complexity of the job. More complex jobs may require fixtures or templates if they are to be machined in the conventional machine tools, which can be reduced to a large extent.
8. CNC machining centres can perform a variety of machining operations that have to be carried out on several conventional machine tools, thus reducing the number of machine tools on the shop floor. This saves the floor space and also results in less lead-time in manufacture. This results in the overall reduction in production costs.
9. Many times the set-up times are reduced, since the set-up involves simple location of the datum surface and position. Further, the number of set-ups needed can also be reduced. All this translates into lower processing times. Many times, a component could be fully machined in a single machining centre or turning centre, each of which has wider machining capabilities. In conventional manufacture, if the part has to be processed through a number of machine tools which are located in different departments, the time involved in completion and the resultant in process inventory, is large. This is greatly eliminated by the use of NC machine tools.
10. Machining times and costs are predictable to a greater accuracy, since all the elements involved in manufacturing have to be thoroughly analysed before a part program is prepared.

11. Operator fatigue does not come into picture in the manufacturing of a part. The NC machine tool can be utilised continuously since these are more rigid than the conventional machine tools.
12. Tools can be utilised at optimum feeds and speeds that can be programmed.
13. The modification to part design can be very easily translated into manufacture by the simple changes in part programs without expensive and time-consuming changes in jigs, fixtures and tooling. This adds to the flexibility of manufacture.
14. The capability (metal removal) of NC machines is generally high because of the very rigid construction employed in machine-tool design compared to the conventional machine tools.

9.5.3 Limitations of NC

Though the NC machines have a range of advantages, there are certain limitations one should take care of while making a choice of them.

1. The cost of an NC machine tool is much high compared to an equivalent conventional machine tool. The cost is often 5 to 10 times. Also, the cost of tooling is high. This is a very high initial investment. All this makes the machine hourly rate high. As a result, it is necessary to utilise the machine tool for a large percentage of time.
2. Cost and skill of the people required to operate an NC machine is generally high in view of the complex and sophisticated technology involved. The need is for part programmers, tool setters, punch operators and maintenance staff (electronics and hydraulics) who have to be more educated and trained compared to the conventional machine operators.
3. Special training needs to be given to the personnel manning the NC machine tools. NC manufacturing requires training of personnel both for software as well as hardware. Part programmers are trained to write instructions in desired languages for the machines on the shop floor. They need also to be acquainted with the manufacturing process. Similarly, machine operators have to be prepared for the new NC culture. These factors are important for the successful adoption and growth of NC technology.
4. As NC is a complex and sophisticated technology, it also requires higher investments for maintenance in terms of wages of highly skilled personnel and expensive spares. The need for maintenance engineers trained in all the sub-systems present such as mechanical, hydraulic, pneumatic and electronics makes the job more difficult. Though the latest machines are equipped with a large number of diagnostic facilities, still maintenance is one of the major limitations.
5. The automatic operation of NC machines implies relatively higher running costs. Moreover, the requirements of conditioned environment for operating NC technology adds further to the running costs.

9.5.4 Practical NC Machines

The earliest development in NC machines, as discussed earlier, started with the milling machines in view of their application in aerospace industry. The typical first-generation NC milling machine which was basically a tool-room milling machine was converted by the addition of numerical control is shown in Fig. 9.11. These machines typically combined the functionality of drilling and milling machines.

The NC milling machines have been successfully improved with a variety of options to improve their productivity and flexibility. As a result, these are now called machining centres to reflect that versatility, and it is possible to do all the milling and hole-making operations to the highest possible accuracy, thereby reducing the other finishing operations. A typical CNC machining centre is shown in Fig. 9.12.

Fig. 9.11 *Early form of CNC milling machines: EZ-TRAK DX II milling machine (Courtesy, Bridgeport Machines Inc., Bridgeport, USA)*

Fig. 9.12 *Present-day production horizontal axis CNC machining centre Makino A55 (Courtesy, Makino Milling Machines Co. Ltd., Tokyo, Japan)*

The reduction in the idle time in the case of machining centres is done with the help of a number of automatic options. The main options with machining centres are the Automatic Tool Changer (ATC) used for changing the tool in the spindle in the shortest possible time of the order of 3 to 6 seconds. Similarly, an Automatic Pallet Changer (APC) is used to change the workpieces automatically in the shortest possible time of the order of 10 seconds. A typical CNC machining centre equipped with ATC and APC are shown in Fig. 9.13 with the safety guards removed.

In addition to these conventional machine tools, the copy milling machines are also provided with CNC control to provide better copying with more transformation flexibility as shown in Fig. 9.14.

Fig. 9.13 *High production horizontal axis CNC machining centre Makino A60 with automatic tool changer and automatic pallet changer (Courtesy, Makino Milling Machines Co. Ltd., Tokyo, Japan)*

Fig. 9.14 *CNC copy milling machine Makino FD NC-128 (Courtesy, Makino Milling Machines Co. Ltd., Tokyo, Japan)*

Similar to the milling machines, the lathes are also provided with CNC controls. The major innovation provided in the CNC lathes is the provision of a slant bed to help remove the chips from the machining zone more efficiently, while a large variety of software functions to machine the most complex axi-symmetric shapes. A typical CNC turning centre is shown in Fig. 9.15.

Another class of CNC turning machine is the chucker-type shown in Fig. 9.16 which is with a flat bed and is ideally suited for turning chucking components. The machine has the advantages of quick cycle time, high rapid rates and easy chip disposal.

The CNC turning centres are normally provided with gantry robots for work and tool handling as shown in Fig. 9.17.

Fig. 9.15 *CNC Turning centre with slant bed GF NDM-16 (Courtesy, George Fischer, Switzerland)*

Fig. 9.16 *CNC Chucker (Turning) LC16 (Courtesy, ACE Designers)*

Fig. 9.17 *CNC turning centre with a gantry loader for workpiece handling (Courtesy, George Fischer, Switzerland)*

Another important class of CNC machines used in the industry are the Electric Discharge Machines (EDM). In particular, the CNC wire-cut EDM machines are very common in which the relative motion of the workpiece with respect to the EDM wire is controlled in 2 or 4 axes for machining a variety of complex die shapes used in sheet-metal industries. A typical wire EDM machine is shown in Fig. 9.18.

Sheet-metal components are produced by means of blanking and forming dies. For short-volume production, the cost of sheet-metal dies becomes uneconomical. Hence, in CNC turret punch presses the ability to quickly change the die and punch set along with the coordinate movement can be obtained using a CNC control. A typical CNC turret punch press is shown in Fig. 9.19.

Fig. 9.18 *EDM wire-cut machine DWC 110HA (Courtesy, Mitsubishi Electric Corp., Tokyo)*

Fig. 9.19 *CNC turret punching press Amada Vipros 255 (Courtesy, Amada, Japan)*

Another class of CNC machines that are being very common are the CNC coordinate machines which are used for the dimension measurement and automatic inspection. A typical example is shown in Fig. 9.20.

Fig. 9.20 *CNC Coordinate Measuring Machine Mitutoyo Super RV304 (Courtesy, Mitutoyo Asia Pacific Pte. Ltd., Singapore)*

In addition to the applications, as explained above, a number of other applications can also be found for the CNC machine tools as follows:

- Grinding machines
- Gear-generating machines
- Press brakes
- Flame cutting machines
- Laser cutting machines
- Pipe-bending and forming machines
- Folding and shearing machines
- Filament winding machines
- Assembly machines

Summary

- Numerical control of machine tools which was first demonstrated in 1952, has been rapidly adopted and applied to a large number of manufacturing applications, with machining accounting for a major share of these applications.
- Numerical Control (NC) has been conceived with the developments in digital computers in the mid 20th century.
- Numerical control utilises numbers and words to control the various functions of machine tools.
- Machining instructions are translated from the part drawing through the use of numbers to automatically run the machine tool.
- For the successful operation of an NC machine tool, it is necessary to understand the various components in NC manufacturing.
- The developments in microprocessors have changed the NC to CNC by incorporating a computer inside the machine-tool control unit. This provides a number of benefits to the user.
- Numerical-control machine tools can be operated in point-to-point mode or contour mode. In the contour mode, it is possible to have simultaneous control of multiple axes in order to get complex part geometries.
- NC tooling is an important element in achieving proper geometry.
- The application of NC machine tools is most appropriate for complex contours, small-volume production, high accuracy, and frequent design changes.
- There are a number of advantages to be gained by the proper application of NC machine tools. The principal advantage is the reduction in processing time because of the reduction in set-ups, simplifying or eliminating jigs and fixtures, and reducing idle times.
- Machining accuracy of NC machining is improved without involving any operator skill.
- Most of the operations can be completed by the use of general-purpose cutting tools, thereby reducing the need for special form tools for complex shapes.

- In spite of the advantages, care has to be taken to provide appropriate support facilities for maintenance and special operator training.
- Though machining is a major application for NC, a large number of other metal-forming applications as well as other forms of manufacturing are successfully utilising NC.

Questions

1. Explain what you understand by the term 'numerical control'?
2. What are the factors that contributed to the development of numerical control?
3. Briefly explain the functions that are expected to be served by numerical control in machine tools.
4. Explain the principle of numerical control.
5. Show schematically the different forms of numerical control, viz., open-loop and closed-loop control systems.
6. What are the applications where numerical control is most suitable?
7. Give the advantages and disadvantages of numerical control of machine tools.
8. What is the difference between NC and CNC?
9. Explain the advantages to be gained by using CNC compared to NC.

10

CNC HARDWARE BASICS

Objectives

A CNC machine tool responds directly to the program that is in operation. The accuracy to be achieved depends to a great extent on the various elements that form part of the machine tool. It is therefore important to understand various elements within a CNC machine tool as to understand their contribution to the overall accuracy of machining. After completing the study of this chapter, the reader should be able to

- Understand the developments in the design and construction of modern CNC machine tool structures
- Know about the developments in the spindle design particularly for very high speeds
- Learn the various types of drives used in CNC machine tools for driving the spindle as well as the feed motion
- Understand the contribution of the various actuation systems used, such as the lead screw and slideways to the accuracy
- Learn about the use of various feedback sensors used in CNC machine tools to control the accuracy of motion
- Learn the way the axes systems in CNC machine tools is defined

10.1 STRUCTURE OF CNC MACHINE TOOLS

While designing the machine tool structure, it is important to provide sufficiently high static stiffness along with the best stiffness-to-weight ratio. The large static stiffness would allow for very small deflection of the structural elements under the operative load of the machine tool such as the cutting forces, weight of the workpiece, etc. It would, therefore, be possible to operate the machine tool optimally in a broad range of machining conditions. The stiffness-to-weight ratio is important to achieve better dynamic response. This is particularly true for those machine tools which are used for very high speed of movement, approaching up to 50 to 60 m/min in some of the recent high-speed machine tools.

Typical CNC machine tool design criteria that should be considered along with their operating considerations are shown in Table 10.1 (Rogers).

Table 10.1 *Some design criteria for CNC machine tool design [Rogers]*

Machine response	*Component characteristics*	*Operating and cost considerations*
Type of command signal	Undamped natural frequency	Reliability
Input configuration	Power requirement	Maintainability
Maximum feed rate	Friction characteristics	Cost of operation
Static accuracy	Inertia	Capital investment
Dynamic accuracy	Stiffness	Installation requirements
Magnitude of load	Amount of backlash	
Range of travel	Speed range	
Weight of moving members	Bandwidth	
Power source		

Typical machine tool structures used in CNC machine tools are generally cast-iron based with heavy ribbing to provide high stiffness and low weight. Further, the cast-iron structure provides the necessary material damping to reduce the vibrations which is essential for large material removal rates and high-speed machining. The heavy structural design is generally provided with the ribs at the strategic locations to improve the static stiffness. The frames are often optimised by the use of the finite element analysis techniques (discussed in Chapter 8). An example of a typical machine tool frame for a heavy machining centre is shown in Fig. 10.1.

Fig. 10.1 *Heavy machine tool structure of a CNC machining centre (Courtesy, Makino Milling Machines Co. Ltd., Tokyo, Japan)*

Sometimes, the use of welded steel structure, which decreases the weight to stiffness ratio is adopted for some machine tools. However, because of their limitation in the effective structural damping of the vibrations, they are not generally preferred for CNC machine tools.

Recently the use of concrete as a bed material is gaining ground. The main advantage of concrete is its low cost and better damping capacity. The slideways of the machine tool are secured to the concrete bed with studs after providing a resin seating as shown schematically in Fig. 10.2. The concrete bed as used in a CNC turning centre is shown in Fig. 10.3. The experimental damping curve for the concrete material which is six times better than that for a metallic structure is also seen in Fig. 10.3.

Another innovation present in many of the heavier CNC machine tools is that they do not require a separate foundation. They are generally provided with mounting pads with a damping and non-slip coating so that no anchoring is required.

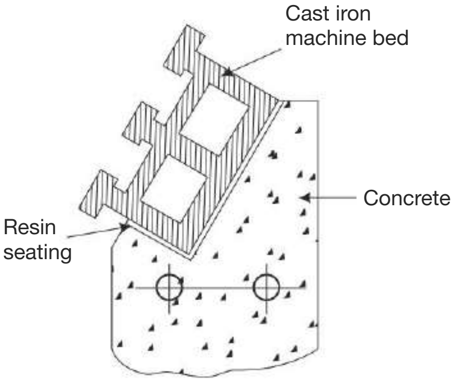


Fig. 10.2 *Schematic of a concrete bed of a CNC turning centre*

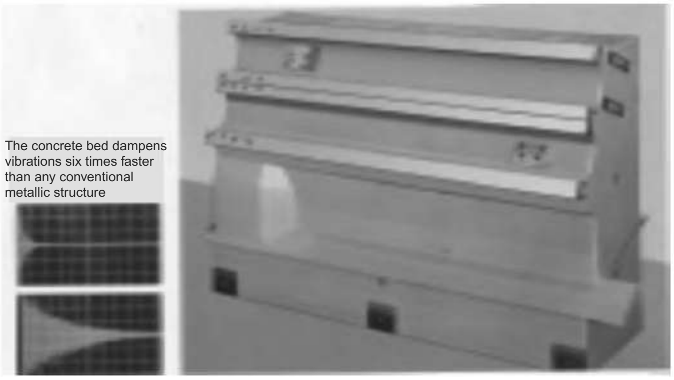


Fig. 10.3 *Concrete bed of CNC turning centre GF NDM-16 with slant bed (Courtesy, George Fischer, Switzerland)*

10.2 SPINDLE DESIGN

In the machine tool, spindle provides the necessary motion and power for the machining. Thus, it is a very important element whose accuracy is to be taken care of by proper design. The machining force is directly transmitted to the spindle as axial and radial forces. In CNC machine tools, because of the larger material removal rates, the magnitude of the cutting forces is so larger. Hence it is necessary in spindle design to see that the spindle deflection be minimised so as to get proper surface finish and also to reduce the possibility of chatter.

A typical spindle arrangement used in a CNC turning centre is shown schematically in Fig. 10.4. A typical feature to be noted in the design is that the spindle is well supported with very little overhang beyond the end bearings. The spindle is supported with sufficiently large ball and roller bearings to take care of the large axial and radial cutting forces.

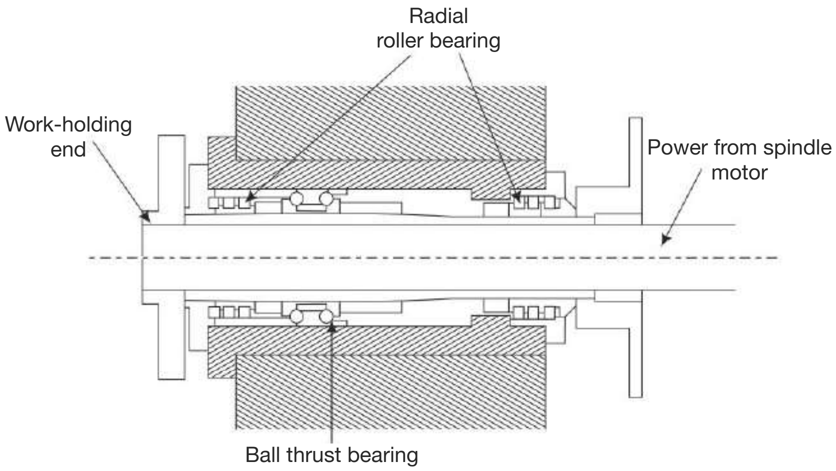


Fig. 10.4 *Spindle design for a CNC turning centre*

The heat generated in the spindle will be substantial because of the large power used and high spindle speeds used. It, therefore, becomes necessary to provide proper cooling of the spindle so as to maintain thermal equilibrium so that the spindle growth is maintained in reasonable limits. For this purpose, many methods are employed. The oil circulated through the spindle is provided with a heat exchanger so that the heat is removed continuously from the spindle. Optionally, it may also be provided with a chilling arrangement for more efficient heat removal from the spindle.

The spindle motor and the gearbox are isolated from the main structure so that part of the heat generated in them is not transmitted to the machine tool structure. Bridgeport has introduced a temperature compensation system in which four temperature sensors (three on the machine spindle and one for measuring the ambient temperature) are used to continuously measure and then get compensated by the MCU.

Another system being pioneered for high-speed machining is the building in of the spindle motor directly in the spindle housing as shown in Fig. 10.5. This reduces the overall inertia and allows faster cutting speeds. However, the motor is thermally separated from the spindle with a gap and is also provided with a spindle cooling arrangement as shown in Fig. 10.5. The spindle is supported in hybrid bearings (ceramic balls).

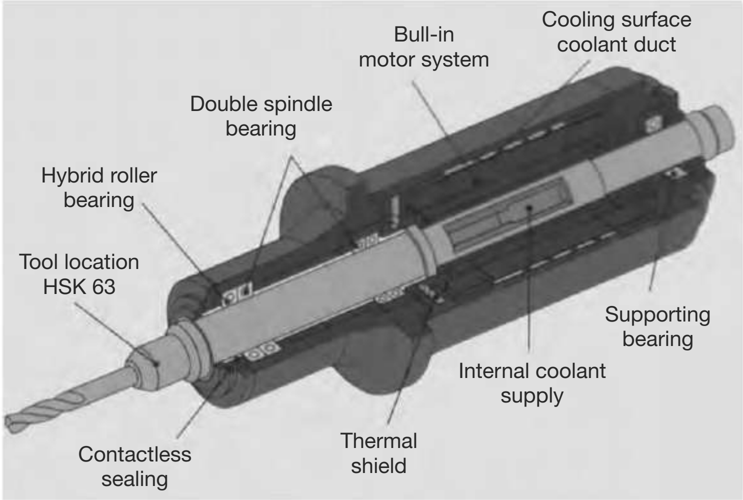


Fig. 10.5 *Spindle design with an integral Spindle Motor and Cooling System for a CNC Machining Centre (Courtesy, Hiller Hille GMBH, Germany)*

The spindle arrangements discussed are suitable for turning centres and horizontal axis machining centres. However, in the case of vertical axis machining centres, the spindle will generally move to provide the *Z*-axis movement. This causes the extended spindle to deflect under heavy machining conditions. Hence to provide the necessary rigidity in many a vertical machining centres, the whole spindle assembly will move instead of the spindle as shown in Fig. 10.6. In order to provide sufficient working area, it becomes necessary to keep the vertical spindle head away from the column which

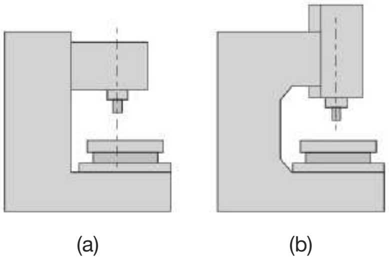


Fig. 10.6 *Spindle assembly with the slideways of CNC vertical axis machining centre*

causes more overhang from the spindle head as shown in Fig. 10.6(a). This can be reduced by modifying the column design to a C-frame as shown in Fig. 10.6(b).

It is also noticed that in the machining centres the cutting forces cause the twisting of the spindle. Hence a bifurcated structure shown schematically in Fig. 10.7 is used in many horizontal axis machining centres. The column housing the *Y*-axis movement is in a two-pillar structure between which is mounted the spindle housing which moves on two separate slideways as shown in Fig. 10.1. This increases the torsional rigidity of the support structure.

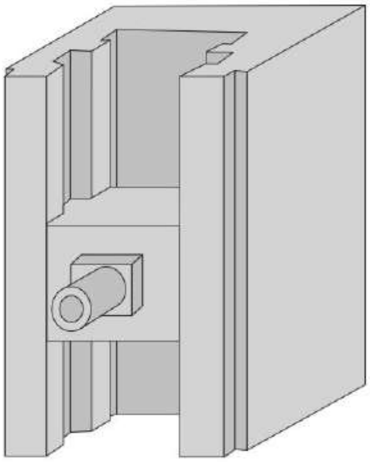


Fig. 10.7 *Bifurcated column structure for CNC machining centre to improve torsional rigidity (Also see Fig. 10.1)*

10.3 DRIVES

Following are the two drives used in CNC machines.

- Spindle drives to provide the main spindle power for cutting
- Feed drives to drive the axis as per the programme

10.3.1 Spindle Drives

In view of the large material removal rates used in the CNC machines, large power motors are used. Further, the speed is generally infinitely variable. Hence to provide such a control generally dc motors are used. The speed is controlled by varying the voltage infinitely. However, with the developments in the microprocessor controlled frequency converters, the use of ac motors is being preferred in the current generation of CNC machine tools. Typical torque rating of a spindle motor with ac drive is shown in Fig. 10.8. One more advantage of using the ac drive for spindle is that it can also be used for positioning the spindle axis (C axis) such as in turn mill centres.

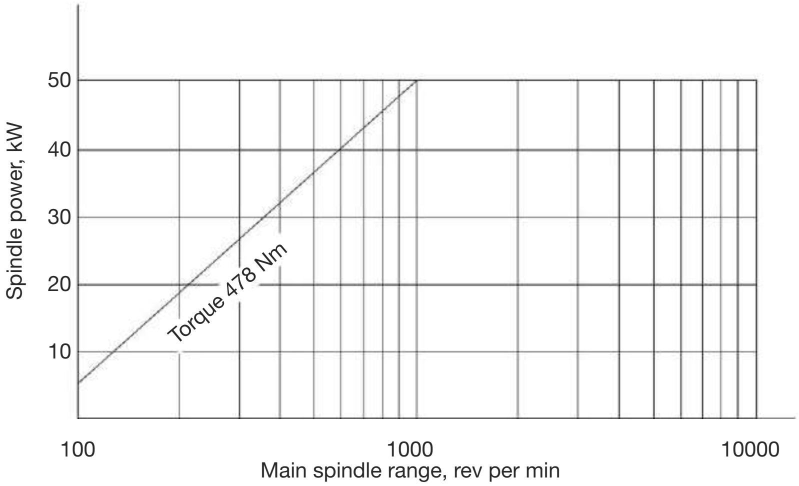


Fig. 10.8 *Typical torque–speed characteristic of a 50 kW ac spindle motor used for CNC machines*

10.3.2 Feed Drives

The feed drives that are used in CNC machine tools are the following.

- dc servomotors
- brushless dc servomotors
- ac servomotors
- stepper motors
- linear motors

DC Servomotors The force that rotates the motor armature is the result of the interaction between two magnetic fields (the stator field and the armature field). To produce a constant torque from the motor, these two fields must remain constant in magnitude and in relative orientation. This is achieved by constructing the armature as a series of small sections connected in sequence to the segments of a commutator. Electrical connection is made to the commutator by means of two brushes. As successive commutator segments pass the brushes, the current in the coils connected to those segments changes direction. This commutation or switching effect results in a current flow in the armature that occupies a fixed position in space, independent of the armature rotation and allows the armature to be regarded as a wound core with an axis of magnetisation fixed in space. This gives rise to the production of a constant torque output from the motor shaft. The axis of magnetisation is determined by the position of the brushes. If the motor is to have similar characteristics in both directions of rotation, the brush axis must be positioned to produce an axis of magnetisation that is at 90° to the stator field.

The dc servomotors are high-performance motors and are useful as prime movers in numerically controlled machine tools where starts and stops must be made quickly and accurately. The lightweight and low-inertia armatures of dc servomotors respond quickly to the excitation voltage changes. Also low armature inductance in these motors results in a low electrical time constant (typically 0.05 to 1.5 ms) that further sharpens motor response to command signals.

Brushless dc Servomotors In the brushless motor, the construction of the iron-cored motor is turned inside out, so that the rotor becomes a permanent magnet and the stator becomes a wound iron core. The permanent magnets, located on the rotor, requires that the flux created by the current carrying conductors in the stator rotate around the inside of the stator in order to achieve motor action. The stator windings are interconnected so that introduction of a three-phase excitation voltage to the three stator windings produces a rotating magnetic field. This construction speeds heat dissipation and reduces rotor inertia. The permanent magnet poles on the rotor are attracted to the rotating poles of the opposite magnetic polarity in the stator creating torque. The magnetic field in the stator rotates at a speed proportional to the frequency of the applied voltage and the number of poles.

In the brushless motor, the flux of the current carrying winding rotates with respect to the stator; but, like the dc motor, the current carrying flux stays in position with respect to the field flux that rotates with the rotor. The major difference is that the brush less motor maintains position by electrical commutation, rather than by mechanical commutation.

Stepper Motors A stepper motor rotates (steps) in fixed angular increments. Step size or step angle, is determined by the construction of the motor and the type of drive scheme used to control it. Typical step resolution is 1.8 degrees (200 steps per rev). However, micro-step motors are capable of 0.0144 degree steps (25 000 steps per rev). Micro-step motors are hybrid 200 step per rev motors that are electrically controlled to produce 25 000 steps per rev.

Stepper motors are usually used in open-loop control systems as shown in Fig. 10.9, even though an encoder may be used to confirm positional accuracy. There are many types of step-motor construction. However, Permanent Magnet (PM) and Variable Reluctance (VR) are the most common types.

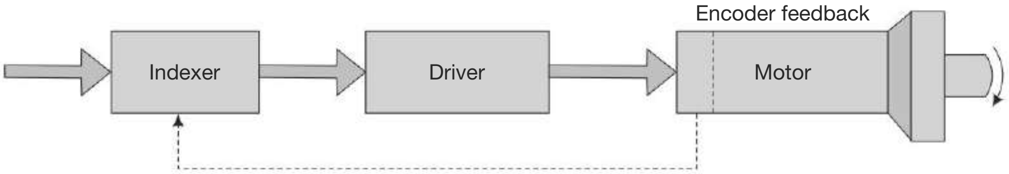


Fig. 10.9 *Typical step-motor system. Precise step systems have feedback loop (Dotted Line) using encoders or resolvers*

PM Step Motors The permanent-magnet step motor moves in steps when its windings are sequentially energised. Figure 10.10 illustrates a permanent magnet rotor surrounded by a two-phase stator. Two rotor sections (N and S) are offset by one half-tooth pitch to each other. As energy is switched from Phase 2 to Phase 1, a set of rotor magnets will align with Phase 1 and the rotor will turn one step. If both phases are energised simultaneously, the rotor will establish its equilibrium midway between steps. Thus, the motor is said to be half-stepping.

Stepper motors have the following benefits, which call for their use in motion-control applications.

- Low cost
- Ruggedness
- Simplicity in construction
- High reliability
- No maintenance

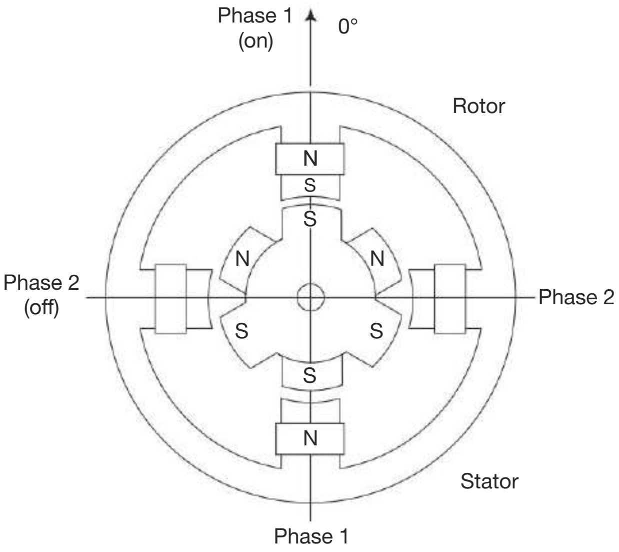


Fig. 10.10 *Step motor with permanent magnet rotor*

There is virtually no conceivable failure within the stepper-drive module that could cause the motor to run away. Stepper motors are simple to drive and control in an open-loop configuration. They only require four leads. They provide excellent torque at low speeds, up to 5 times the continuous torque of a brush motor of the same frame size or double the torque of the equivalent brush less motor. This often eliminates the need for a gearbox. A stepper-driven system is inherently stiff, with known limits to the dynamic position error.

Micro-step Motors These are usually hybrid motors. The rotor consists of one, two or three sets or stacks, of toothed cylindrical magnets. The toothed stator is wound so that alternate poles are driven by two separate phase currents. This results in a 200 step per rev. motor that when the two windings are energised proportionately, enable the motor to make 125 intermediate steps between each full step. Thus, using digital logic control and bipolar pulse width modulation, the motor makes 25 000 micro-steps per rev.

Linear Motors Recently linear motors are being increasingly considered for use in high performance CNC machine tools. The linear motor consists of a series of magnets attached to the machine base and a set of electrical coils potted around a steel laminate core attached to the moving slide. The fact that there are no mechanical parts in contact means that there is no wear or periodic maintenance required. Linear motors are not limited in travel like ball screws. Larger ball screws are required to achieve high velocity, with a longer travel to prevent undue vibration. This larger ball screw results in a higher inertia. This means a larger motor with more torque is required (introducing additional inertia) and the responsiveness and bandwidth of the system is reduced, resulting in poor servo performance.

Machines built with linear motors and all-digital drive systems can produce parts with higher accuracy and tighter tolerances at higher feeds and speeds. Also, they reduce significantly the non-machining time with high acceleration and deceleration rates.

Table 10.2 *Comparison of ball screws vs linear motors*

Characteristics	*Ball screw*	*Linear motor*
Max. speed	30 m/min (lead dependent)	120 m/min typical (180-240 m/min possible)
Max acceleration	0.5–1g	2–10g
Static stiffness	9–18 kgf/μm	7–27 kgf/μm
Dynamic stiffness	9–18 kgf/μm	16–21 kgf/μm
Settling time	100 ms	10–20 ms
Max. force	26.7 kN	9.0 kN/coil
Reliability	6000–10 000 hours	50 000 hours

10.4 ACTUATION SYSTEMS

Lead Screws The rotary motion from the drive motor needs to be converted to the linear motion to move the various axes of the machine tool. In conventional machine tools, the square (Acme) thread is normally used for this purpose. However, in view of the metal to metal and sliding contact between the nut and the screw, the friction is very high. This results in greater power being utilised for the movement of the axes. In view of the fact that it is necessary to increase the speeds of movement of the axes to the rates (about 30 m/min) which are typically used in most of the CNC machine tools, this friction as shown in Table 10.3 is a hindrance. Further, in view of the clearance provided between the nut and the screw in the case of Acme thread as shown in Fig. 10.11 to reduce friction, there is the problem of backlash whenever there is a reversal of motion. If any attempt is made to reduce the backlash, the friction increases. Hence most of the CNC machine tools use a lead screw with a recirculating ball nut.

Table 10.3 *Lead screw efficiencies*

	Efficiency (%)		
Type	*High*	*Median*	*Low*
Recirculating Ball screw - nut	95	90	85
Acme with metal nut*	55	40	35

(*Since metallic nuts usually require a viscous lubricant, the coefficient of friction is both speed and temperature dependent.)

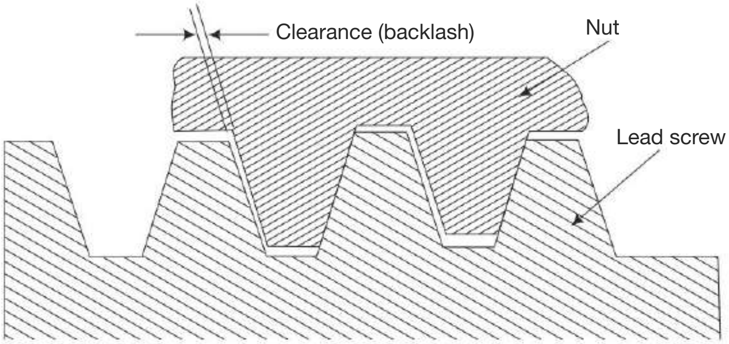


Fig. 10.11 *Lead screw with Acme nut*

In the case of recirculating ball screws, the nut is replaced by a series of balls which circulate in the channel in the form of threads as shown in Fig. 10.12. This results in a highly efficient rolling motion of balls in the space between the screw shaft and nut. The balls at the end of the thread portion in the nut will be repositioned back into the beginning of the thread form by a deflector as shown in Fig. 10.12. The size of the nut being an internal return of the balls is small as compared to the external return type using an external return tube.

Another type of nut used is where the balls at the end of the thread will be picked up by a return tube which recirculates the balls to the beginning of the load zone by providing continuous rolling motion (Fig. 10.13). This is the most common form used.

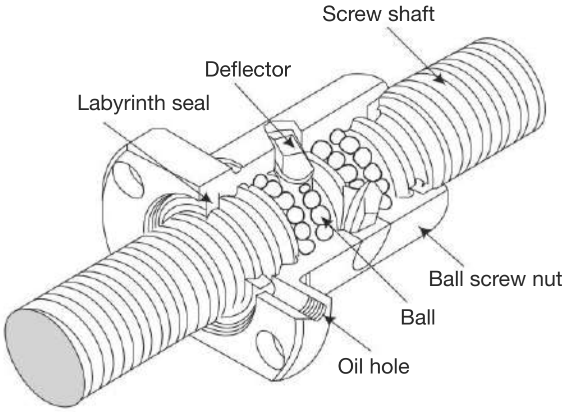


Fig. 10.12 *A recirculating ball screw and nut arrangement (Courtesy, THK Co. Ltd., Japan)*

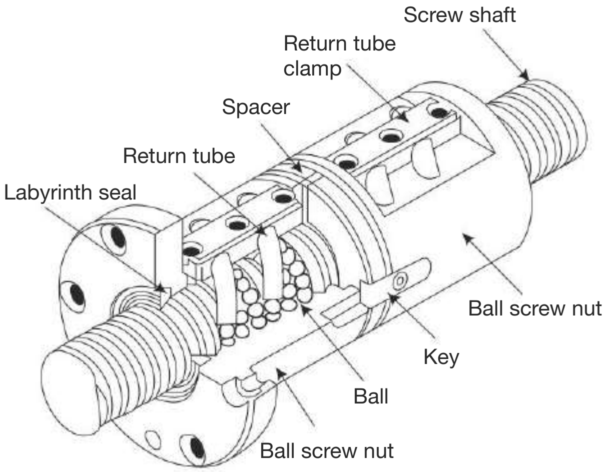


Fig. 10.13 *A recirculating ball screw and nut arrangement with external return tube (Courtesy, THK Co. Ltd., Japan)*

Further, the ball screws can be preloaded to eliminate the axial displacement which consequently also reduces the backlash. One of the method followed for pre-loading is keeping a spacer between the two nuts as shown in Fig. 10.14. This increases the axial rigidity of the nut while decreasing the axial displacement.

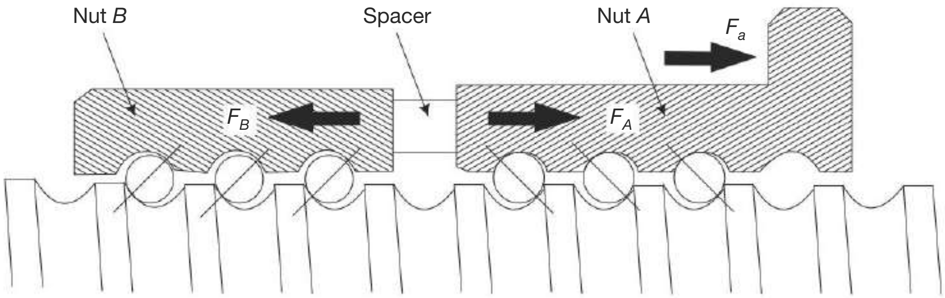


Fig. 10.14 *Preloading of the recirculating ball screw and nut arrangement*

The recirculating ball screws have a number of advantages in comparison to the conventional type of screws.

1. They have a longer life.
2. The wear of the screw is relatively small. Hence, it maintains accuracy through the entire life of the screw.
3. The frictional resistance offered is small, hence can be used for carrying heavier loads at faster speeds.
4. The power required for driving is small due to small friction.

Slideways Another important element for consideration during the design of the CNC machine tool is the slide motion. The conventional slideways (Fig. 10.15) such as the *V*, flat, round or dovetail have a large amount of friction because of the sliding contact between the sliding members. This will not allow for faster slide movement demanded by most of the CNC machine tools. As a result, a number of rolling friction elements capable of providing a very low friction have been developed, which are generally classified as linear-motion or LM devices.

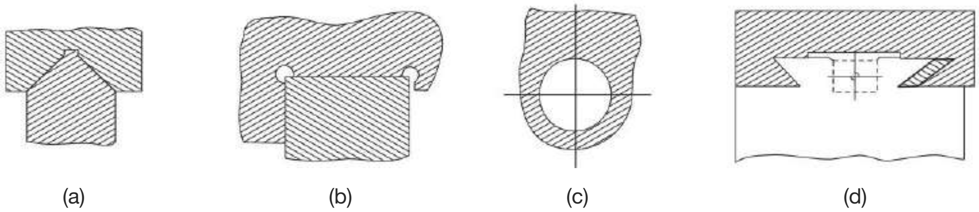

(a) (b) (c) (d)

Fig. 10.15 *Conventional slideway systems used in machine tools*

Linear-Motion Systems Since the friction is high in the conventional slideways, the antifriction slideways are generally used which makes use of the rolling friction by the use of recirculating balls. A typical linear motion guide using the rails is shown in Fig. 10.16. As shown in the cross-sectional view there are a number of recirculating balls providing a rolling motion between the slider and the rail. At the end of the block there

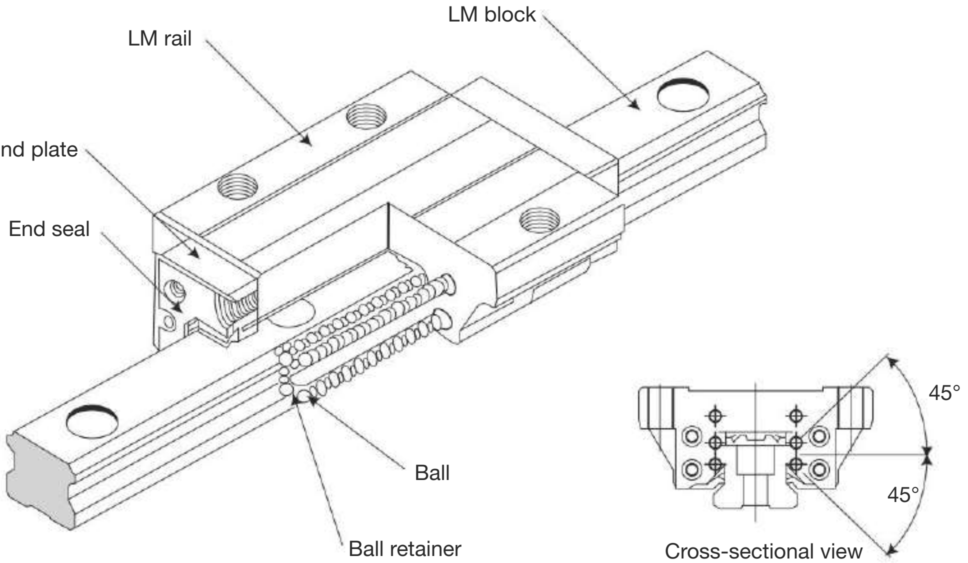


Fig. 10.16 *Antifriction guideways used in CNC machine tools (Courtesy, THK Co. Ltd., Japan)*

are end plates to ensure that the balls circulate through the rolling tracks. These provide a very high rigidity and very low friction for the movement of the axes. In view of the low friction, there is less wear and hence these systems are able to maintain the accuracy throughout its long life. A number of varieties of these LM guides are available off the shelf.

Another type of linear-motion device is the use of a ball bush (Fig. 10.17), where the balls are arranged in a track inside of a bush which can slide along a ground rod to provide the linear motion similar to a round slideway used in conventional machine tools.

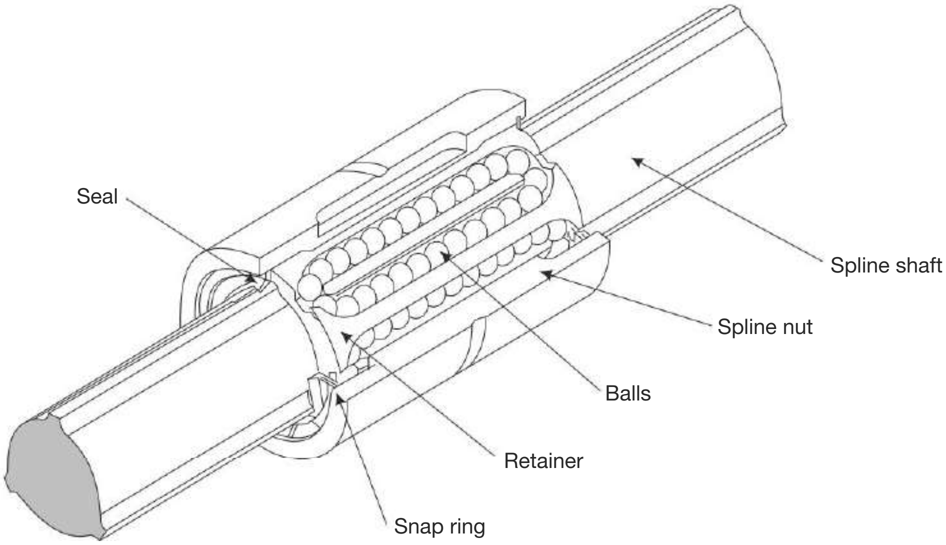


Fig. 10.17 *Ball bush used for linear movement in CNC machine tools (Courtesy, THK Co. Ltd., Japan)*

A typical machine bed is shown in Fig. 10.18 using the recirculating ball lead screw along with LM guides for providing a very fast feed motion.

10.5 FEEDBACK DEVICES

The CNC machine tools use closed-loop control system with an appropriate feedback to provide accurate control to the movement of the axes. It requires an appropriate feedback device as shown in Fig. 10.19 to provide the necessary input to the control system. The command position comes from the MCU as the actual amount programmed. This is compared in the comparator with the current position of the slide and provides the actual pulses required to move the motor. These pulses are converted to the analog signal by a DAC and fed through an amplifier to run the motor. The actual signal to the motor will be further compared by the velocity feedback obtained through the tacho generator.

A large variety of sensors have been used in CNC machine tools with varying success for providing the necessary measurement of the displacement (current position of the axis). The sensors that have become more common in the present day CNC machine tools are the following.

- Encoders
- Linear scales

Fig. 10.18 *Use of recirculating ball screw and the LM device for axis movement in the bed of a CNC machine tool (Courtesy, Makino Milling Machines Co. Ltd., Tokyo, Japan)*

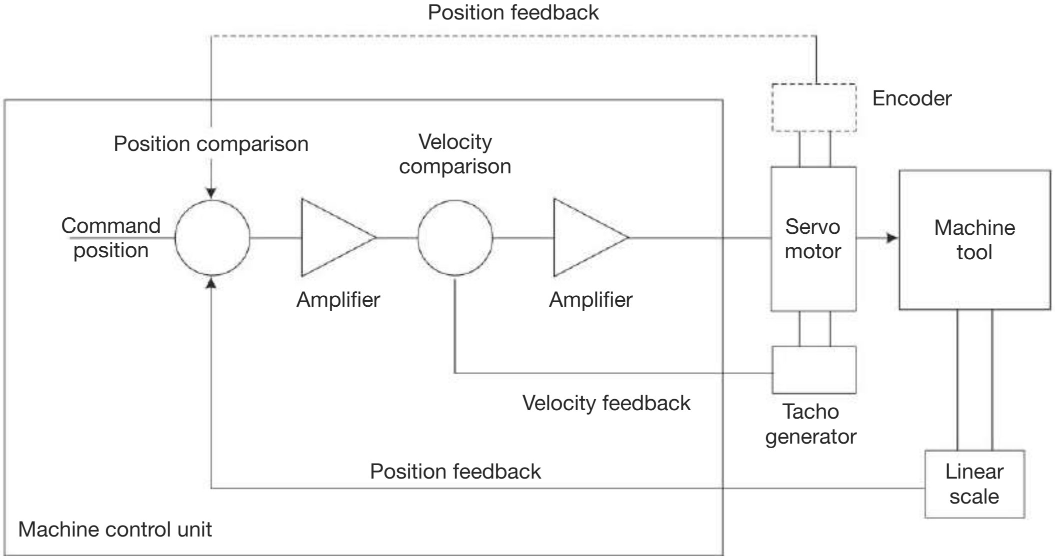


Fig. 10.19 *Closed-loop control system used for the control in a CNC machine tool*

The transducer that is connected directly to the rotor or the lead screw is the simplest arrangement requiring no additional gearing (many a servomotors come with integral rotary encoders). However the backlash present in the lead screw nut arrangement as well as any pitch errors of the lead screw may need to be eliminated or compensated by other means.

10.5.1 Optical Rotary Encoder

An optical rotary encoder converts the rotary motion into a sequence of digital pulses. The pulses are counted to convert to either absolute or incremental position measurement. The encoders generally come in two forms, absolute encoder and incremental encoder. The absolute encoder provides the exact rotational position of the shaft whereas the incremental encoder gives the relative position of the shaft in terms of digital pulses. The optical encoder consists of a disc (as shown in Fig. 10.20) with a number of accurately etched equidistant lines or slots along the periphery. The encoder disc is attached to the shaft of the machine whose rotary position needs to be measured. The disc is placed between a light source (generally infra red LED) and a light measuring device (photo diode). When the disc rotates the lines are interrupted and the light measuring device counts the number of times the light is interrupted. By a careful counting and necessary calculations it is possible to know the position traversed by the shaft.

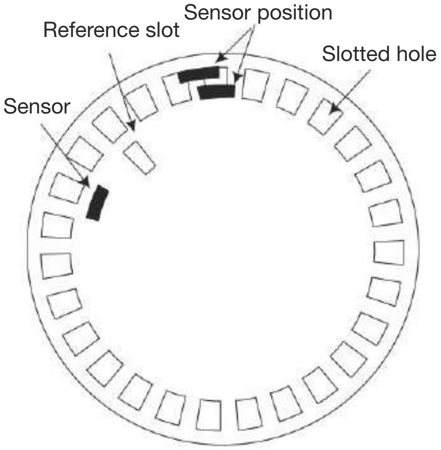
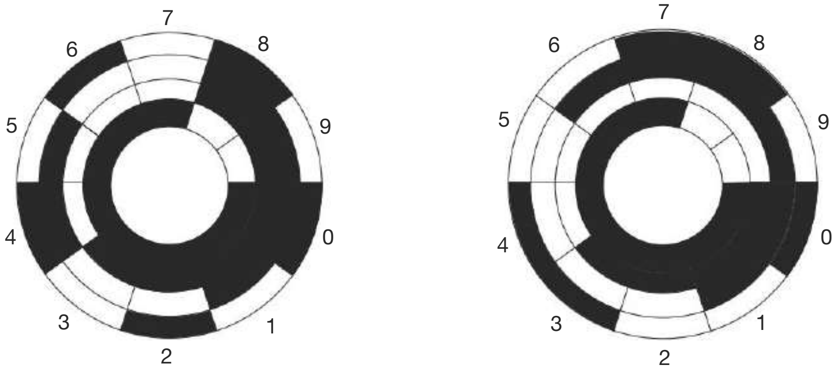


Fig. 10.20 *Encoder disc for rotary position measurement*

Absolute Encoder In case of an absolute encoder the encoder disc is etched with distinct positions, so that the associated sensor can know the exact position of the shaft. This is illustrated simply with a four-track encoder disc shown in Fig. 10.21. If it uses the natural binary code as shown in Fig. 10.21(a), then whenever the position changes, there is a possibility of more than one bit changing as shown (e.g. between numbers 1 and 2 as 0001 to 0010). If there is an error in sensing the change of bits, there is a possibility that the reading can be wrongly interpreted. To reduce this possibility, the grey code is designed as shown in Fig. 10.21(b) in such a way that only one track changes state for each count transition.



(a) Natural binary
(b) Gray code

Fig. 10.21 *Absolute encoder disc for rotary position measurement*

Incremental Encoder The encoder disk of an incremental encoder consists of one track and two sensors as shown in Fig. 10.20, whose outputs are called channel *A* and channel *B* (Fig. 10.22). As the shaft rotates, pulse trains occur on these channels at a frequency proportional to the rotational speed. The phase difference between these two signals yields the direction of rotation. With this arrangement, as shown in Fig. 10.21, the output gets multiplied by four times to yield a higher resolution. Incremental encoder provides more resolution at lower cost compared to the absolute encoder. However, it can only measure relative position, hence it has to be used in conjunction with another home position defined by a limit switch.

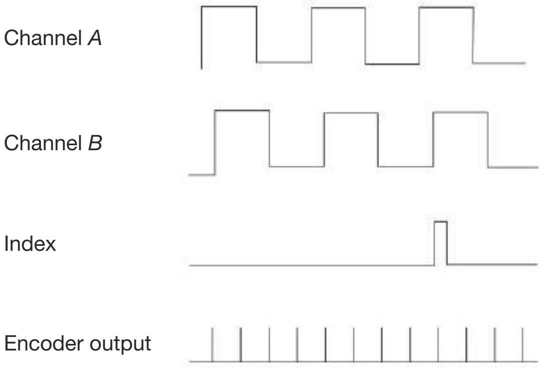


Fig. 10.22 *Operation of a digital rotary encoder for position measurement*

The encoder is directly mounted on the servomotor shaft or at the end of the lead screw as shown in Fig. 10.23. With such an arrangement the actual distance moved by the machine tool table needs to be calculated from the rotary motion by using the lead of the lead screw. If there is any backlash in the lead screw or there is a difference in the lead at various positions of the lead screw, then the actual distance moved by the table will be different from that calculated by the conversion of the position indicated by the encoder.

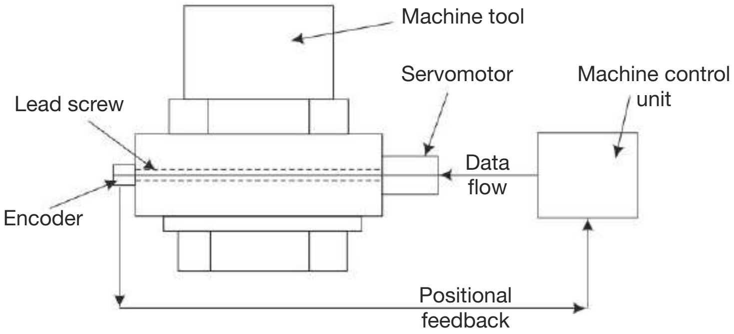


Fig. 10.23 *The encoder disc mounted on the lead screw for rotary position measurement*

10.5.2 Linear Scale

For knowing the exact position reached by the slide, it is better to measure the absolute position directly rather than in an indirect way using an encoder as in the above case. The linear scale provides such a system. In the linear scale, there is a finely graduated scale (grating) made of either glass or stainless steel, which provides a measuring surface along with a scanning unit. One of them is fixed to a stationary part of the machine tool while the other is fixed to the moving part as shown in Fig. 10.24. The scanning unit consists of a light source (such as infra red LED), a glass grid with graduated windows and some photo diodes as receptors.

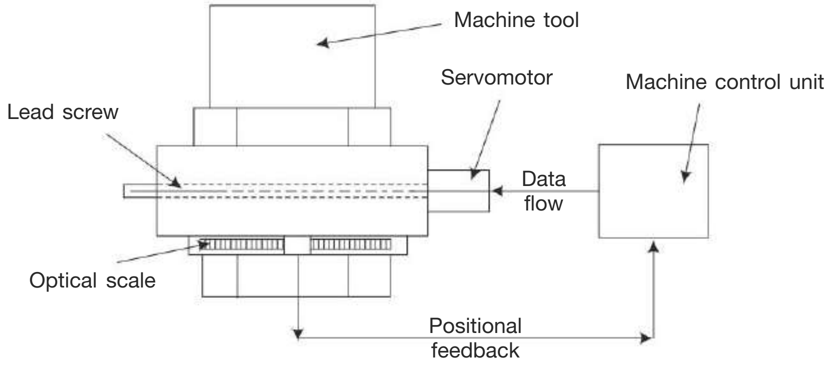


Fig. 10.24 *The linear scale fixed to the machine tool structure for direct position measurement*

For linear measurement in a linear scale, optical gratings are used whose principle is demonstrated in Fig. 10.25. When two gratings overlap each other, then depending upon the displacement, a Moirè fringe pattern is formed. It is possible to calculate the actual distance moved by the fringe pattern which depends on the grating spacing, the angle of the grating and the distance moved.

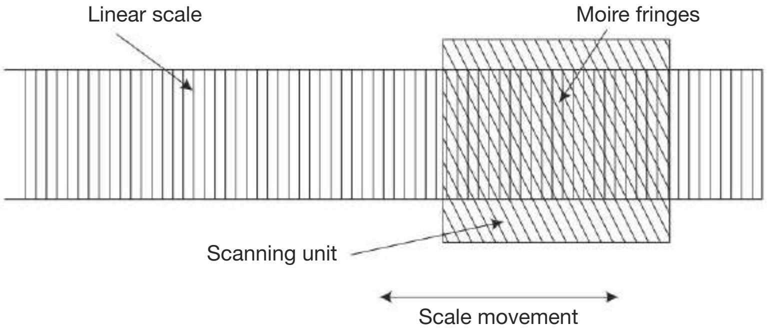


Fig. 10.25 *Principle of optical grating for position measurement in linear scales*

10.6 AXES—STANDARDS

The major component of a NC program involves the input of coordinates of the tool end point. To produce any machining profile, it is necessary to follow a proper coordinate system. To this extent the axes designation was standardised by EIA (Electronics Industry Association, USA) and ISO. Most of the NC machine builders follow the International Standard ISO/R841 to designate the axes of their machines. The principles followed in this standard are explained below.

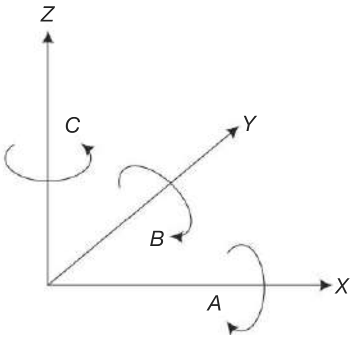
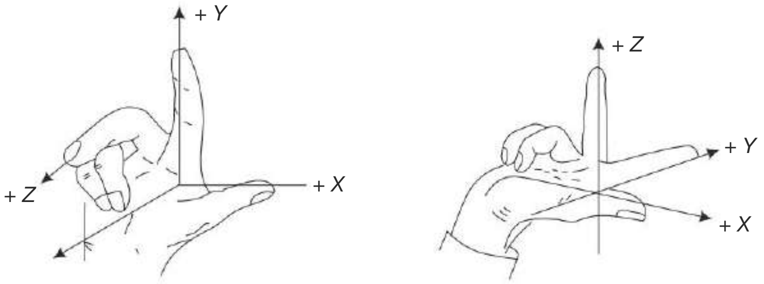


Fig. 10.26 *Right-hand coordinate systems*

Coordinate System All the machine tools make use of the Cartesian coordinate system for the sake of simplicity. The guiding coordinate system followed for designating the axes is the familiar right hand coordinate system. The main axes to be designated are the rectangular axes and the rotary axes. Typical right-handed coordinate system is shown in Fig. 10.26. One could use his right hand (Fig. 10.27) to arrive at these alternate variable positions of the same right hand coordinate system.



(a) Axis designation for horizontal *Z* (b) Axis designation for vertical *Z*

Fig. 10.27 *Finding directions in a right-hand coordinate system and also the positive directions for rotary motions*

Designating the Axes The first axis to be identified is the *Z* axis. This is then followed by the *X* and *Y* axes respectively.

10.6.1 Z axis and Motion

Location The *Z* axis motion is either along the spindle axis or parallel to the spindle axis. In the case of machine without a spindle such as shapers and planers, it is identified as the one perpendicular to the work-holding surface, which may or may not be passing through the controlled point (e.g., the cutting tool tip in case of shaper).

Direction The tool moving away from the work-holding surface towards the cutting tool is designated as the positive *Z* direction. This means in a drilling machine the drill moving into the workpiece is the negative (−) *Z* direction. This helps in reducing the possible accidents because of wrong part program entry in the coordinate signs.

When there are Several Spindles and Slideways In such cases, one of the spindles, preferably the one perpendicular to the work-holding surface may be chosen as the principal spindle. The primary *Z* motion is then near to the primary spindle. The tool motions of other spindle quills or other slides, which are termed as secondary and tertiary motions, may be designated as *U*, *V*, *W* and *P*, *Q*, *R* respectively.

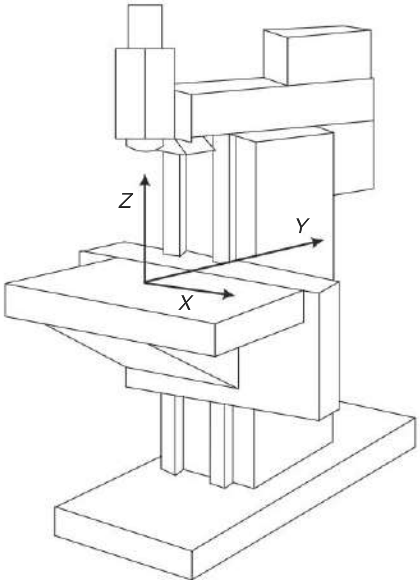


Fig. 10.28 *Vertical axis milling machine or machining centre*

For other machines, the positive (+) *Z* motion increases the clearance between the work surface and the tool-holder. The designation of *Z* axis is demonstrated in Fig. 10.28 for a vertical axis milling machine.

10.6.2 X axis and Motion

The *X* axis is the principal motion direction in the positioning plane of the cutting tool or the workpiece.

Location It is perpendicular to the *Z* axis and should be horizontal and parallel to the work-holding surface wherever possible.

Direction When looking from the principal spindle to the column, the positive (+) *X* is to the right. For turning machines, it is radial and parallel to the cross slide. *X* is positive when the tool recedes from the axis of rotation of the workpiece. For other machine tools, the *X* axis is parallel to and positive along the principle direction of movement of the cutting or the guided point.

10.6.3 Y axis and Motion

It is perpendicular to both *X* and *Z* axes and the direction is identified by the right-hand Cartesian coordinate system.

Rotary Motions *A*, *B* and *C* define the primary rotary motions.

(a) Location These motions are located about the axis parallel to *X*, *Y* and *Z* respectively. If, in addition to the above mentioned primary rotary motions, there exist secondary rotary motions, those should be designated as *D* or *E* regardless of whether they are parallel or not to *A*, *B* and *C*.

(b) Direction Positive *A*, *B* and *C* are in the directions which advance right-hand screws in the positive *X*, *Y*, and *Z* directions respectively. In Fig. 10.27, the fingers of the right-hand point towards the positive direction of the rotary motions.

As already discussed, most of the machine tool manufacturers adhere to the standard to a very great extent. However, some deviations may be present in some cases because of the historical reasons or specific convenience in operation or programming of the machine tool. Some examples of the axes designation as suggested above and applied to practical machine tools is described below. A turning centre with twin turrets and a rotary axis is shown in Fig. 10.29. The primary turret is given the designation of *X* and *Z*, while the second turret which moves independently is given *U* and *W* designations. The rotary axis for indexing the workpiece becomes a *C* axis. The rotary motion of the spindle is not a controlled axis and will not be designated as such in ordinary turning machines.

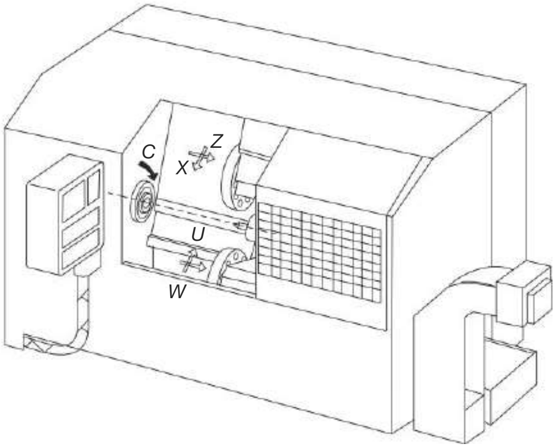
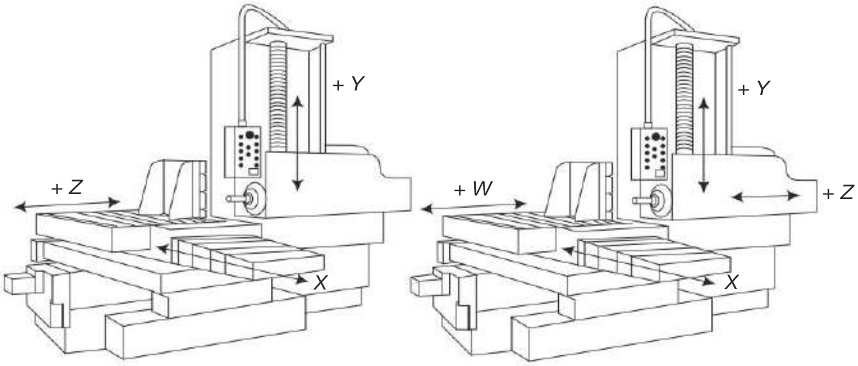


Fig. 10.29 *Axes designation for CNC turning centre with twin turrets and driven tooling*

A typical horizontal axis boring mill in three and four axes configuration is shown in Fig. 10.30. In the four-axes version, a complimentary motion parallel to the spindle movement (*Z* axis) is designated as *W* axis.

A five-axes machining centre with a horizontal spindle is shown in Fig. 10.31. In addition to the normal three-axes (*X*, *Y* and *Z*), two rotary axes *A* and *B* are added. In one case, the spindle originally horizontal is swivelling about the *X* axis. A rotary table is added on the table to give a rotary motion about the *Y* axis.



(a) Three-axis bring mill (b) Four-axis bring mill

Fig. 10.30 *CNC horizontal axis boring mills in 3 or 4 axes versions*

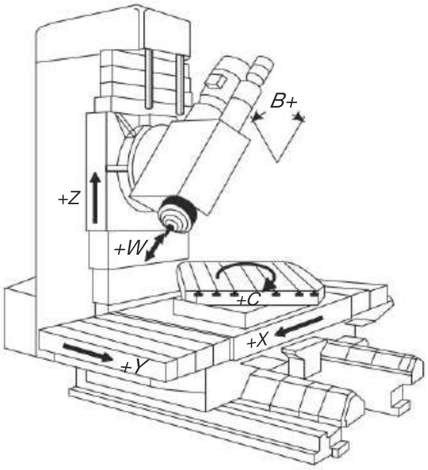


Fig. 10.31 *Five-axes CNC vertical axis machining centre configuration*

Summary

- Application of NC machine tools depends upon the capability. The accuracy and capability of a CNC machine tool is controlled by the number of subsystems that are present in it. Understanding the subsystems helps in getting to know the way to control the final accuracy of machining.
- Rigidity of the machine tool structure plays a very important role in machining accuracy by reducing the unwanted deformation in the structural elements. The modern machine tool structures are inherently very heavy with welded steel and concrete as some of the forms for bed construction.
- In view of the very high speeds in machine tool spindles, bearings and heat dissipation is an important requirement with most of the spindle designs.
- Spindle drives and feed drives used in CNC machine tools require infinitely variable speed control and a number of drive varieties are available for the purpose. Brushless servo motors are universally used in CNC machine tools.

- Recirculating ball screws and nut systems are universally used because of their low friction and high accuracies. There are a number of nut designs that are used for the purpose.
- The antifriction slideways such as the linear motion devices are used for slide movement to provide very high feed rates.
- Most of the CNC machine tools run on feedback loop and for this purpose, a number of sensors such as encoders and resolvers are used.
- The axes designation is standardized by ISO and is generally used by most of the machine tool manufacturers. The *Z*-axis is the principal spindle axis in the right hand cartesian coordinate system used in machine tools.

Questions

1. What are the requirements of structure in the case of CNC machine tools?
2. What are the design criteria to be used in designing CNC machine tools?
3. What are the various types of materials used in the construction of CNC machine tools?
4. What factors should be kept in mind during the design of spindles for CNC machine tools?
5. Briefly describe about the type of electric drives used in CNC machine tools.
6. Why is a recirculating ball screw universally used in the actuation system in CNC machine tools?
7. List different types of nut arrangements used in recirculating ball screws. Give a comparative evaluation.
8. State the advantages of recirculating ball screws compared to the conventional Acme screws.
9. Give a brief description of the linear motion elements as used in CNC machine tool slides.
10. What are the requirements of feedback devices in CNC machine tools?
11. Give a comparison of the encoder and linear scale as a feedback device for displacement in CNC machine tools.
12. Give a brief write-up on encoders used in CNC machine tools.
13. Briefly explain the basis of designating the coordinate axes in CNC machine tools.
14. Examine the following CNC machines (Fig. 10.32) and show the axes designation as per the ISO standard. Justify your choice.

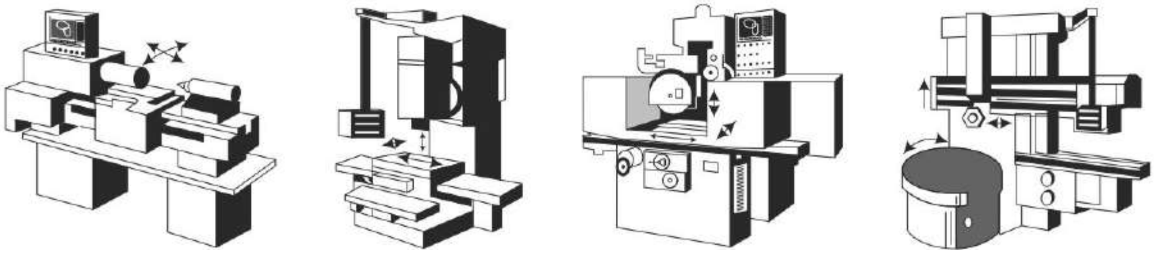


Fig. 10.32 *Typical CNC machine tools*

11

CNC TOOLING

Objectives

A CNC machine tool is as good as the cutting tools it uses. A large variety of tools and tool materials are used in CNC machine tools to improve the productivity of these machines. After completing the study of this chapter the reader should be able to

- Learn the different types of cutting-tool materials that are most commonly used in CNC machine tools
- Discuss the turning-tool geometry with an aim to select the appropriate geometry of the tool to suit the part-geometry configuration
- Understand the need for modular turning tools to facilitate the quick tool-change requirements
- Understand the milling tooling systems to cater to the automatic tool-changing systems
- Know the need for tool presetting for machining centre tooling as well as for the turning centre tools
- Learn the different types of tool magazines suitable for the automatic tool-changing function
- Discuss the automatic tool-changing function as used in the CNC machining centers
- Understand the work-holding requirements for CNC machine tools
- Calculate the power and time required for carrying out the various types of machining operations

11.1 CUTTING-TOOL MATERIALS

Various cutting-tool materials have been used in the industry for different applications.

11.1.1 High-Speed Steel

High-Speed Steel (HSS) tool materials have significant quantities of tungsten, (W), molybdenum, chromium and vanadium. The complex carbides of tungsten, molybdenum and chromium distributed through out the metal matrix provides very good hot hardness and abrasion resistance. The major alloying elements, which contribute to the hardness is tungsten and molybdenum. Tungsten is expensive, while molybdenum is cheap but has higher toughness. For the same hardness, less amount of molybdenum needs to be added, however more care need to be exercised in hardening as decarburising takes place in molybdenum steels. Also they have narrow temperature range for heat treatment. Molybdenum tool steels are more popular.

The main advantages of high-speed steels is their high hardness, hot hardness, good wear resistance, high toughness and reasonable cost. Toughness of high-speed steels is highest among all the cutting tool materials. Thus they are quite extensively used in interrupted cutting such as in milling. The hardness of HSS falls rapidly beyond 650°C and thus they are limited to lower cutting speeds of the order of 0.5 to 0.75 m/s.

The physical coating process (PVD—Physical Vapour Deposition) allows the HSS tools to be coated with hard nitrides of titanium and aluminium. With much favourable cutting geometries and the hard coatings the cutting performance and tool life of HSS tools has improved substantially. The PVD coatings are generally done at low temperatures as a result the adherence of coating is a problem, which is solved by improved cleaning and etching techniques. There are efforts to further improve the cutting performance by improving the coating characteristics by combining various nitrides.

11.1.2 Cemented Carbides

Cemented carbides are produced by the cold compaction of the tungsten carbide powder in a binder such as cobalt, followed by liquid-phase sintering. These have a very large number of advantages compared to the other cutting tool materials. The following guidelines would be useful for selecting a carbide grade.

(a) Choose a grade with the lowest cobalt content and the finest grain size consistent with adequate strength to eliminate chipping.
(b) Use straight tungsten carbide grades if cratering, seizure or galling are not experienced in case of work materials other than steels.
(c) To reduce cratering and abrasive wear when machining steel, use grades containing titanium carbide.
(d) For heavy cuts in steel where high temperature and high pressure deform the cutting edge plastically, use a multi carbide grade containing W-Ti-Ta and/or lower binder content.

As the cobalt content increases, toughness and impact strength of cemented carbide increase while hardness, Young's modulus and thermal conductivity decrease. Fine grain carbides are harder compared to coarse grain carbides. Multi-carbide grades increase chemical stability, hardness and hot hardness.

Since tungsten and cobalt are expensive, some special cemented carbides having predominantly tantalum carbides with Ni and Mo as binder have been developed for auto-industry application for finish machining of steels and malleable cast irons. These are some times called cermets. These are relatively brittle and easy to chip. These are relatively cheap and should find widespread use in future.

Cemented carbides being expensive are available in insert form in different shapes such as triangle, square, diamond and round. Each of the edge would act as a cutting edge. After the use of a single edge, the tip would be indexed in the cutting tool holder and thus these are called indexable bits. After all the edges are utilised, the tools are thrown out and a new bit is used in the tool holder. Thus, these are also called throwaway bits. Because of their brittleness, generally small negative rake angles are used with the bits. However, in view of the developments in the processing methods and compositions, a number of grades are being offered by the various manufacturers which can have a positive rake angle also.

11.1.3 Coated Carbides

Since the late 1960s, a thin (about 5 μm) coating of TiN has been used on cemented carbide tools. The life of the coated tools is often two to three times that of the uncoated, also these can be used at higher cutting speeds, thus increasing productivity. These coatings such as titanium carbide, titanium nitride, aluminium oxide, hafnium nitride and hafnium carbide or multiple coatings of the above, are deposited generally on the carbide tool bits by the Chemical Vapour Deposition (CVD) process. Multiple coating generally provides higher tool life and offers more broad use for machining differing work materials. By virtue of the general applicability of a single grade for a spectrum of machining situations, the shop needs to maintain an inventory of small number of varieties. Coated carbides are being increasingly used in the industry in comparison to the uncoated varieties.

11.1.4 Ceramics

Ceramics are essentially alumina-based high refractory materials introduced specifically for high-speed machining of materials which are difficult to machine such as cast iron. These can withstand very high temperatures, are chemically more stable and have higher wear resistance than the other cutting-tool materials. In view of their ability to withstand high temperatures, they can be used for machining at very high-speeds of the order of 10 m/s. The main problems of ceramic tools are their low strength, poor thermal characteristics and the tendency to chipping. They are not suitable for intermittent cutting or for low cutting speeds.

Apart from the pure alumina-based ceramics, sometimes other materials such as titanium carbide are added to enhance the transverse rupture strength. Some yittria may also be added as a sintering agent. Other ceramics of relatively recent origin are alumina−titanium diboride, alumina−zirconia−tungsten compound and silicon−aluminium−oxygen−nitrogen (Si−Al−O−N) complex compound. These are less hard than alumina ceramics, but are tougher.

Ceramic tools should be used with very high cutting speeds on steels. They are neither suitable for low cutting speeds nor for intermittent cutting. Cutting fluid if applied should be in flooding with copious quantity of fluid to thoroughly wet the entire machining zone, as ceramics have very poor thermal shock resistance. It can also be machined with no coolant. Ceramic tools are used for machining workpieces, which have high hardness such as hard castings, case hardened and hardened steels. Ceramic tools cannot machine some materials such as aluminium, titanium, since they have strong affinity towards them, as a result of which chemical reactions are expected.

Among other things, some of the vital requirements when machining with ceramics are the following.

- Use the highest cutting speed recommended and preferably select square or round inserts with large nose radius.
- Use rigid machine with high spindle speeds and safe clamping angle.
- Machine rigid workpieces.
- Ensure adequate and uninterrupted power supply.
- Use negative rake angles so that less force is applied directly to the ceramic tip.
- The overhang of the tool holder should be kept to a minimum, i.e., not more than 1.5 times the shank thickness.
- Large nose radius and side cutting edge angle on the ceramic insert to reduce the tendency of chipping.

- Always take a deeper cut with a light feed rather than a light cut with heavy feed as ceramic tips are capable of cuts as deep as one-half the width of the cutting surface on the insert.
- Avoid coolants with aluminium oxide based ceramics.
- Review machining sequence while converting to ceramics and if possible introduce chamfer or reduce feed rate at entry.

The recommendations and characteristics of various cutting tool materials have been summarised in Table 11.1. These can act as guidelines, however many of the cutting tool manufacturers such as Sandvik, Widia, Seco, Kennametal provide detailed literature to help in choosing cutting tools. These along with the Metal Cutting Handbook should be used for finalising the tool material selection.

Table 11.1 *Summary of applications for various cutting-tool materials*

Tool material	*Work materials*	*Remarks*
Carbon steels	Low strength, softer materials, non ferrous alloys, plastics	Low cutting speeds, low-strength materials
Low/medium alloy steels	Low strength, softer materials, non ferrous alloys, plastics	Low cutting speeds, low-strength materials
HSS	All materials of low and medium strength and hardness	Low to medium cutting speeds, low to medium strength materials
Cemented carbides	All materials up to medium strength and hardness	Not suitable for low-speed application
Coated carbides	Cast iron, alloy steels, stainless steels, super alloys	Not for titanium alloys, not for non-ferrous alloys as the coated grades do not offer additional benefits over uncoated
Ceramics	Cast iron, Ni-base super alloys, non ferrous alloys, plastics	Not for low speed operation or interrupted cutting. Not for machining Al, Ti alloys.

11.2 TURNING-TOOL GEOMETRY

As the cost of the CNC machine tool is high, it is necessary to use the machine to the fullest extent possible. The use is in terms of the actual time as well as the material removal capacity. In this respect the choice of the appropriate cutting tool and the process parameters make a lot of difference. Since the CNC machine tools have adequate rigidity as well as high spindle speeds, it is necessary to use either cemented carbide or ceramic tools according to the situations.

As of now a majority of the tools used are of the cemented carbide type with indexable insert type. It therefore becomes necessary to understand the ISO coding systems for these, to be able to easily make the selection. The ISO coding system (as per ISO 1832 - 1991) for tungsten carbide inserts and external turning tools is shown in extracted form in Figs 11.1 and 11.2.

The actual selection of the tools for a particular application has to carefully match the geometry. Generally the manufacturer's catalogue provides such application information. For example, referring to Fig. 11.3 the use of PDJNR type tools is for the external turning and requires that the maximum contour angle be 30° as shown in Fig. 11.3. It also shows other types of features that can be machined by the tool. Figure 11.4 shows the capability of internal turning tools to produce the contour bores. Similar capability charts can be found in the manufacturer's catalogues for the range of cutting tools available.

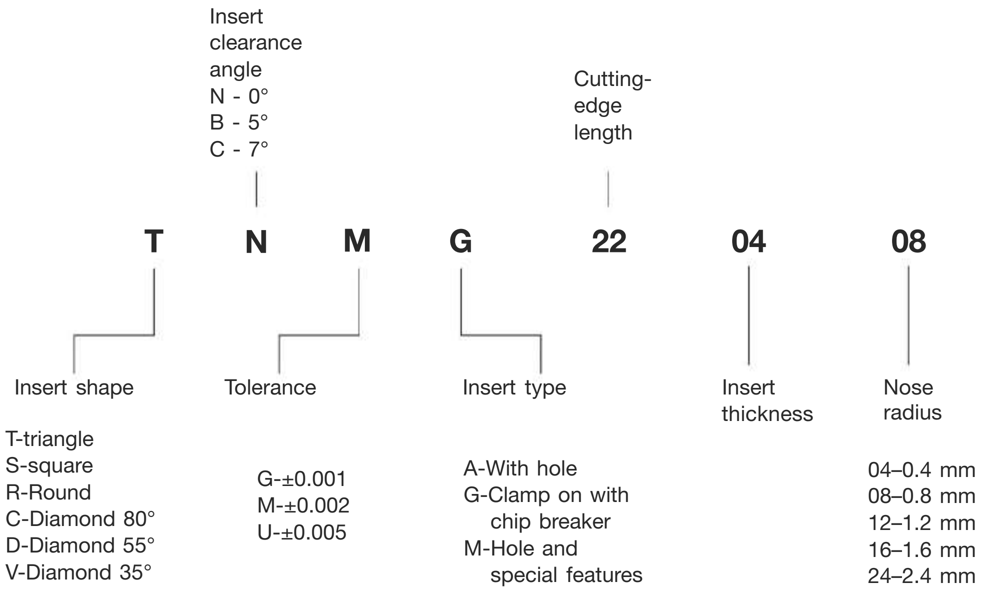


Fig. 11.1 *ISO coding system for tungsten carbide inserts used in turning*

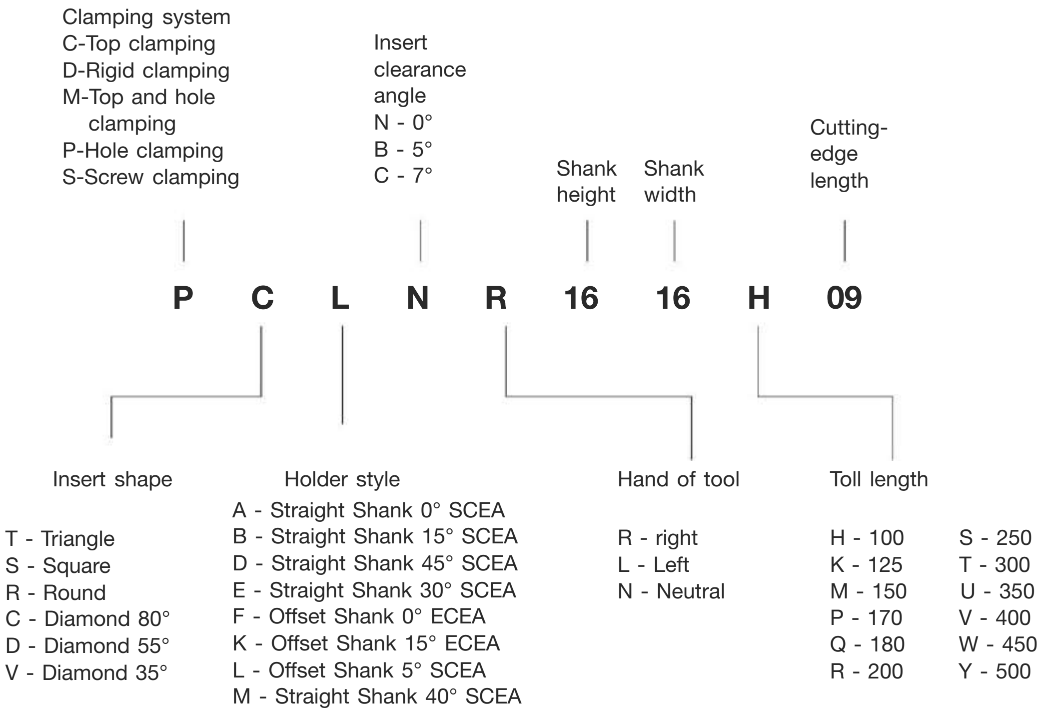


Fig. 11.2 *ISO coding system for tungsten carbide turning-tool holders used in external turning (SCEA–Side cutting edge angle, ECEA–end cutting-edge angle)*

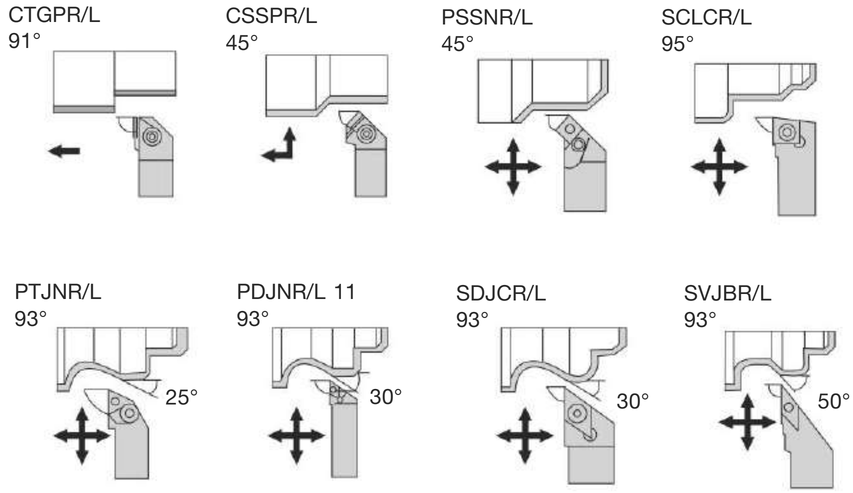


Fig. 11.3 *Typical contour capability of external turning tools (Courtesy, Seco Tools, Germany. Redrawn from Seco Catalogue)*

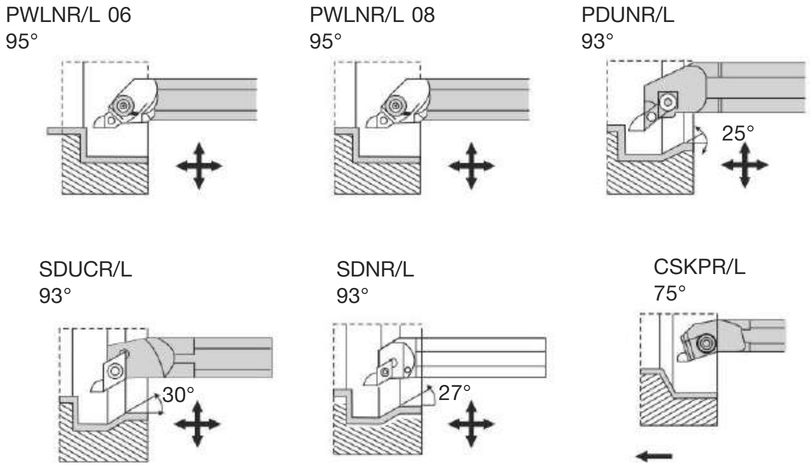


Fig. 11.4 *Typical contour capability of internal turning tools (Courtesy, Seco Tools, Germany. Redrawn from Seco Catalogue)*

Modular Tooling Systems for Turning Figure 11.5 shows typical tooling range used in a CNC turning centre. It can be noticed that in view of the variety of machining needs, the tooling systems used in CNC turning centres makes use of different shank sizes and shapes which requires different tool holders for clamping them securely in the tool turret. This makes the tool change function as well as the design of storage magazine a difficult one. This is changed by adopting a modular tooling system such that all the tools have the same type of holding method used, whether it is external or internal turning. An example is shown in Fig. 11.6 from the Seco Capto® system. It may be noticed that in these tool modules the clamping end remains the same, whatever is the type of tool. The typical clamping method for these tools is shown in Fig. 11.7 with the help of a tapered polygon.

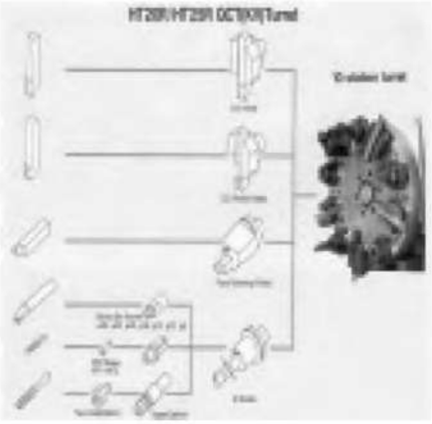


Fig. 11.5 *Typical range of tooling used in turning centres (Courtesy, Hitachi Seiki, Japan)*

Fig. 11.6 *Modular tooling used for turning tools (Courtesy, Seco Tools, Sweden)*

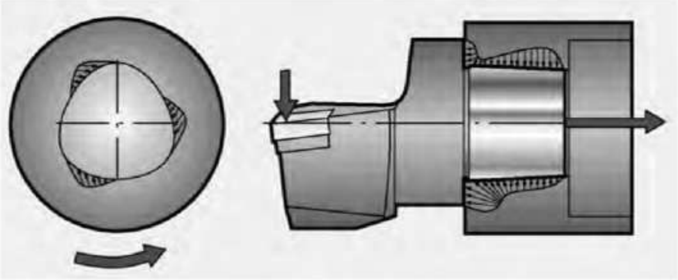


Fig. 11.7 *Clamping system used in the Seco Capto® Modular Tooling used for Turning Tools (Courtesy, Seco Tools, Sweden)*

In these systems the tool change takes only a few seconds, thus increasing the available productive time in the machine tool. The stability is very high. This means that the feed rate can be increased, the workpiece quality is raised and the useful life of the tools and inserts is increased. They have very high repeatable dimensions, which reduce the setting up times and ensure a longer productive time in the machine tool.

Similar systems are available from other tool manufacturers as well, such as KM system from Kennametal, FTS system from Hertel, etc.

11.3 MILLING TOOLING SYSTEMS

A milling tool to be used in CNC machine tool is an assembly of a number of parts besides the actual cutting tool as shown in Fig. 11.8. The assembly consists of the adopter to suit the spindle taper such as ISO 40 or 50, a collet for holding the straight shank of the end mill, a retention knob which is used by the hydraulic draw bar in the spindle housing for retaining or releasing the tool from the spindle, besides the actual cutting tool, the end mill. A typical retention knob design as used in BT spindle tooling is shown in Fig. 11.9.

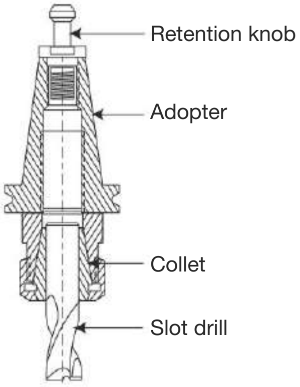


Fig. 11.8 *Complete tool assembly (for parallel shank tooling) as used in a CNC machining centre*

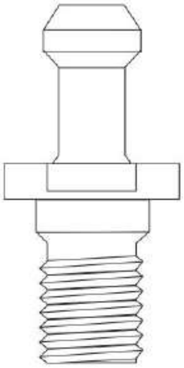


Fig. 11.9 *Retention knob as used in the top of tool assembly for clamping and releasing purpose as used in a CNC machining centre*

Since the CNC machine tools are versatile, they need to use a large variety of tools to accomplish the range of machining tasks they are capable of. Hence the general practice is to allocate a large portion of the budget for the tooling acquisition.

There are a number of shapes of the adopter depending on the machine tool standard followed by the machine tool builder. The typical BT style and the ANSI CATV style are shown in Fig. 11.10. The actual shape of the adopter will have to suit the tool change gripper whose details are given later.

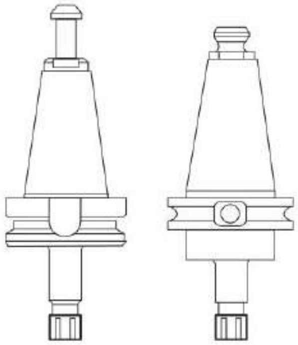


Fig. 11.10 *Adopter shapes for (a) BT style; (b) ANSI CAT-V style for CNC machining centres*

11.4 TOOL PRESETTING

Since the generation of actual geometry is taken care of by the CNC part program, which is essentially the coordinates through which the cutting tool tip moves, it is important to know the actual dimensions of the tool when it is placed in the spindle. The relationship of the tool with reference to the tool-holding mechanism requires a special attention during CNC machining process.

The actual point to be programmed in a CNC part program is the tip of the tool where the axes will be moving with respect to a known point in the spindle, e.g., the centre of the spindle in case of machining centres. It therefore becomes necessary to know precisely the deviation of the tool tip from the gauge point on the spindle. Hence the tool setting equipment is generally used. A simple mechanical type tool setting device is shown in Fig. 11.11.

In this system, the base is provided with the exact taper as used in the actual machine tool. The assembled tool is therefore placed in the spindle taper. The measurement is done with the help of a micrometer head, which is attached to a U-clamp as shown in Fig. 11.11. The U-clamp can be moved manually on a post, which has precise location slots that are separated by an exact distance (e.g., 25 mm). The length of the tool can therefore be measured by the measurement of the micrometer plus the slots along the post.

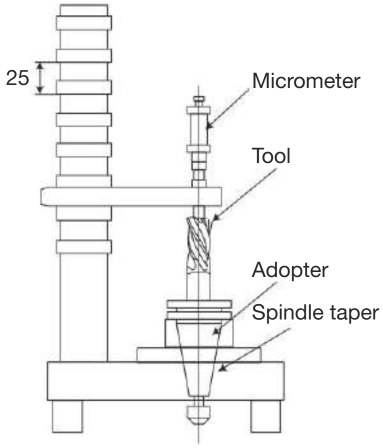


Fig. 11.11 *Typical tool-setting system useful for machining centre tooling*

Fig. 11.12 *Typical digital tool-setting system, which is essentially a digital height gauge useful for machining centre tooling (Courtesy, Trimos, Switerland)*

A similar tool-setting device which can measure the length as well as the diameter of a spindle tooling is shown in Fig. 11.12. The tool-measuring probe moves on two precise axes to measure both the length and the diameter of a spindle tooling unit. The display is shown digitally so that there is no error in measurement. Further, they are provided with a serial port for outputting the measured values directly into any tool management system or a personal computer for the purpose of generating the tool offset values. Typical specifications of one such tool presetting device is given in Table 11.2.

Table 11.2 *Specifications of a tool-presetting device (Trimos 301)*

Measuring range	$X = 100$ mm, $Z = 300$ mm max. dia. 200 mm
Resolution	0.01 mm/0.001 mm
Accuracy	$X = 10\ \mu$m, $Z = 20\ \mu$m
Repeatability	5 μm (+/− 2 S)
Concentricity of spindle unit related to the axis of the ISO location, max. error	0.02 mm/300 mm
Squareness of the measuring arm in relation to ISO location, max. error	0.15 mm/300 mm
Max displacement speed of measuring carriage and measuring arm	1 m/s
Digital display	LCD display, height of digits : 8.5 mm, indication for active functions
Data ouput	RS 232 compatible
Interface	RS 232 compatible interface cable with opto-electronic coupler

Some presetting systems are also provided with an optical projector such that the point of contact between the probe end and the tool can be more accurately identified. The other method that can be used for measuring the tools is the use of probes which is discussed later.

In the CNC turning centre, many machine-tool manufacturers are providing an integral tool-setting device as shown in Fig. 11.13. The measurement arm consists of a probe tip which when extended will meet the tool tip in two perpendicular directions for directly measuring the Z and X offsets.

Fig. 11.13 *Typical tool setter integrated with a CNC turning centre (Courtesy, Yamazaki, Japan)*

11.5 AUTOMATIC TOOL CHANGERS

During the operation of a machine tool, a considerable amount of time is spent in idle movement of tools such as tool engagement and disengagement, tool change and tool set-up. To improve the machine utilisation, it is necessary to minimise these idle motions. To that extent automatic tool changers or ATC as is popularly called, plays a very important role. These are particularly useful in machining applications where a number of tools are to be used for finishing the job. Though there are still some CNC machine tools which are sold without ATC, a majority of the more common CNC machine tools are with ATC. Further, ATC is one single factor which makes a CNC machine tool more autonomous with little operator intervention.

For the automatic tool changer to operate, it is necessary to have the following.

(a) A tool magazine where sufficient number of tools can be stored.
(b) The tool adopter that has a provision for pick-up by the tool change arm.
(c) The ability in the control to perform the tool change function.
(d) Tool change procedure.

11.5.1 Tool Magazines

Tool magazines to be used have to be considered in terms of the following attributes.

- Storage capacity
- Type and shape
- Tool-change procedure

Storage capacity typically starts with about 12 and can go as high as 200 while 30 to 60 appears to be the most common capacity of the tool magazines. The simplest type of tool magazine is a turret as shown in Fig. 11.14. This method combines tool storage with the tool-change procedure, without the need for a tool-change arm. The turret simply indexes to bring the tool into the position of machining, since the spindle is combined with the tool turret as shown in Fig. 11.15. The main advantage of this system is that the tool is identified directly with the pocket position and hence does not require a separate identification. Though it is a relatively simple method, the time taken for actual tool change is normally more except in the case of a tool in the adjacent pocket. Further the turret should have the capability of indexing in both directions to minimise the tool change time.

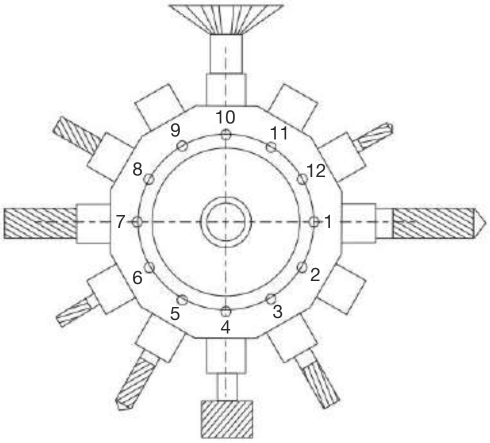


Fig. 11.14 *Typical tool turret used in CNC drilling/milling machines*

Fig. 11.15 *A CNC drilling machine using a tool turret (Courtesy, Fanuc, Japan)*

The next type of tool magazine found in most of the machine tools with lower tooling requirements is the drum or disc-type magazine. A typical drum-type tool magazine is shown in Fig. 11.16. The drum rotates for the purpose of tool change to bring the required tool to the tool change arm. The diameter of the disc is indicative of the number of tools it can hold. As the number of tools in the magazine increases, its diameter becomes too large to be practical. For storing large number of tools, a chain-type tool magazine provides the necessary flexibility. A typical chain-type magazine in a simple configuration is shown in Fig. 11.17. The tools are attached to the pockets which are in turn attached to the chain which is moving on appropriate sprockets. The chain allows for a very large variety of arrangements.

Fig. 11.16 *CNC machining centre with a drum-type tool magazine (Courtesy, OKK Machine Tools, Japan)*

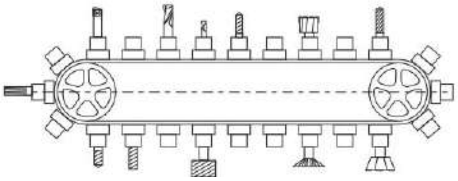


Fig. 11.17 *Chain-type tool magazine for holding larger number of spindle tooling used in CNC machining centres*

The chain can be arranged to follow any path, thereby increasing the capacity of the magazine as shown in Fig. 11.18. The capacity may be as small as 30 to and as high as 100 as shown in Fig. 11.19. It is also possible to make the chain type magazine by duplicating. For example, there can be two chain magazines of 30 each in the same machine tool. In such cases one chain can be made active such that the tools from that magazine will be used for machining. The other tool chain can then be used for replenishing the tools without interrupting the machine for machining. This is particularly useful in Flexible Manufacturing Systems (FMS) where the tool replenishment from a secondary tool store can be easily transported in the form of an entire tool chain. This is discussed in later chapters.

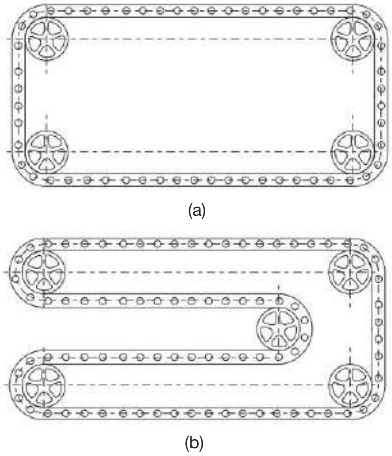


Fig. 11.18 *Variations of chain-type tool magazines*

Fig. 11.19 *An example of a chain-type tool magazine with a capacity of 100 tools (Courtesy, Dixi, Switzerland)*

11.5.2 Tool Changing

In the case of turret, the tool changing is relatively simple, because of the turret indexing. However, in the case of other tool magazines, it is necessary to have a tool changing arm which can provide the necessary tool transfer. Generally, the tool magazine is placed close to the spindle such that the actual tool transfer does not consume a lot of time. Typical tool-change times quoted by the various machine-tool manufacturers range from as low as 2 to a maximum of 10 seconds. The tool-change activity requires the following motions.

(i) Stopping the spindle at the correct orientation for the tool-change arm to pick the tool from the spindle
(ii) Tool-change arm to move to the spindle
(iii) Tool-change arm to pick the tool from the spindle
(iv) Tool-change arm to index to reach the tool magazine
(v) Tool magazine to index into the correct position where the tool from the spindle is to be placed
(vi) Place the tool in the tool magazine
(vii) Indexing the tool magazine to bring the required tool to the tool change position
(viii) Tool-change arm to pick the tool from the tool magazine

(ix) Tool-change arm to index to reach the spindle
(x) New tool is placed in the spindle
(xi) Tool-change arm moves into its parking position

The above sequence of events are true in the case of a tool-change arm which has only a single gripper. As can be noticed, in view of a long sequence of the events involved, the time taken for tool change in such cases is long. Hence most of the machine-tool builders use a double gripper (Fig. 11.20) in place of a single gripper. The use of double gripper allows to do some of the above tasks simultaneously.

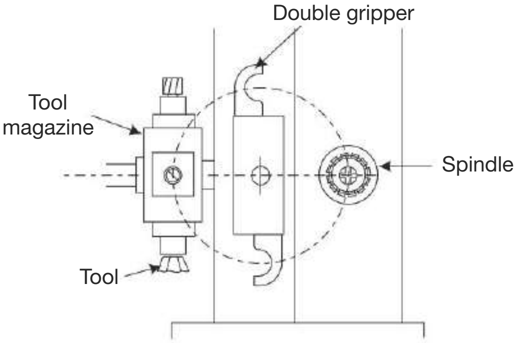
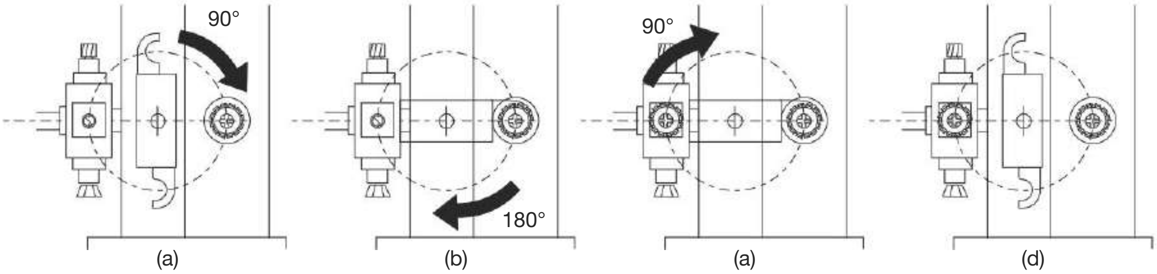


Fig. 11.20 *One common type of tool-change arm used for tool changing with a double gripper*

The following is a possible event sequence for tool changing in case of a double gripper.

(i) Tool magazine to index into the correct position where the tool from the spindle is to be placed.
(ii) Stopping the spindle at the correct orientation for the tool change arm to pick the tool from the spindle (Fig. 11.21a).
(iii) Tool-change arm to index to reach the tool magazine.
(iv) Tool-change arm to pick the tool from the spindle and the tool magazine simultaneously (Fig. 11.21b).
(v) Tool-change arm to index to reach the spindle (Fig. 11.21c).
(vi) New tool is placed in the spindle and the tool magazine (Fig. 11.21d).
(vii) Tool-change arm moves into its parking position.

The above sequence is shown schematically in Fig. 11.21.



(a) (b) (a) (d)

Fig. 11.21 *Tool-change procedure with a tool-change arm having a double gripper*

The system with a double gripper reduced the total number of events. Also, the actual time during which the machine needs to be stopped is also reduced considerably. Thus, most of the machine tools follow this type of arrangement. The only one consideration in such cases is that the old tool will be placed into the magazine pocket location, which is vacated by the new tool. Thus, it is necessary to keep track of the tools location. In the case of tools used with tool coding system described later, this should not be a problem.

Otherwise the MCU will have to keep track of the change in order to ensure that the correct tool offsets are used when that tool is used in another program later. Also, the magazine location has to be very close (in the same plane) to the spindle as shown in Fig. 11.20.

In case when the tool magazine position is not located in the same plane as that of the spindle, then the tool change arm needs to make another motion to move the tool from the magazine to the spindle and vice versa. Then the series of actions would be as follows.

(i) Tool magazine to index into the correct position where the tool from the spindle is to be placed.
(ii) Tool-change arm to index to reach the tool magazine.
(iii) Pick the new tool from the tool magazine.
(iv) Tool magazine indexes to the vacant position where the tool from the spindle is to be placed.
(v) Stopping the spindle at the correct orientation for the tool change arm to pick the tool from the spindle.
(vi) Tool-change arm to index to reach the tool magazine.
(vii) Tool-change arm to pick the tool from the spindle.
(viii) Tool-change arm to index to bring the new tool to the spindle.
(ix) New tool is placed in the spindle.
(x) Tool-change arm to index to reach the tool magazine.
(xi) Tool magazine indexes to bring the original pocket of the old tool to the tool change point.
(xii) Old tool is placed into the tool's original pocket in the tool magazine.
(xiii) Tool-change arm moves into its parking position.

Though there are more actions needed for tool change, many of these are done when the tool is actually doing the cutting action and therefore would not mean a loss of machine time. The machine tool is idle only during the operations (v) to (x). The main advantage in this sequence is that tool is always associated with the magazine position and hence is easier to track the tool offset data.

In the case of turning centres, it is generally the tool turrets that are used so the tool change is not a big problem. However, when the modular tooling is used, then it is necessary to use a tool magazine. There are generally two types of magazines which are used.

- Disk type
- Drum type

In the disk-type tool magazine, a disk has radial pockets where the turning tool modules can be inserted. A typical disk magazine is shown in Fig. 11.22a. The tool-change arm is like a two-finger gripper with an extension arm which can pull or push the tool module into the respective pocket as shown in Fig. 11.22b. For a very large storage requirement the drum (cylindrical) type will be more suitable. In this case, the tool pockets are arranged on the surface of a cylinder along the length direction as shown in Fig. 11.23.

11.6 WORK HOLDING

Work holding in CNC machine tools is more important since the conventional work-holding devices such as vices or chucks are rarely used except for very simple components. For complex shapes of the workpieces, it becomes necessary to use some special fixtures for quick set up of the workpieces. The modular fixturing systems, which are in vogue for conventional machine tools have been refined and are widely used for holding the workpieces in CNC machine tools.

(a) (b)

Fig. 11.22 *Disc-type tool magazine used in CNC turning centres with modular tooling along with a tool-change arm (Courtesy, Boehringer, Germany)*

Fig. 11.23 *Drum-type tool magazine used in CNC turning centres with modular tooling for storing large number of tools (Courtesy, Georg Fischer, Switzerland)*

Grid plates are generally used as one of the fixturing bases. The grid plates are provided with precisely drilled and tapped holes to facilitate the clamping operation as shown in Fig. 11.24. Since the holes on these grid plates are made at precise positions, the operator would know the exact location of the component depending upon where he is clamping. These grid plates can be permanently clamped on the machine-tool table if necessary. This can be very conveniently used together with the zero shift facility (G53 to 58) to clamp even multiple small components.

The grid plate can also come in the form of a cube with four parallel faces which can all be used for clamping multiple workpieces as shown in Fig. 11.25. This fixture in conjunction with a rotary table will allow it to be used as an indexing fixture for clamping more workpieces to the machine tool in a single fixture.

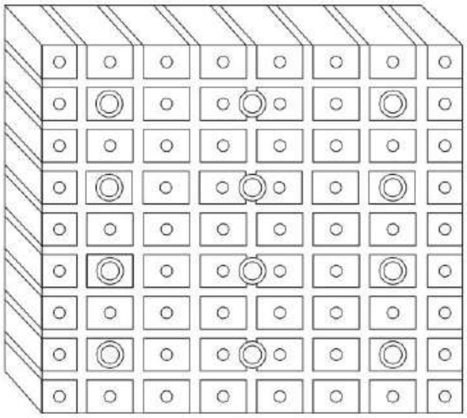


Fig. 11.24 *Grid plate with holes which can be used as a machine table*

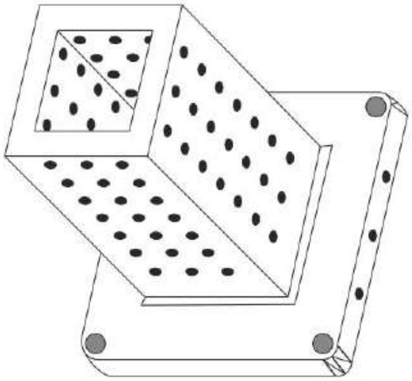


Fig. 11.25 *Tombstone for mounting multiple components on different places*

In addition to these standard fixture bases, a large number of fixture elements such as angle blocks and base elements (Fig. 11.26) are used to quickly clamp the workpieces in position.

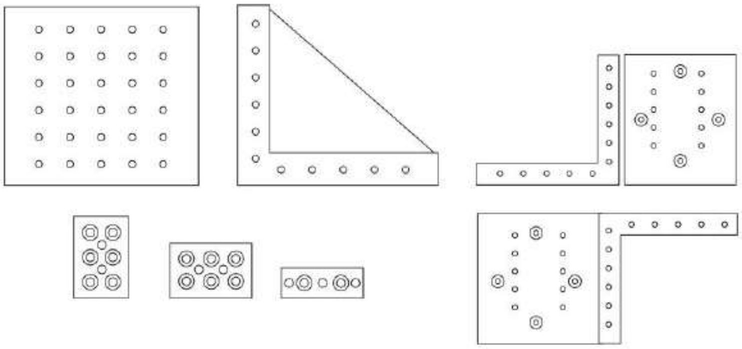


Fig. 11.26 *Modular fixture elements used for supporting complex workpieces*

11.7 CUTTING-PROCESS PARAMETER SELECTION

11.7.1 Milling-Time Estimation

Typical process parameters used in a milling operation are given in Table 11.3. The cutting speed in milling is the surface speed of the milling cutter. Thus

$$V = \frac{\pi DN}{1000}$$

where,
V = cutting speed (surface), m/min
D = diameter of the milling cutter, mm
N = rotational speed of the milling cutter, rpm

$$\text{Time for one pass} = \frac{l}{fZN} \text{ minutes}$$

where Z = number of teeth in the milling cutter
f = feed per tooth, mm
l = length of movement, mm

Table 11.3 *Data for milling*

Work material	*Hardness*	*HSS*		*Carbide*	
	BHN	*Speed m/min*	*Feed mm/tooth*	*Speed m/min*	*Feed mm/tooth*
C20 steel	110 – 160	20	0.13	90	0.18
C35 steel	120 – 180	25	0.13	80	0.18
C50 steel	160 – 200	20	0.13	60	0.18
Alloy steel	180 – 220	30	0.10	60	0.18
Alloy steel	220 – 300	18	0.08	90	0.18
Alloy steel	220 – 300	14	0.08	60	0.15
Alloy steel	300 – 400	14	0.05	60	0.13
Stainless steel	200 – 300	20	0.10	85	0.13
Cast iron	180 – 220	16	0.18	58	0.20
Malleable iron	160 – 240	27	0.15	85	0.18
Cast steel	140 – 200	16	0.15	50	0.18
Copper	120 – 160	38	0.15	180	0.15
Brass	120 – 180	75	0.28	240	0.25
Bronze	160 – 200	38	0.18	180	0.15
Aluminium	70 – 105	120	0.28	240	0.25
Magnesium	40 – 60	210	0.28	380	0.25

11.7.2 Drilling-Time Estimation

Typical process parameters used in a drilling operation are given in Table 11.4. The cutting speed in drilling is the surface speed of the twist drill. Thus

$$V = \frac{\pi DN}{1000}$$

where, V= cutting speed (surface), m/min
D= diameter of the twist drill, mm
N= rotational speed of the drill, rev/min

The drill will have to approach the start of the hole from a distance and also traverse beyond the actual hole by a distance termed as the total approach allowance, A. The initial approach is generally a small value for positioning the drill above the hole. This distance, A_I can generally be taken as 2 to 3 mm. The traverse distance beyond the hole is often termed as the breakthrough distance and is required because of the conical shape of the twist drill as shown in Fig. 11.27. This value is dependent upon the drill diameter and the lip angle and is given by

Table 11.4 *Cutting process parameters for drilling*

Work material	*Hardness*	*HSS*	
	BHN	*Speed m/min*	*Feed mm/rev*
Cast iron	200	25-35	0.13–0.30
Cast steel	280–300	12-15	0.06–0.19
AISI 1020	110–160	35	0.20–0.50
AISI 1040	170–200	25	0.13–0.30
Manganese steel	185–21	5	0.06–0.19
Nickel steel	200–240	18	0.06–0.19
Stainless steel	150	15	0.13–0.30
Spring steel	400	6	0.06–0.19
Tool steel	150	23	0.20–0.50
Tool steel	200	18	0.13–0.30
Tool steel	215	15	0.13–0.30
Tool steel	300	12	0.06–0.19
Tool steel	400	5	0.06–0.19
Malleable Iron	110–130	26	0.20–0.50
Aluminium	95	275	0.13–0.90
Aluminium alloys	170–190	18	0.13–0.30
Copper	80–85	21	0.06–0.19
Brass	190–200	70	0.20–0.50
Bronze	180–200	54	0.20–0.50
Zinc alloys	110–125	70	0.20–0.50
Glass		4.5	0.06–0.19

Breakthrough distance,

$$A = \frac{D}{2 \tan \alpha}$$

For the most common case of $\alpha = 59°$, it is given by

$$A = \frac{D}{3.3286}$$

Total length of tool travel,

$$L = l + A + 2 \text{ mm}$$

where

l = length of the hole, mm

$$\text{Time for drilling the hole} = \frac{L}{fN} \text{ minutes}$$

where

f = feed rate, mm/rev

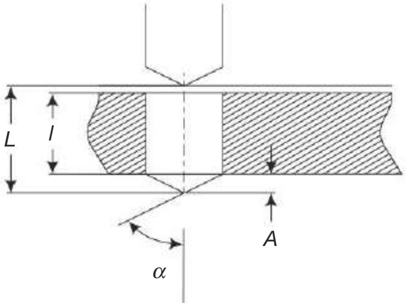


Fig. 11.27 *End of the twist drill showing the break through distance*

The material-removal rate is indicated by the total volume of the material in the hole. In the case of a solid material without coring, the Material-Removal Rate MRR is given by the area of cross-section of the hole, times the tool travel rate through the material. Thus,

$$\text{MRR} = \frac{\pi D^2 f N}{4}$$

Drilling Force Estimation The two major forces acting on the drill during the machining operation are the torque and the thrust. The torque acting on a twist drill is given by

$$M = C\, d^{1.9} f^{0.8} \text{ N mm}$$

where

d is the diameter of the drill in mm

f is the feed rate of the drill in mm/rev

C is a constant whose values are given in Table 11.5.

Table 11.5 *Constant C for torque calculations*

Material	*Hardness, BHN*	*Constant, C*
Steel	200	616
	300	795
	400	872
Aluminium alloys		180
Magnesium alloys		103
Brasses		359

The thrust force is given by

$$T = K\, d\, f^{0.7} \text{ Newtons}$$

The values of K are given as

Steel = 84.7

Cast iron = 60.5

The tolerance on the achieved dimension in drilling depends not only on the geometry of the drill but to a great extent, on the diameter of the drill also. Typical tolerances that can be achieved in drilling with a properly ground drills is given in Table 11.6.

11.7.3 Turning Time and Power Estimation

To estimate the machining times, it is necessary to select the proper process parameters. For this purpose, it is necessary to know the workpiece material and the cutting-tool material combinations to arrive at the right combination of the process parameters, cutting speed, feed and depth of cut. Their choice is somewhat difficult and a lot depends upon the shop practices as well as the experience of the operator/planner.

Some typical values of these parameters are given in Table 11.7 for the materials that are generally used. These should be considered as starting values and should be modified further based on the shop experience.

Turning The cutting speed in turning is the surface speed of the workpiece. Thus

$$V = \frac{\pi D N}{1000}$$

Table 11.6 *Limits of tolerance on drilling*

Diameter, mm	*Limits of tolerance, mm*	
	High (+)	*Low (–)*
Up to 3	0	0.014
3 to 6	0	0.018
6 to 10	0	0.022
10 to 18	0	0.027
18 to 30	0	0.033
30 to 50	0	0.039
50 to 80	0	0.046
80 to 120	0	0.054

Table 11.7 *Suggested cutting process parameters for turning*

Work material	*Hardness*	*High-speed steel tool*		*Carbide tool*	
	BHN	*Speed m/min*	*Feed mm/rev*	*Speed m/min*	*Feed mm/rev*
Grey cast iron	150 – 180	30	0.25	140	0.30
Grey cast iron	220 – 260	20	0.25	90	0.30
Malleable iron	160 – 220	33	0.25	50	0.25
Malleable iron	240 – 270	—	—	45	0.30
Cast steel	140 – 180	40	0.25	150	0.30
Cast steel	190 – 240	26	0.25	125	0.30
C20 steel	110 – 160	40	0.30	150	0.38
C40 steel	120 – 185	30	0.30	145	0.38
C80 steel	170 – 200	26	0.30	130	0.30
Alloy steel	150 – 240	30	0.25	110	0.38
Alloy steel	240 – 310	20	0.25	100	0.30
Alloy steel	315 – 370	15	0.25	85	0.25
Alloy steel	380 – 440	10	0.20	75	0.25
Alloy steel	450 – 500	8	0.20	55	0.25
Tool steel	150 – 200	18	0.25	70	0.25
Hot work die steel	160 – 220	25	0.25	120	0.25
Hot work die steel	340 – 375	15	0.25	75	0.25
Hot work die steel	515 – 560	5	0.20	23	0.20
Stainless steel	160 – 220	30	0.20	120	0.25
Stainless steel	300 – 350	14	0.20	70	0.25
Stainless steel	375 – 440	10	0.20	30	0.25
Aluminium alloys	70 – 105	210	0.30	400	0.38
Copper alloys	120 – 160	200	0.25	300	0.25
Copper alloys	165 – 180	85	0.25	230	0.25

where, V = cutting speed (surface), m/min

D = diameter of the workpiece, mm

N = rotational speed of the workpiece, rpm

The diameter, D to be used can be either the initial diameter of the blank or the final diameter of the workpiece after giving the depth of cut. However, there is practically not much change in the values obtained by using either of the values. To be realistic, the average of the two diameters would be better.

From the above equation, we get

$$N = \frac{1000\ V}{\pi D}$$

The time, t for a single pass is given by

$$t = \frac{L}{fN}$$

where

L = length of the travel, mm

f = feed rate, mm/rev

The number of passes required to machine a component depends upon the left over stock (stock allowance). Also depending upon the specified surface finish and the tolerance on a given dimension, the choice would have to be made as to the number of finishing passes (1 or 2) while the rest of the allowance is to be removed through the roughing passes. The roughing passes, P_r is given by

$$P_r = \frac{A - A_f}{d_r}$$

where A = total machining allowance, mm

A_f = finish machining allowance, mm

d_r = depth of cut in roughing, mm

The value calculated from the above equation is to be rounded to the next integer.

Similarly the finishing passes, P_f is given by

$$P_f = \frac{A_f}{d_f}$$

where d_f = Depth of cut in finishing, mm

Power Required in Turning The power required at the spindle for turning depends upon the cutting speed, depth of cut, feed rate and the workpiece material hardness and machinability. The power required depends upon the cutting force, which is a function of feed rate, f and depth of cut, d. However, for the sake of gross estimation it can be safely assumed that

cutting force, $F = K \times d \times f$

where K is a constant depending on the work material which is given in Table 11.8.

Then power,

$$P = F \times V$$

Combining the above two equations

Power,

$$P = K \times d \times f \times V$$

Table 11.8 *Constant K for power calculation*

Material being cut	*K (N/mm^2)*
Steel, 100 – 150 BHN	1200
Steel, 150 – 200 BHN	1600
Steel, 200 – 300 BHN	2400
Steel, 300 – 400 BHN	3000
Cast iron	900
Brass	1250
Bronze	1750
Aluminium	700

Summary

- Tooling plays an important part in CNC machining.
- A variety of cutting-tool materials are used in CNC machining. High Speed Steel (HSS) with coatings is used for small size cutting tools. HSS tools are often used with coatings to improve the tool life. Carbides and coated carbides are the largest percentage of cutting tools used for large volume manufacture. Ceramics are sometimes used for specific applications such as for machining cast irons.
- Turning tool holders and tool bits have standard geometry specified by the ISO. The tool geometry will have to be carefully selected to suit the geometry of the part being machined.
- Milling tooling system will have to accommodate the type of adopter to be used in conjunction with the automatic tool changing system used.
- Tool presetting becomes an important element for tool management systems in CNC manufacturing. Many tool presetters are available that can be used for offline tool setting for machining centers as well as turning centres.
- Different types of tool magazines are used for automatic tool changing. Drum and chain type tool magazines are more common for holding large number of tools.
- Automatic tool changers with double grippers reduce the tool changing time by doing part of the tool-changing function during the actual machining operation.
- Modular work-holding elements are convenient for quick work holding fixture construction.
- Machining power and time can be computed using the tool material and work material requirements for various machining operations.

Questions

1. What are the types of cutting-tool materials used in CNC machine tools?
2. How do you select the carbide grade for a given machining application?
3. Write a small note on coatings used in cutting tools.
4. What are the requirements for using ceramics in machining?
5. Briefly explain the ISO coding system used carbide-tool bits.
6. Briefly explain the ISO coding system used carbide-turning tool holders.

7. What are the advantages of modular tooling units in CNC turning?
8. What are the requirements of tool pre-setting in CNC machining?
9. Give a brief description of any one of the tool presetter you are familiar with.
10. What are the various types of tool magazines used in CNC machine tools? Give their relative merits.
11. How does a tool change occur in a typical automatic tool changer?
12. What are the work-holding methods suitable for CNC machining centres.

12

CNC MACHINE TOOLS AND CONTROL SYSTEMS

Objectives

A large variety of CNC machine tools have been developed over the years to cater to the wide spectrum of machining activities in the industry. In order to select an appropriate machine tool for a given application, it is necessary to understand the range of facilities in the CNC machine tools available in the market. After completing the study of this chapter, the reader should be able to

- Understand the type of CNC machining centres and their capabilities for large-volume production as well as flexibility to cater to a wide variety of geometries
- Discuss about the varieties of CNC turning centres and the range of facilities that are available with them to cater to the different part families that are either axisymmetrical or axisymmetrical with some non-axisymmetrical features
- Appreciate the need for high-speed machining and the range of high-speed machine tools that are catering to that need
- Discuss the organisation of various software modules in a machine control unit that perform all the functions required of them
- Learn about the various support systems available for the CNC machine tools
- Understand about the touch-trigger probes and their functions in connection with dimension measurement as well as tool offset determination

12.1 CNC MACHINING CENTRES

In the beginning, the successful tool-room milling machines were converted to CNC by simply replacing the motion elements by automated devices. However, that could not exploit fully the potential of the CNC. Hence, the CNC machine tools were redesigned with greater emphasis on the structural rigidity, power available as well as the ability to perform a variety of functions. Also, most of the modern CNC milling machines have expanded machining capabilities by the addition of accessory devices, making them more versatile. That is why these are now called *machining centres*, rather than milling machines. Some even like to call them *manufacturing centres* since many of them have other capabilities such as inspection abilities built into them, which is described later in this chapter.

The CNC machining centres can be broadly categorised into two varieties:

- Vertical-axis machining centres, and
- Horizontal-axis machining centres.

12.1.1 Vertical-Axis Machining Centres

The vertical-axis machining centres, or VMC as is popularly abbreviated, are generally more versatile in terms of the tool being able to generate more complex surfaces compared to the horizontal axis. Most of the early CNC machine tools therefore are of this category. Figure 9.11 shows the Bridgeport CNC milling machine which is more like a tool-room milling machine showing its roots. However, the subsequent developments have improvised the machines in such a way that the current machines have a much rigid construction as evidenced in Fig. 12.1. Comparative specifications of the CNC vertical machining centres are given in Table 12.1 to appreciate the range of facilities incorporated in them.

Fig. 12.1 *Present-day production vertical axis CNC machining centre Bridgeport VMC 760 (Courtesy, Bridgeport Machines Inc., Bridgeport, USA)*

It is also that this category of machine tools are generally provided with a variety of accessories to cater to a larger spectrum of jobs that can be completed by them in a single set-up. These are mostly useful in the tooling industry for the machining of dies and moulds.

Most of the general machines come with 3 axes. However, there are some versions, which come with more than 3 axes. For example, the spindle head can be swivelled in one or two axes (about X or Y axis) to provide A or B axis motion. These are required for machining sculptured surfaces. In the case of sculptured surface machining, the ball-nose end-mill surface needs to be maintained tangential to the part surface in order to see that no gouging or undercutting takes place. A typical 5-axis CNC machining operation is shown in Fig. 12.2.

Table 12.1 *Typical specifications of some vertical machining centres*

	Bridgeport EZ-TRAK DX	*Cincinnati Milacron Arrow 500 VMC*	*Chiron FZ22S*	*Makino GF6*
X-axis	813 mm	20″ (510 mm)	750 mm	1050 mm
Y-axis	368 mm	20″ (510 mm)	520 mm	600 mm
Z-axis	127 mm	20″ (510 mm)	630 mm	560 mm
Work surface	1473 × 279 mm	27.5 × 20.5″ (700 mm × 520 mm)	1150 × 570 mm	1400 × 600 mm
Max. workload	272 kg	1103 lb (500 kg)	1200 kg	1500 kg
Rapid traverse *X-Y-Z*	2540 mm/min	944 ipm (24 m/min)	30 m/min	12 m/min
Feed Rate (Max.) *X-Y-Z*	2540 mm/min	590 ipm (15 m/min)	30 m/min	12 m/min
Spindle taper	No. 30	No. 40	ISO 40	No. 50
Speed range	60–4200 rpm	60–6000 rpm	20–10 500 rpm	10–4000 rpm
Power (Max.)	1.5 kW	17.5 hp (13 kW)	18 kW	7.5 kW
Tool changer capacity		21 tools	40 tools	20 tools
Max. tool weight		15 lb (6.8 kg)	8 kg	15 kg
Max. tool dia. (Full storage)		3.14″ (80 mm)	75 mm	145 mm
Max. tool dia. (Alt. storage)		6.3″ (160 mm)	105 mm	200 mm
Max. tool length		15″ (385 mm)	250 mm	400 mm
Tool change time (metal-to-metal)		7 seconds	3.5 seconds	5.5 seconds
Accuracy positioning (*X*, *Y*)	± 25 microns	± 0.00011″ (± 3 microns)		± 1.5 microns
Positioning (*Z*)	± 25 microns	± 0.00016″ (± 4 microns)		± 1.5 microns
Repeatability	± 20 microns	± 0.00004″ (± 1 micron)		± 1 micron
Dynamic contouring		± 0.0006″ (15 microns)		

The variations that can be found in the vertical machining centre category are

- Travelling column
- Gantry structure
- Multiple spindle

A typical travelling-column machining centre is shown in Fig. 12.3. Here, the tool head is positioned on a cross-rail supported by two heavy travelling columns by the side of the table. This type of construction allows for the tool head to be very well supported. Hence, they can take heavy cuts with very long table travels. These are normally used for very long workpieces such as those used in the aerospace industry for milling the aircraft structural components. By virtue of the tool head being mounted on a cross-rail, it is also possible to provide the necessary swivelling motions to it for possible multiple-axis machining as shown in Fig. 12.4. The actual machine tool machining an aerospace component is shown in Fig. 12.5.

The above concept could be minimised in a gantry-type construction for supporting the spindle in the case of a vertical machining centre. Such a machine tool is shown in Fig. 12.6. This allows for the workpiece to remain stationary and be machined by the moving cutting tool. The two columns are provided with synchronised drives for *Y*-axis movement to reduce any unequal moments. This machine is capable of achieving rapid traverses of the order of 60 m/min with the spindle providing a maximum of 18 000 rpm.

For large batch production, generally multi-spindle machines are useful. These may have 2 to 4 spindles with each spindle carrying out identical machining operations. Thus, it is possible to obtain a production of 2 to 4 components per cycle depending upon the number of spindles present. A typical machine tool with 4 spindles is shown in Fig. 12.7, though only 2 are visible in the view. It also has another double spindle head, like the one shown, to make the total spindles number up to 4. It is possible to use either 1, 2 or 4 spindles simultaneously.

Fig. 12.2 *A 5-axis machining operation (Courtesy, Yamazaki Mazak Corp., Japan)*

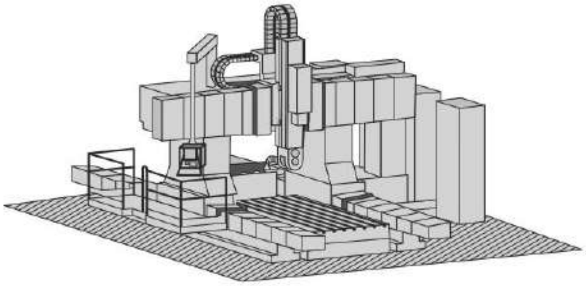


Fig. 12.3 *A schematic representation of a CNC travelling-column machining centre (Courtesy, Jobs S.p.A., Italy)*

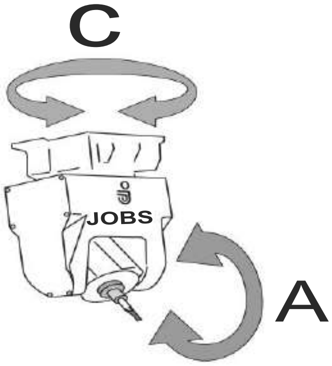


Fig. 12.4 *The possible rotary motions of the tool head of a CNC travelling-column machining centre (Courtesy, Jobs S.p.A., Italy)*

Because of the layout, the space available for an individual component will be limited, but not too small. These machines are provided with special work-holding accessories (workpiece carrier) with rotary motions to improve the productivity. The pallet changer incorporated allows for the setting up of the workpiece when the machine is carrying out the machining action.

Fig. 12.5 *A 5-axis CNC travelling-column machining centre for aerospace machining applications (Courtesy, Jobs S.p.A., Italy)*

Fig. 12.6 *A gantry-type CNC machining centre DMC 65V for high-speed machining (Courtesy, Deckel Maho Gildemeister, Germany)*

As discussed in Chapter 11, the ATC is the most important element in most of the machining centres which works towards reducing the idle time involved. The VMC allows for varied ATC placement with respect to the spindle. The placement of the ATC close to the spindle allows for a very fast chip-to-chip time, which means the time between the tool disengaging with the workpiece to its engagement again. For example, the Chiron machining centre as shown in Fig. 12.8 places the ATC right above the spindle. It also indexes it about the spindle axis as shown in Fig. 12.8. As a result, the chip- to-chip time quoted is 1.9 seconds.

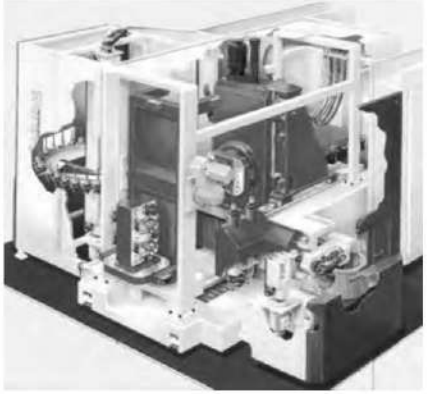


Fig. 12.7 *A 4-spindle CNC machining centre 4CUT for high-speed machining (Courtesy, Hüller Hille Gmbh, Germany)*

Fig. 12.8 *Chiron CNC machining centre FZ 12W for high-speed machining with a very fast tool-change time (Courtesy, Chiron Werke Gmbh, Germany)*

12.1.2 Horizontal-Axis Machining Centre

A typical horizontal-axis machining centre is shown in Fig. 9.12. By its very configuration, the horizontal axis machining centre, popularly called HMC, is sturdier than the vertical configuration and hence is used for heavier workpieces with large metal-removal rates.

Since these machines provide for heavier metal-removal rates, the cutting tools used are normally big. As a result, the tool magazine has to provide for larger place for each tool. This results in the tool magazines for HMC being heavier. Also, they are normally provided with tool magazines having higher capacity. Comparative specifications of horizontal machining centres are given in Table 12.2.

Table 12.2 *Typical specifications of some horizontal machining centres*

	Mikron HCE 400	*Cincinnati Milacron HMC-400EP*	*Yamazaki H500*	*Makino A55*
X-axis	610 mm	22.4″ (570 mm)	720 mm	560 mm
Y-axis	508 mm	20.3″ (515 mm)	650 mm	560 mm
Z-axis	559 mm	20.5″ (520 mm)	650 mm	560 mm
Pallet working area		400 × 400 mm	500 × 500 mm	400 × 400 mm
Max. workload	360 kg	882 lb. (400 kg)	1000 kg	400 kg
Rapid traverse *X–Y–Z*	18 m/min	945 ipm (24 m/min)	24 m/min	30 m/min
Feed rate (max.) *X–Y–Z*	12.7 m/min	0.1–945 ipm (2.5–24,000 mm/min)	8 m/min	30 m/min
Spindle taper	No. 40	No. 40	ISO 50	No. 40
Speed range	7500 rpm	75–8000 rpm	35–6000 rpm	50–12 000 rpm
Power (Max.)	11 kW	15 hp (11 kW)	15 kW	22 kW
Tool-changer capacity	24 tools	30 tools	40 tools	40 (60, 128, 208) tools
Max. tool weight		15.4 lb (7 kg)	27 kg	8 kg
Max. tool dia. (full storage)		2.95″ (75 mm)	135 mm	70 mm
Max. tool dia. (alt. storage)		3.94″ (100 mm)		140 mm
Max. tool length		11.8″ (300 mm)	470 mm	300 mm
Tool-change time (metal- to-metal)		5 seconds	6.5 seconds	3.4 seconds
Accuracy positioning (*X*, *Y*)		± 0.0002″ (± 5 microns)		± 2 microns
Positioning (*Z*)		± 0.0002″ (± 5 microns)		± 2 microns
Repeatability		± 0.0006″ (± 1.5 microns)		± 1 micron
Dynamic contouring		± 0.0006″ (15 microns)		

The rotary table is one of the most common accessory used with the HMC. This provides a fourth axis in the form of *B*. Since HMC is used for machining the prismatic (boxlike) components, the availability of a rotary table makes it feasible for the machining of all four faces of the component that are facing the tool as shown in Fig. 12.9. Thus, a large part of the component can be finished in a single set-up allowing for closer tolerances for the critical components.

Further, some horizontal machining centres are provided with an additional facility for swivelling the entire spindle by the use of a G code such that the horizontal axis of the spindle becomes vertical. This allows for a fifth face to be machined in addition to the four faces machined with the rotary table as shown in Fig. 12.10.

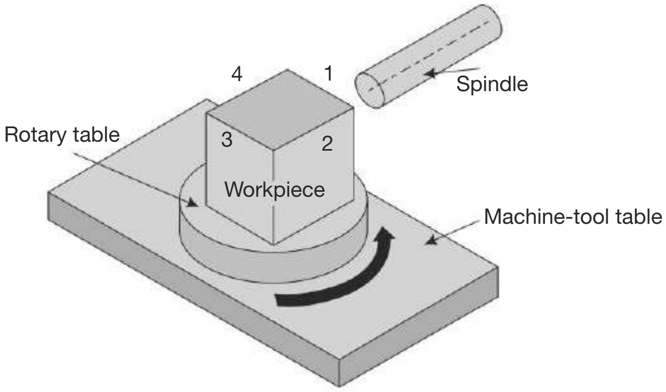


Fig. 12.9 *Rotary table used in HMC for machining all four faces*

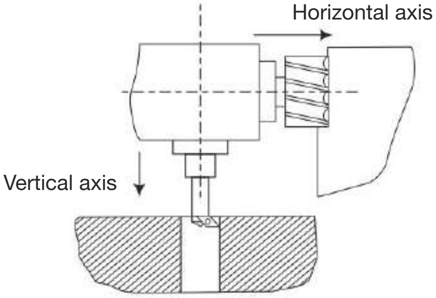


Fig. 12.10 *Spindle swivelling facility in HMC for machining in two different planes (XY as well as XZ)*

The rotary table can also have more- than- one-axis rotation capability. For example, the rotary table can have two rotary motions in *A* and *B* axes in place of the conventional *C*-axis as shown in Fig. 12.11. If such a rotary table is interfaced with a conventional 3-axes horizontal machining centre (or a VMC) then it will be possible to machine complex sculptured surfaces. Typical planes that can be obtained by such a rotary table are shown in Fig. 12.12.

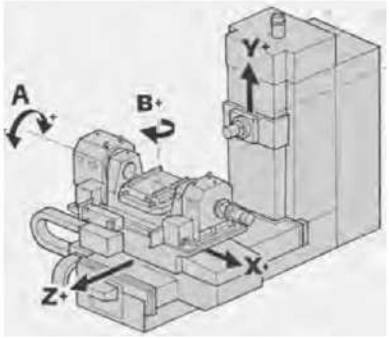


Fig. 12.11 *Horizontal-axis machining centre provided with a 2-axis rotary table (Courtesy, Dixi Machines SA, Switzerland)*

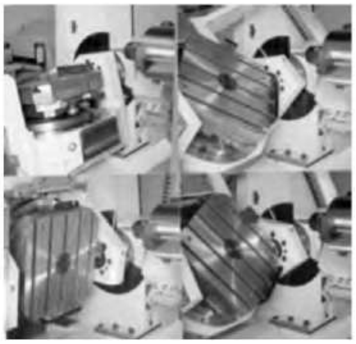
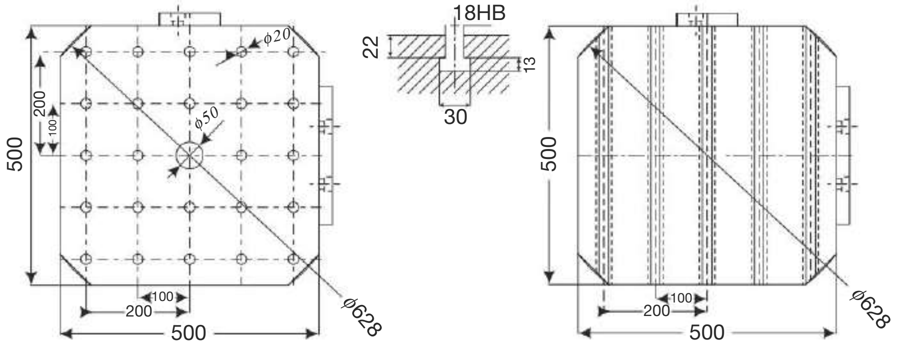


Fig. 12.12 *Various planes possible by the 2-axis rotary table with an HMC (Courtesy, Yamazaki Mazak Corp., Japan)*

Pallet Changer Generally, the components machined on HMC or VMC require large cycle times. Further complex jobs used on these machines require relatively complex set-up times. Thus, the actual setting up of the workpieces need to be done away from the machine tool such that the machine utilisation is improved. For this purpose, pallets are most commonly used with the machining centre. A pallet may be considered as a small table having standardised dimensions as shown in Fig. 12.13. These are normally available in standard sizes such as 500 × 500 mm, 630 × 630 mm, etc. They come in generally two forms, one with precision-drilled holes (Fig. 12.13a) while the other with T-slots (Fig. 12.13b). The blank components can be clamped, set-up and unclamped on these pallets at a location away from the machine spindle thus, not disturbing the machining function.



(a) (b)

Fig. 12.13 *Typical pallet designs used with HMC: (a) Pallet with holes (b) pallet with T-slots*

A typical machine tool with an Automatic Pallet Changer (APC) is shown in Fig. 12.14. The pallet changer shown in Fig. 12.14 consists of 3 pallet locations with two pallets. One pallet location is near the spindle (*A*) where the actual machining takes place, while the other two locations *B* and *C* are meant for workpiece set-up, loading and unloading. When the machining of the component on the pallet at position *A* is completed, the table moves into position *B* or *C* which is empty. Then the shuttle mechanism allows the pallet to be moved into the empty position. Then the table moves into the other location where the pallet with the workpiece is already set up for machining. The pallet then moves onto the table. The table then moves into the machining position at *A*. The operator can then proceed with the unclamping of the finished workpiece from the pallet and clamp a new workpiece to be machined.

Another form of pallet changers that are found are the rotary type as shown in Fig. 12.15. As shown, the pallets are arranged in rotary positions, out of which, one position is closer to the spindle. As the machining is completed, the rotary indexer indexes such that the next pallet with the unmachined component in the line reaches the spindle while the pallet with the machined component moves into the setting position. This is a relatively simple arrangement. It is also possible to have a larger number of pallets in a rotary carousal with a twin pallet shuttle for interchange between the table and the carousal as shown in Fig. 12.16. The pallet carousals can be linear or rotary with a larger number of pallets for long hours of unattended operation of the machines. Such systems are normally used with flexible manufacturing systems which are discussed later.

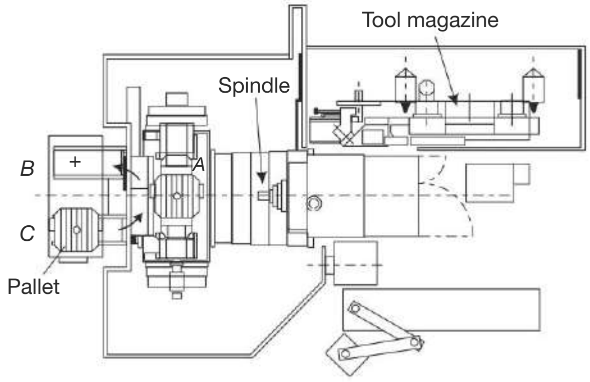


Fig. 12.14 *A typical horizontal machining centre with an automatic pallet changer*

12.2 CNC TURNING CENTRES

The majority of the components machined in the industry are of the cylindrical shape and, as such, the CNC lathes are more appropriately called *turning centres* and are also important machine tools. The evolution of the CNC turning centres follows that of the developments in the CNC machining centres closely.

The major change to be noticed in the turning centres is the early adoption of the slant bed (Fig. 12.17) to allow for a better view of the machining plane as well as for easy placement of the various devices involved in the machining zone. Most of the turning centres are also provided with a tool turret (Fig. 11.5) which may have a capacity of 8 to 12 tools of various types.

The major categories of CNC turning centres can be classified as under:

(a) Turnmill centres (X, Z, C)
(b) Multiple-axis turning centres (X, Z, C, Y)
(c) Vertical turning centres
(d) Twin-turret turning centres
(e) Multiple-spindle turning centres
(f) Integrated material handling

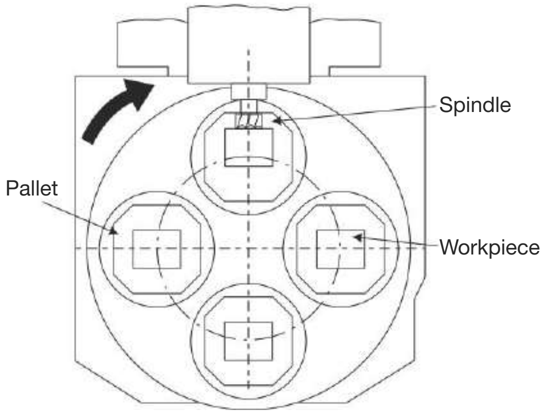


Fig. 12.15 *A typical pallet changer with a rotary-style pallet changer*

12.2.1 Turnmill Centres (X, Z, C)

The major development in the CNC turning centres is the development of the turnmill centre. Many turning components are not completely machined in lathes alone. After finishing the lathe operations, it may have to go to a drilling or milling machine to carry out operations such as drilling cross-holes, milling of flats, keyways, slots, etc. For example, to mill a keyway or drill a hole off-centre as shown in Fig. 12.18 is not

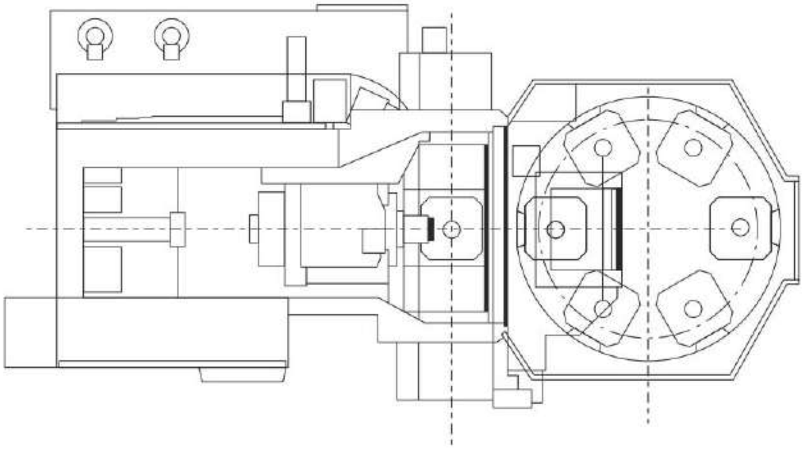


Fig. 12.16 *A typical pallet changer with a shuttle pallet changer combined with a 6-pallet carousal*

possible in a normal turning centre. The use of multiple machine tools calls for multiple set-ups, which decrease the overall tolerances. Also, they cause delays in scheduling the jobs on different machine tools, transportation, storage, etc. Many of these features to be machined are relatively simple and can be completed in the turning centres itself in a single set-up, if there is a milling attachment available for them.

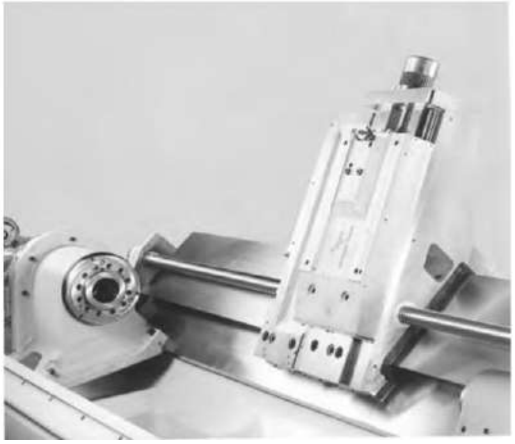


Fig. 12.17 *The slant bed of a CNC turning centre (Courtesy, Boehringer werkzeugmaschinen GmbH, Germany)*

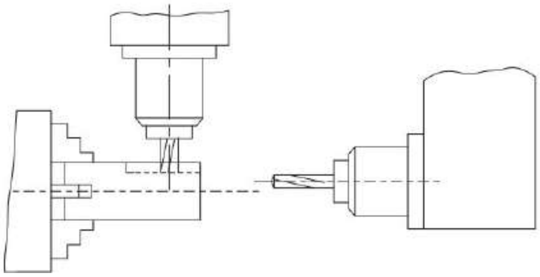


Fig. 12.18 *Machining of a keyway or drill a hole away from the centre of the workpiece*

This calls for the development of a combined machine tool, which can perform the turning as well as milling operations in the same machine tool. Such a machine tool is called the CNC Turnmill Centre. In order to do the milling operations in a lathe, the main spindle will not rotate. Instead, the tool will rotate in the tool turret. For this purpose, a separate drive is provided in the turret, so that the tools such as twist drills and end mills rotate in the tool turret as shown in Fig. 12.19. Such tools are called *driven tooling*. The main spindle holding the workpiece can therefore be indexed suitably to get the necessary profiles. This is called the *C*-axis. A typical tool turret with the driven tooling is shown in Fig. 12.20.

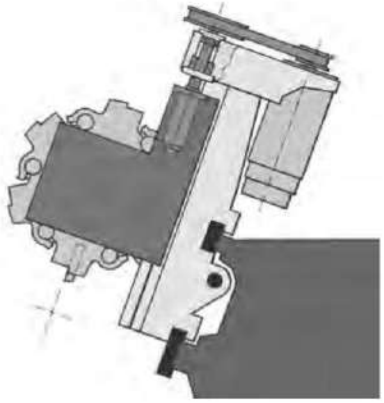
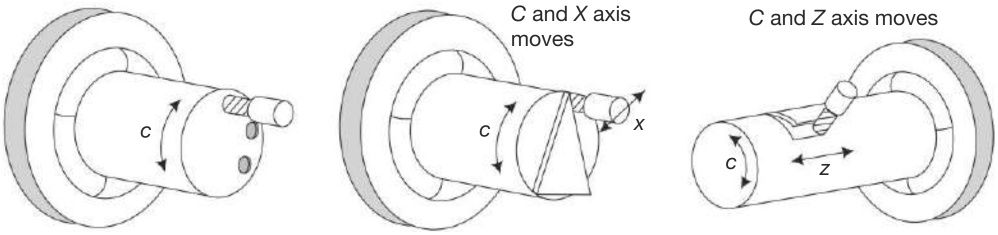


Fig. 12.19 *The drive for tooling in the turret of a CNC turnmill centre (Courtesy, George Fischer, Switzerland)*

Fig. 12.20 *A tool turret with driven tooling to be used in a CNC turnmill centre (Courtesy, George Fischer, Switzerland)*

The driven tooling can therefore move in addition to the *X* and *Z* directions like the normal turning centre, and can also move in the *C*-axis. The combination of these 3 axes can make the machine tool really versatile and allows for a range of complex surfaces to be produced. Typical surfaces that can be machined by these machines are shown in Fig. 12.21.



(a) Only *C*-axis (b) Combined *C* and *X* axis (c) Combined *C* and *Z* axis

Fig. 12.21 *Typical shapes of components that can be machined by the combination of X, Z and C-axis movements*

12.2.2 Multiple-Axis Turning Centres

There are a large number of variations to be found in this category of machine tools where a large number of axes are provided in addition to the 2 or 3 axes machines that were seen so far.

Twin-Turret Turning Centres One of the possibilities is providing two turrets in a turning centre in place of 1 in the normal turning centre as shown in Fig. 12.22. This makes it a 4-axes turning centre with both turrets capable of moving independently in two axes each. The main advantage to be gained in such cases is that machining can be done by two tools as well as having a larger tool capacity. Since two tools will be machining simultaneously, a large amount of chips would be generated which needs to be carefully disposed off in such cases.

Multiple-Spindle Turning Centres For large-volume production of small and medium sized parts with standard tolerances, it may be necessary to go for multiple-spindle turning centres. One such machine is shown in Fig. 12.23. In this, the headstock consists of two spindles and a tailstock with two quills, which are arranged in parallel to each other at a fixed distance. Two turrets are provided to provide the tools for these two spindles. These machines, in view of the capability of simultaneous machining of two parts, are capable of high productivity, which is normally demanded in the automotive sector.

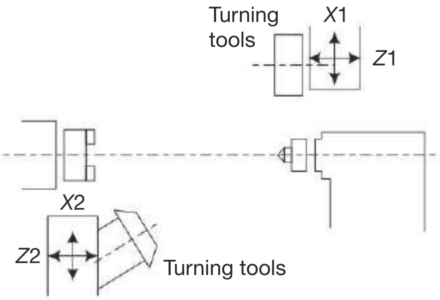


Fig. 12.22 *CNC turning centre with twin turrets*

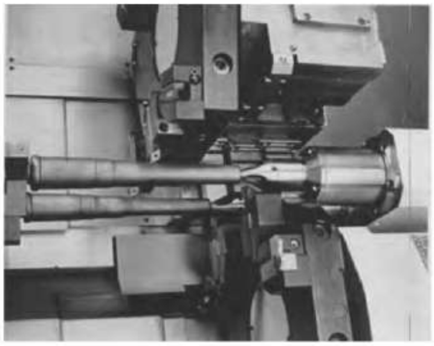


Fig. 12.23 *Twin-spindle CNC turning centre (Courtesy, Georg Fischer, Switzerland)*

Another twin-spindle turning centre is shown in Fig. 12.24. The twin spindles and turrets are separately controlled in this case also, except that the spindles in this case are maintained in line such that it is possible to carry out the machining of a component which requires two set-ups.

The machining of the first set-up is completed in the left spindle. Then the workpiece is transferred from the left spindle to the right spindle to realise automatic continuous machining operations. Then the workpiece can be completely machined in the right spindle. This allows for higher productivity like the twin-spindle turning centre described earlier, but also provides for complete machining when two set-ups are required for completing the job.

Fig. 12.24 *Twin-spindle CNC turning centre Mazak Dual Turn 20 (Courtesy, Yamazaki Mazak Corp., Japan)*

Still another class of machine tools found in this category are those machine which have the *Y*-axis machining capability in addition to the *X*, *Z* and *C*-axes. One such machine tool, Yamazaki Integrex 30, is shown in Fig. 12.25. This is almost like a full combination of a turning centre and a machining centre. The separate spindle provided above the turret is capable of moving in all the 3-axes like a machining centre to completely machine any profile as shown in Fig. 12.26a. In addition, if this capability is combined with the *C*-axis, more complex surfaces as shown can be machined as shown in Fig. 12.26b. The *Y*-axis turret is also provided with a tool magazine similar to that used in machining centres.

Fig. 12.25 *CNC turnmill centre with Y-axis, Yamazaki Integrex 30 (Courtesy, Yamazaki mazak Corp., Japan)*

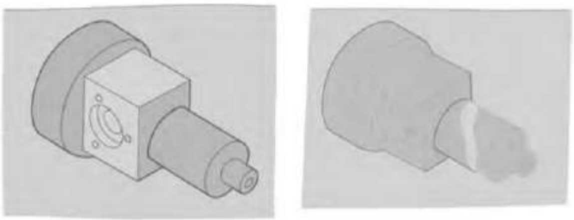


Fig. 12.26 *The types of machining that can be done using a CNC turnmill centre with Y-axis (Courtesy, Yamazaki mazak Corp., Japan)*

It is possible to further configure turning centres by having a number of combinations with larger number of axes by combining them in any order of the following options in a modular manner:

- Turret with turning tools
- Turret with driven tools
- Turret with *Y*-axis movement
- Programmable tailstock
- Tool changer for the *Y*-axis turret
- *C*-axis movement

12.2.3 Vertical Turning Centres

For very large-diameter workpieces, the turning centre with a horizontal spindle becomes difficult. Hence, in such cases a vertical turning centre as shown in Fig. 12.27 is used. The vertical turning centre generally has the spindle in the vertical direction in place like the table of a machining centre. The heavy and large workpieces can then be clamped on to the chuck, which is in the horizontal plane. The tool turret is kept in a plane above the spindle. For quick changing of the jobs, the automatic pallet changer which replaces the chuck in a manner very similar to the shuttle-type pallet changer in a machining centre is shown in Fig. 12.27.

Another concept in vertical turning centres is to make the spindle above and the tool turret below by inverting the conventional vertical turning centre. This is generally used for smaller workpieces compared to the conventional vertical turning centre. As shown in Fig. 12.28, the workpiece is held in the chuck and rotated by the vertical spindle above, while the tool turret lies below.

The vertical turning centre useful for large-volume production of small automobile components is the pick-up turning centre. This has two vertical spindles, one of which is movable (Fig. 12.29). This movable spindle acts like a tool-change arm, except in this case it is transferring the workpiece. The movable spindle picks up the workpiece from the conveyor and moves into the machining area. After completing the initial machining in the movable spindle, it directly transfers the semi-finished component to the other spindle's working area. The second set-up machining will be completed in the stationary spindle. After the complete machining, a gripper in the second turret removes the part from the spindles and places on the conveyor belt. With such a machine design, separate loading and turnover systems are not required. This provides a complete machining line for complete machining of components with two set-ups from the two ends.

Fig. 12.27 *CNC vertical turning centre, Yamazaki (Courtesy, Yamazaki Mazak Corp., Japan)*

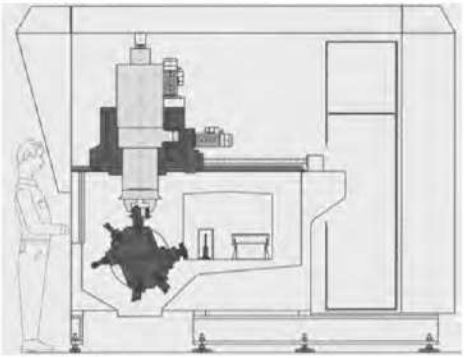


Fig. 12.28 *CNC vertical turning centre with inverted arrangement EMAG VSC 250HDS, (Courtesy, EMAG Maschinenfabrik, Germany)*

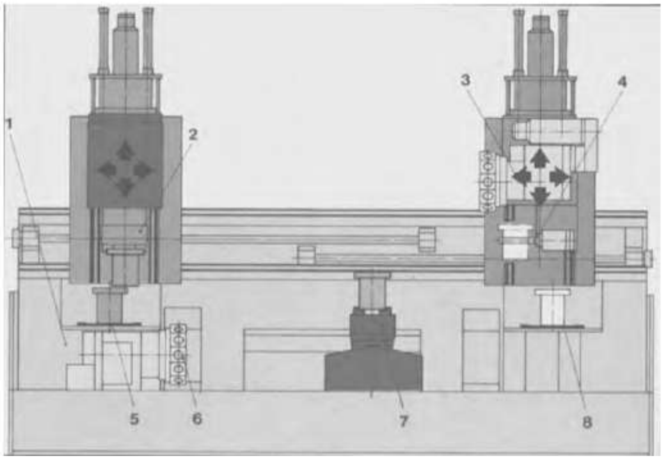


Fig. 12.29 *CNC vertical pick-up turning centre Hüller Hille DVTH300, 1-DV-Transfer, 2-movable spindle, 3-turret head on cross-slide, 4-unloading gripper on cross-slide, 5-workpiece supply, 6-stationary turret head, 7-stationary spindle, 8-workpiece removal (Courtesy, Hüller Hille, Germany)*

These machines are further developed by EMAG to what it calls as ‘Multi-Functional Vertical production Centres’ shown in Fig. 12.30 which can do any of the following operations:

- Turning
- Drilling
- Milling
- Grinding
- Gear cutting
- Balancing
- Gauging

The machine is designed to be fully flexible to cater to the varying machining needs of the automotive components. These machines can be equipped with two work spindles to machine two workpieces simultaneously—the TWIN (Fig. 12.31a) for identical operations, the DUO for two-side operations—or with two turrets or several drilling/boring/milling spindles to machine a component with two or more tools simultaneously. For machining complicated contours or for components requiring extensive milling operations, the spindle slide unit can be equipped with a *Y*-axis offering full 3D capability, and a pivoting axis *B* be incorporated in the tool carrier.

These machines can be provided with specific tooling to suit the individual components. Some of the typical options available are

- Disc-type turret for stationary turning and driven drilling and milling tools
- Heavy-duty drilling turret; can also be equipped with turning tools

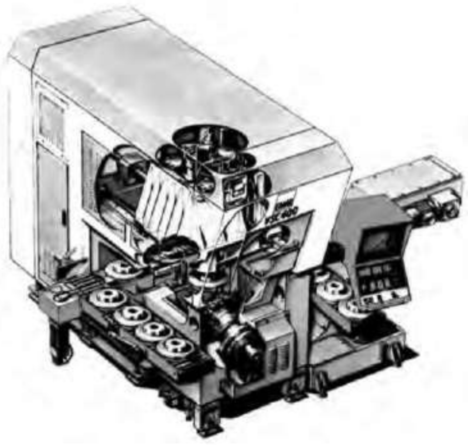
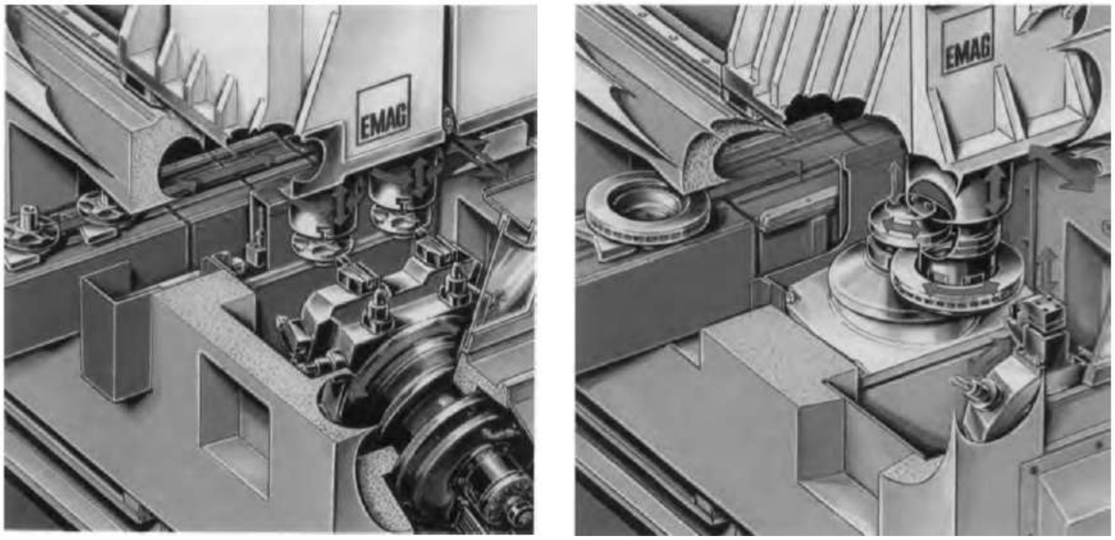


Fig. 12.30 *Multi-functional production centre EMAG VSC 400 (Courtesy, EMAG Maschinenfabrik, Germany)*

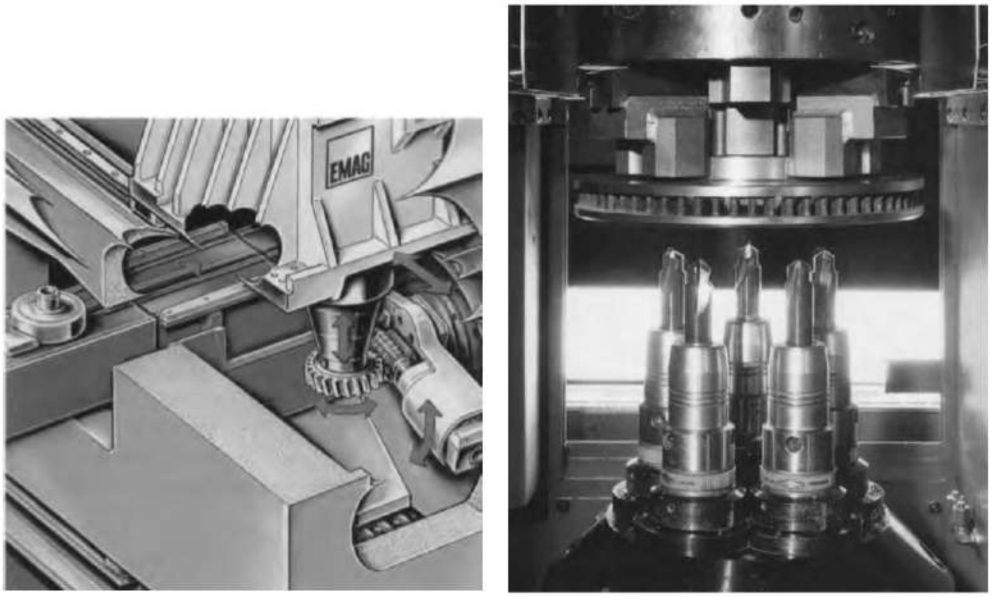


(a) Twin spindles to machine two workpieces simultaneously

(b) Turning and griding in the same set-up

Fig. 12.31 *Some options for the multi-functional production centre EMAG VSC 400, (Courtesy, EMAG Maschinenfabrik, Germany)*

- CNC heavy-duty drilling turrets with *Y*-axis for double drilling
- Combination of the standard disc-type turret for stationary turning tools and a high-performance milling spindle



(a) High-speed gear hobbing without coolant

(b) Multiple-spindle drilling head

Fig. 12.32 *Some options for the multi-functional production centre EMAG VSC 400, (Courtesy, EMAG Maschinenfabrik, Germany)*

- Workpiece specific multi-spindle drilling heads (Fig. 12.32b)
- HSC (High-Speed Cutting)—high-speed drilling and milling spindles for process and workpiece specific machining
- Automatic tool changer for drilling and milling tools
- Grinding spindles and turning tools in one turret for turning and grinding especially of hardened components in one clamping set-up (Fig. 12.31b)

In view of such a large variety of options available for these multi-functional production centres, they are highly adaptable for the complete machining of any of the automobile components such as brake drums, disc-brake rotors, wheel hubs, differential gear housings, CV joints, pistons, transmission gears, gearbox parts, parts for airbags, etc.

12.3 HIGH-SPEED MACHINE TOOLS

High-Speed Machining (HSM) is currently being considered as one of the main areas of work in the manufacturing sector. It is not new and work in the area has been going on for the last 30 years or so. However, the widespread interest in the technology is considered more now because of the maturity of the various technologies involved, namely, the computers, cutting tools, machine tools, CNC controllers and the CAM systems. The major applications that are pushing the technology towards high-speed machining are mould and die manufacture, aerospace component machining and automotive component manufacture. Each of these segments of industries has different goals as well as reasons for adopting high-speed machining.

A generic definition of high-speed machining is difficult. Machining speed is very application specific. High speed is particularly a relative term since what is high speed for high-speed steel may be low for carbides, and the same continues with ceramics as well under the normal conditions. Hence it is difficult to give a definitive value to high speed. It is, therefore, better to say that high-speed machining means cutting metal faster than is customary for the particular operation in consideration.

High-speed machining occurs when the tooth-passing frequency approaches the dominant natural frequency of the system. High-speed machining is always related to the achievable Metal-Removal Rate (MRR). However, in the normal machining, the MRR frequently is limited by the onset of chatter. The success of HSM, therefore, depends heavily on the ability to recognise and deal with the machining dynamics.

The term 'high-speed machining' is being questioned by some who prefer to use the term 'high-velocity machining.' However, the term 'high-speed machining' is the most widely used expression. High-velocity machining will imply a process that requires not only high spindle speeds but other parameters as well. For example, high feed rates, high rapid traverse rates, high acceleration and deceleration rates and fast tool change times can all become part of high-velocity machining. For example, conventional machining may use a feed rate of 500 mmpm while it is not uncommon to use feed rates as high as 2.5 mpm to 50.0 mpm or more. Note that the feed rates are now being talked of in metres per minute and not millimetres per minute any more. Typical parameters that will be considered as characteristic of the high-speed machining are shown in Table 12.3.

Table 12.3 *Comparison of conventional and HSM parameters*

Parameters	*Conventional*	*High-speed machining*
Spindle speeds, rpm	4,000	8,000 ~ 50,000
Axis feed rates, mm/min	10,000	2,500 ~ 60,000
Rapid feed rates, mm/min	20,000	20,000 ~ 60,000
Accelerations, g	—	0.5 ~ 2.0

In the mould and die industry, high-speed machining allows a trade-off between time on the milling machine and time spent on the polishing bench. This shift often yields a double bonus—high-speed machining reduces total machining time as well as reduces or almost eliminates manual die finishing processes. It is possible to come with similar examples for other classes of industries as well where high-speed machining is becoming increasingly popular.

High-speed machining, specifically vertical-axis milling, is not very different from conventional milling. In high-speed machining, the slower and heavy cuts are being replaced by faster and lighter cuts. As a result, the material-removal rates are higher using high-speed milling techniques. The high-speed milling cutter makes large volumes of small chips instead of the big chips as done in the conventional milling process.

In general, 8 000 rpm is considered the starting point for the rotational speed of the spindle in case of high-speed machining. However, it also depends upon the diameter of the cutter used, with smaller diameters going for much larger rpms approaching that of 30 000 to 50 000.

In HSM, the depth of cut is decreased while increasing the feed rates such that the material-removal rate is increased reducing the total machining time for the job. The main advantage of the small depth of cut is that the cutter takes less power to get through the material. Further, chip load is created by feeding a rotating milling cutter into the workpiece which is given by

$$\text{Chip load} = \frac{\text{feed rate}}{\text{number of teeth} \times \text{rpm}}$$

Using cutters with more teeth is another method of getting the benefits of high-speed machining with a limited rpm spindle. That means that a four-flute milling cutter permits twice the feed rate of two-flute cutter at the same rpm to maintain the same chip load. This increases the cutting forces, which in turn increases the horsepower required at the spindle. This can be offset by the reducing the depth of cut. Care has to be taken to see that a reduction in the depth of cut is more than offset by the increase in the feed rate such that the total material-removal rate is higher.

In order to optimise the HSM process, it is important to select the speeds and feeds, which allow for chatter-free machining. For this purpose, an automatic system called Chatter Recognition and Control (CRAC) system (Fig. 12.33) can be used. During machining, the sound generated by the cutting process is continuously sensed by the microphone. The frequencies of the sound are matched for any of the frequencies other than tooth-passing frequency or its harmonics (which are present even in stable machining). If other significant frequencies are detected then those are chatter frequencies. The CRAC System follows a simple rule of trying to make the new tooth-passing frequency match the old chatter frequency. The result is a move from an unstable spindle speed to a stable one. The feed rate is changed along with the spindle speed to keep the chip load a constant. Since the slot cutting is the worst case, no chatter will occur if cuts are chosen

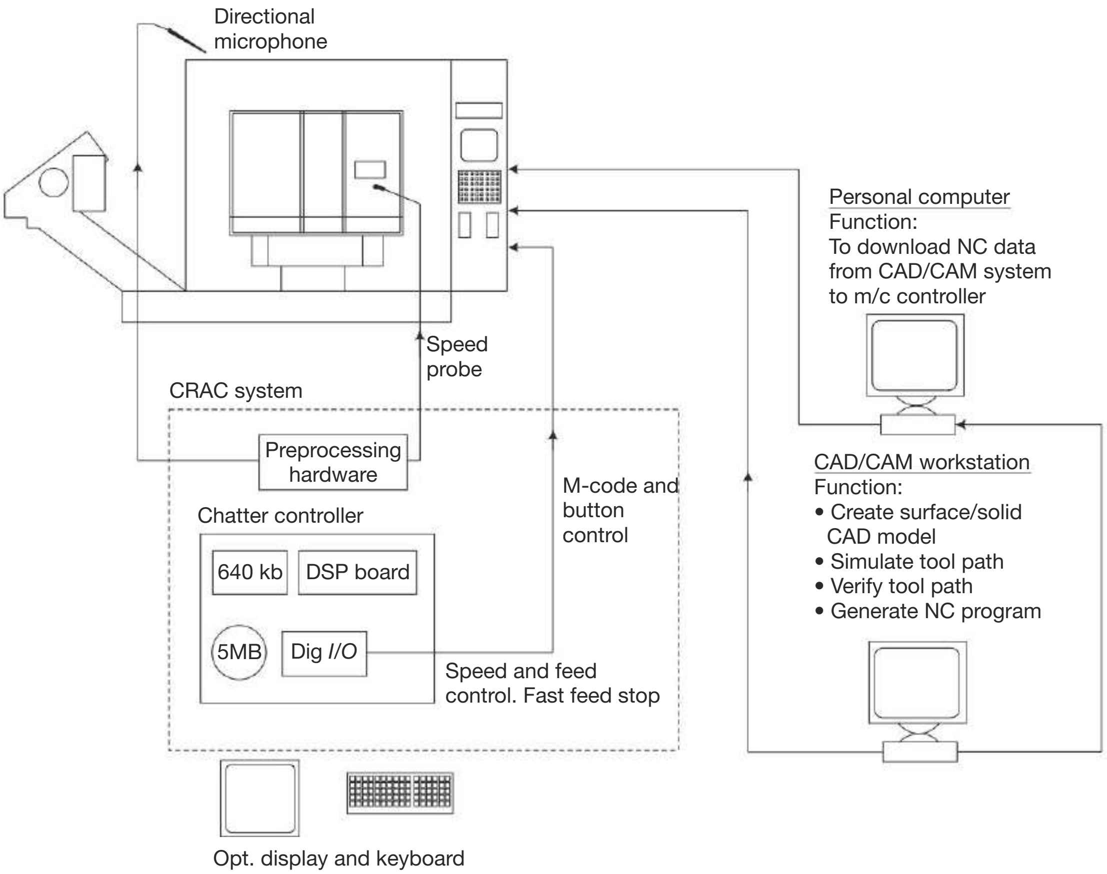


Fig. 12.33 *Chatter recognition and Control (CRAC) system used for optimising the metal cutting process parameters in HSM*

during NC programming which stays below the permissible axial depth of cut and the tool being used in a near-optimum condition (highest Metal-Removal Rate, MRR). Typical values thus obtained are shown in Table 12.4.

Table 12.4 *Tool database of stable cutting parameters for MAKINO A55*

Dia. mm	*Flute length mm*	*Overall length mm*	*Chip load mm/t*	*Corner radius mm*	*Opt. speed rpm*	*Opt. DOC mm*
8 b	23	40	0.08	4	19 447	1
10 b	40	40	0.08	5	20 000	2
12 b	40	40	0.1	6	20 000	5.5
16 f	35	57	0.1	0	17 232	5.5
8 b	40	60	0.1	4	18 500	0.2

Kennametal Short Series Holder with HSS end mill (b - Ball end, f - Flat end) with allowable speed = 20 000 rpm.

Material : Al 7075

In view of the considerable demand for high-speed machine tools, a large variety of machine tools are available from the major manufacturers. The comparative specifications of some of the machine tools that are currently available are shown in Tables 12.5 and 12.6. Table 12.5 refers to the vertical-axis machining centres while Table 12.6 refers to horizontal-axis machining centres. It can be noticed that there are a large variety of machines available in the vertical axis because of their application in the die and mould industry.

Table 12.5 *Specifications of high-speed machines vertical axis*

Specifications	*Mikron HSM 700*	*Mazak FJV-25N*	*Makino V55*	*Deckel DMC V65*
X-Axis, mm	700	1020	900	650
Y-Axis, mm	550	510	500	500
Z-Axis, mm	450	460	450	500
Spindle power, kW	10	30		15
Optional, kW				
Spindle range, rpm	42,000	25,000	14,000	18,000
Optional, rpm				30,000
Acceleration	10 m/s^2	2.8 s	0.6 g	1 g
Tool holder	ISO40	ISO40	ISO50	SK40
Optional	HSK-E50			HSK-E50
Feed range, mm/min	40,000	50,000	50,000	60,000
Rapid rate, mm/min	40,000	50,000	50,000	60,000
Tool magazine, Stations	12 40 (opt)	30		30 60 (opt)
Controller	ATEK HS-Plus	Mazak		SIN 840D TNC 430M

Table 12.6 *Specifications of high-speed machines horizontal axis*

Specifications	*Hüller Specht 500L*	*Makino A55*	*Hüller Nbh 110*
X-Axis, mm	630	560	630
Y-Axis, mm	630	560	630
Z-Axis, mm	500	560	600
Spindle power, kW	22	18.5	13
Optional, kW			32
Spindle range, rpm	16,000	12,000	12,000
Optional, rpm	24,000		16,000
Acceleration	14 ~ 20 m/s^2	1.7 s	10 m/s^2
Tool holder	HSK 63	ISO40	HSK 63
Optional			ISO40
Feed range, mm/min	100,000	30,000	75,000
Rapid rate, mm/min	120,000	30,000	75,000
Tool magazine, Stations	24 36/48/72	40 60/128/208	36 72/130
Controller		Makino	

A few of the actual case studies conducted by the machine tool manufacturers (Makino and Mikron) are given in summary form in Table 12.7. It can be noticed from Table 12.7 that the complete finishing of these dies and moulds require very little time, thus saving a large amount of money and effort in the process. This is able to justify the additional cost of the high speed machining of dies and moulds.

Table 12.7 *Some case studies in dies and moulds*

Description	*Hardness HRC*	*End mill dia used*	*Speeds used RPM*	*Machining time, min*	*Reduction in machining time*
Cross-valve mould	53	2, 3, 6, 10	8,000 ~ 12,000	54	—
Switch-cover mould base		2, 5, 6, 8	2390 ~ 8000	61.5	—
Gearbox cover	50	6, 10	10,000	134	90%
Cassette case	56 ~ 58	1.5, 3, 6	2600 ~ 12,500	98	85%
Watch die	60	1, 2, 6, 8	5000 ~ 34,000	185	—
Fitting mould cavity	52 ~ 54	2, 6	9000 ~ 30,000	92	—
Connecting rod die	48 ~ 50	3, 6, 10	5000 ~ 32,000	176	—

12.4 MACHINE CONTROL UNIT

The NC control systems in the beginning were hardwired with all the control logic to operate the machine tool embedded in the hardware circuitry. However, with the availability of the microprocessor towards early 1970's, rapid developments have taken place in the controller development. All the present-day controls are computer numerical control or CNC systems incorporating the latest microprocessors.

The availability of microprocessors has greatly enhanced the capability of the machine control unit in terms of the functions as well as the reliability by eliminating a large number of devices into a few. Thus, the present-day MCU is a very small unit compared to the earlier units. Along with the developments in the hardware, the control software also has gone through a large amount of changes and some of the following are the major benefits that were accrued because of the incorporation of the microprocessors.

- Larger part-program storage running to MB rather than kB or single blocks in the previous controllers
- Part-program graphical proving and editing
- Part-program generation using conversational part-programming methods such as FAPT TURN
- Tool-life management function, which includes larger number of tool offset registers as well as monitoring the life of the individual tools used
- Background part-programming methods
- Drip feeding of part programs when they are very large in cases such as finish machining of 3D contours of dies and moulds
- Enhanced part-programming facilities such as
    - Complex interpolations such as parabolic and helical
    - Additional canned cycles (other than the drilling series G 80 to 89)
    - Repetitive part programming using functions such as DO loops,
    - Use of subroutines and macros
    - Probes for inspection programs
    - Use of parameters in part programming
    - Help for operator instructions
    - Special geometric calculation facilities
- Better interfaces to the outside world (serial as well as parallel communications)
- Diagnostic facilities with the possibility of direct linking with the service centres using modems
- Enhanced DNC (direct numerical control) functions with links to factory networks
- Use of standard operating systems such as Windows 95/98 with the associated use of the controller for other functions
- Better shop-floor control by the use of two-way linking through the PLC (programmable logic control) with the outside world
- Enhanced machine control such as adaptive control, lead screw pitch error compensation, thermal compensation, etc.
- Enhanced machine control for high-speed machining by having a look-ahead facility
- Multiple-axis machining with more axes simultaneously

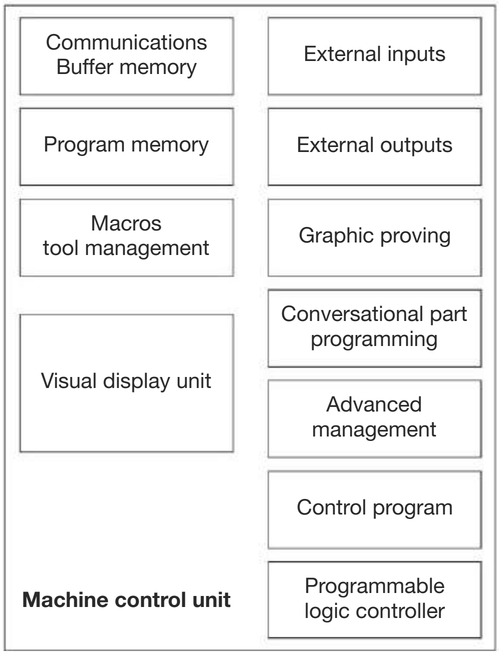


Fig. 12.34 *Organisation of the modern machine control unit functions*

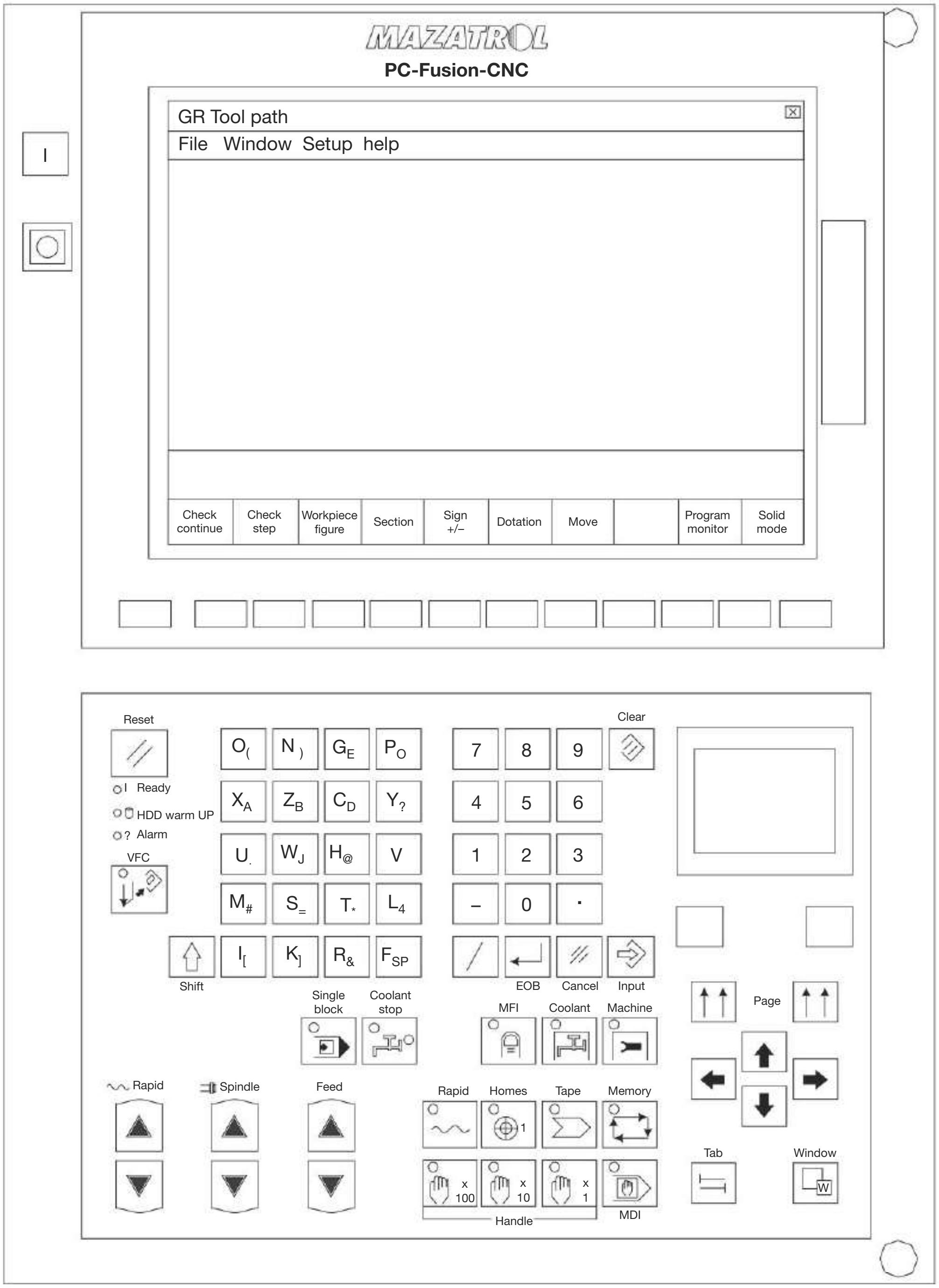


Fig. 12.35 *Present-generation machine control unit showing the various options present on the front panel Yamazaki Mazak Mazatrol PC Fusion CNC*

A typical CNC Machine Control Unit (MCU) has a number of units present inside to do all these functions. An example of subsystems present in a MCU are shown in Fig. 12.34. A typical MCU of the present generation is shown in Fig. 12.35. A brief description of what these subsystems mean for the performance of the MCU is given below:

Control Program The control program is that part which actually controls the various elements of the CNC machine tool. It has to take into account the interactions between the control-system components, the machine-tool mechanics, tooling, material changes during machining and other interactions that result in the final machine-tool performance. It should analyse the part program a few blocks ahead so that the speed of the slides can be adjusted in advance to cater for future difficulties in the path (sharp changes in direction, tight radii). The control program should fully integrate with the servo and feedback systems involved.

External Inputs This allows for interaction of the CNC with the outside world. For example, to link with a PC for DNC functions such as part-program downloading, remote datuming and operation so that it could be linked to integrated systems such as robots and Automated Guided Vehicles (AGV), etc.

External Outputs To send output from the CNC to outside world, for such tasks as uploading the generated or corrected programs, sending the controller information such as certain PLC register status, diagnostic information, etc.

Program Memory To store the part programs and macros in the MCU for use. Earlier controls had very little such memory, but the current-day controllers have much larger memory. With larger memory it is possible to store more programs as well as longer programs.

Additional Programming Facility Better programming facilities can be provided in addition to the basic G and M-code programming as standardised by ISO. One such major area is in the provision of newer canned cycles to facilitate major machining of the components using simple programming. Some additional canned cycles as available in Anilam Controls is given below:

- Ellipse
- Spiral
- Helical
- Irregular pocket / area clearance
- Rectangular profile (in/outside)
- Circular profile (in/outside)
- Mold rotation (any axis)
- Rectangular plunge pocket
- Circular plunge pocket
- Draft angle pocket
- Rough turning
- Rough facing
- Grooving (ID/OD)
- Face grooving

Communications (Buffer Memory) When the part programs become very large, it is not possible to store them completely in the MCU memory. In such cases, the part program resides in the DNC computer and it is fed into the computer in small drips. The MCU communications are, therefore, be in a position to support such an activity.

Advanced Management In addition to the basic functions as described earlier, it is more important for the newer controllers to provide additional facilities. Some typical possibilities are outlined below:

- Axis calibration
- Adaptive control
- Thermal compensation
- Pitch error compensation

Tool Management Tooling is a very important element in the running of a CNC machine tool not only for the productivity but also for the accuracy of the parts machined. Some form of tool-life management function needs to be present.

Diagnostics To keep the machine utilisation high, it is necessary to check the health of the machine tool continuously. For this purpose, a number of sensors are normally built into the machine tools. The control, therefore, should have the necessary ability to get the information from these sensors, analyse and then diagnose any faults already present or likely to come in future. Such diagnosis can be immediately communicated as alarms to the operator through the VDU. Alternatively, with a built-in modem and a connected telephone line, the user can send such information to the manufacturer or service facility for proper attention.

Graphic Proving The part program before actually doing the machining can be simulated on the screen of the MCU to verify the accuracy of the geometry generated. This verification normally has a number of options such as the actual tool path shown in various 2D-view planes or with 3D such as isometric or axonometric projections. However, this type of simulation though fast will not give the real feel of the actual material-removal process. Hence, most modern controls provide a solid simulation process in which the blank and the tool are actually shown in 3D shaded image. As the tool moves through the work material, the material is actually removed as a solid subtraction between the work volume and the swept volume of the tool in contact with the workpiece. The layer removed is shown with a different colour to give a realistic feel of the material-removal process. This process is actually able to simulate the uncut material at the corners because of the tool radius, burr formation, etc., more realistically. Some controls can also incorporate the surface-finish generation process in terms of the feeds and speeds used, so that the operator can get a feel of the final surface finish to be obtained on the component.

Conversational Part Programming Many of the controllers are available with some form of conversational part programming facility built in. This can be in many forms. FAPT-TURN was one of the first such systems which uses a vector drawing technique for making the turning part contour in 2D on screen. Based on that, the system automatically selects the tools, and generates completely the part program. Similar systems have become available in all the systems.

Keypad Traditionally, the keyboard is organised to suit the numerical control functions. That is how the keys follow the natural order of the part-program block that is most commonly used, as shown in Fig. 12.36. However, with the advent of the developments in the microprocessors, the current-day controls use either a 32 or 64-bit microprocessor, thus giving them a large amount of processing power. In spite of the more demands of the present-day machine tools, the processors still have sufficient processing capacity unutilised by the CNC machine tool in the conventional sense. Hence, these are now provided with a common operating system such as Windows 95 as used in most of the desktop personal computers. This allows the operator to use the controller for any other computational activities, while the machine is doing the metal-cutting operation in the background. This requires that the keypad should mimic the normal keyboard in terms of the key layout. Figure 12.37 shows such a keypad as used in Heidenhain control. Those that use the earlier type of key layout optionally allow for a normal keyboard to be attached to the MCU to carry out any of the Windows related tasks.

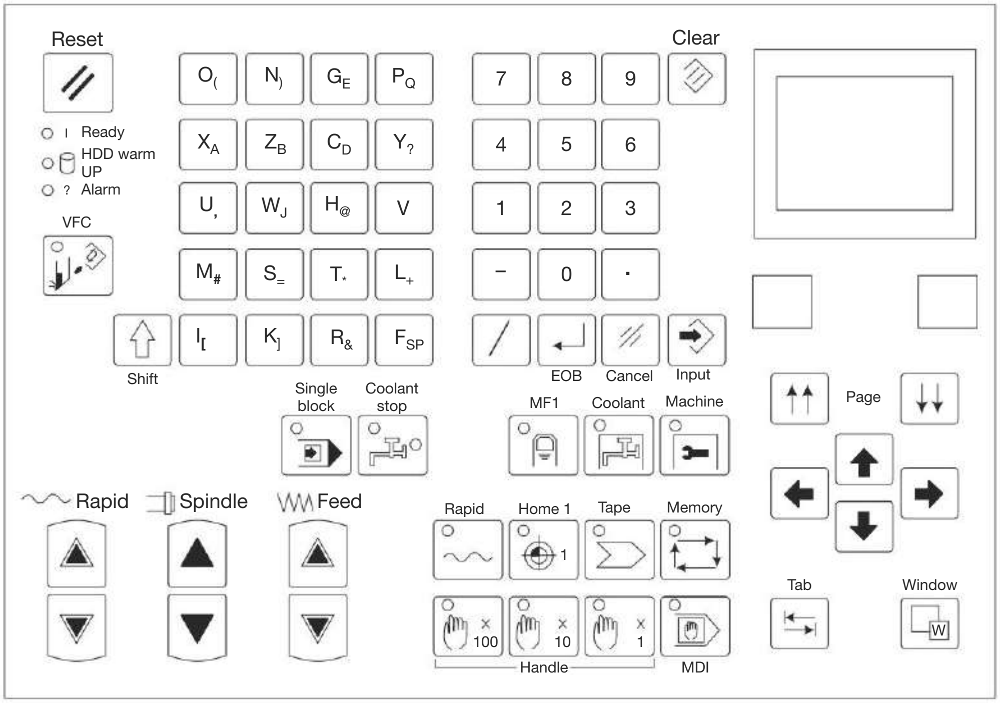


Fig. 12.36 *Organisation of the keypad of a machine control unit in traditional lines (Yamazaki Mazatrol)*

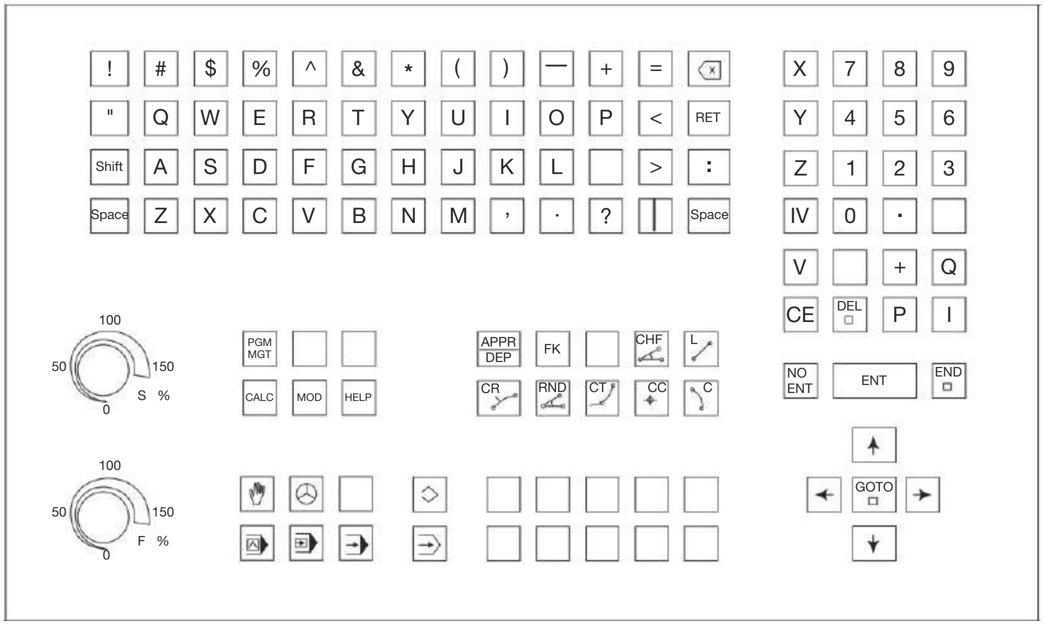


Fig. 12.37 *Organisation of the keypad of a machine control unit with QWERTY format as used in a personal computer (Heidenhain)*

12.4.1 Adaptive Control

According to the Webster's dictionary, to adapt means 'to change (oneself) so that one's behaviour conforms to new or changed circumstances.' The terms 'adaptive systems' and 'adaptive control' have been used as early as 1950. In the context of CNC Machine Tools, Adaptive control implies that the CNC system is responsive to adapt itself to operate at those machining parameters that result in higher productivity.

Starting in the early 1950s, the design of autopilots for high-performance aircraft motivated an intense research activity in adaptive control. High-performance aircraft undergo drastic changes in their dynamics when they fly from one operating point to another that cannot be handled by constant-gain feedback control. A sophisticated controller, such as an adaptive controller, that could learn and accommodate changes in the aircraft dynamics was needed.

An adaptive controller is formed by combining an online parameter estimator, which provides estimates of unknown parameters at each instant, with a control law that is motivated from the known parameter case. The way the parameter estimator, also referred to as adaptive law in the book, is combined with the control law gives rise to two different approaches:

- In the first approach, referred to as *indirect adaptive control,* the system performance parameters are estimated online and used to calculate the controller parameters. This approach has also been referred to as *explicit adaptive control*, because the design is based on an explicit system performance model. It is also called adaptive control optimisation or ACO.
- In the second approach, referred to as *direct adaptive control*, the system performance model is parameterised in terms of the controller parameters that are estimated directly without intermediate calculations involving system-performance parameter estimates. This approach has also been referred to as *implicit adaptive control* because the design is based on the estimation of an implicit system-performance model. It is also called Adaptive Control with Constraints or ACC.

When the part programs are written, the programmer checks the machinability handbooks to get the appropriate cutting speed, feed and depth of cut for the particular machining operation taking the cutting tool and the workpiece material combination and the required surface finish. These values from the handbooks are generally conservative values and are useful as starting values. If the CNC program is run with those values, the actual machining time for the part may not be optimal. An MCU having adaptive control will actually modify these values continuously based on the actual system performance in real time by measuring certain parameters.

Adaptive Control Optimisation (ACO) In adaptive control optimisation, the system-performance model P(PI) is parameterised with respect to some known or unknown parameter vector PI as shown in Fig. 12.38. For example, in the case of CNC machine tools, the system-performance model, PI may represent some parameter that identifies a performance index (PI) as follows:

$$\mathrm{PI} = \frac{\mathrm{MRR}}{\mathrm{TWR}}$$

where MRR represents material removal rate, and TWR represents the tool wear rate. In metal cutting, MRR is dependent upon the cutting speed, feed rate and depth of cut. Tool-wear rate depends

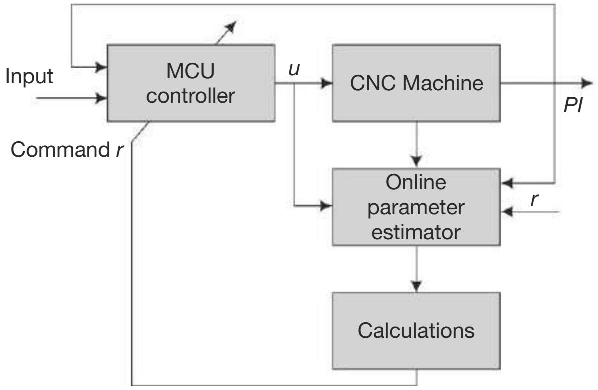


Fig. 12.38 *Adaptive control optimisation*

upon the cutting-process parameters as well as the other dynamic characteristics of the machine-tool–cutting tool-workpiece system. The tool wear cannot be measured directly from the process, and hence some parameters are measured such as cutting power, cutting forces, vibration signature of the cutting tool, acoustic emission from the worn cutting tool and cutting-tool temperatures and are fed into the parameter estimator program. The online parameter estimator generates an estimate PI (*t*) of PI at each time *t* by processing the system performance inputs *u* and outputs *y*. This index is compared with the desired value by another program in the system software which recalculates the cutting process variables like feed rate and speed rate so as to keep performance index at highest level without violating any of the constraints. The process variables are thus continuously changed so as to obtain optimum value of the performance index.

From the above-mentioned discussion on AC systems, it can be observed that these systems may be advantageously used [Rao]:

- to increase productivity
- to maintain high levels of workpiece precision
- to raise the level of automation by reducing operator intervention
- to increase cutter life by reducing tool loads
- to increase workpiece accuracy and CNC machine life by maintaining low levels of vibration

Adaptive Control with Constraints (ACC) In this system, the control is simplified by incorporating certain safe limits for any of the relevant process parameters in the form of constraints. A few examples are

- use the available spindle power to maximum
- limit the deflection of the cutter
- limit the cutting-tool temperature
- limit the cutting force
- limit the vibration amplitude of the cutter

Typical process variables measured are cutting forces, torques, spindle motor power/current, tool wear, tool deflection, cutting temperature, vibrations, etc. The type of variable measured depends upon the convenience of measurement and how closely it affects the system performance. A number of sensors or transducers are integrated into the CNC control systems. The choice of either contact type (cutting forces, vibration signature, etc.) or non-contact type (temperature, acoustic emission, etc.) instruments could be used. Generally, a single parameter is not able to provide a very clear performance measure. Hence, a number of parameters are measured and then they are fused to come up with a more reasonable performance indicator.

The measured value is compared with the permissible value which has been incorporated in the adaptive control software. It is ensured that at no time the measured value exceeds the permitted one. The program calculates those values of machining parameters which give the desired extreme objective, say maximum metal-removal rate. The programmed values of machining parameters like that of the feed rate are continuously replaced by the new calculated values. The real-time effect of operating these values is checked by measuring the constraints' values as explained earlier. The value of operating parameters is accordingly adjusted so as to remain within the imposed constraints (Fig. 12.39).

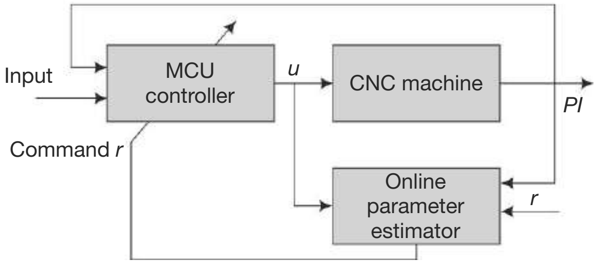


Fig. 12.39 *Adaptive control with constraint*

Most of the adaptive controls which are commercially available are ACC type, though a lot of research continues to be done in the ACO systems.

12.5 SUPPORT SYSTEMS

Though the main functional units of CNC machine tools as described earlier are important, they also need a number of support units in order to perform their intended functions satisfactorily.

Chip Removal One of the main problems of CNC machine tools is the amount of chips generated. Since these machines are designed to be operated 24 hours a day and also have considerable amount of metal-removal capability, the working area of the machine tool gets clogged by the amount of chips if they are not removed quickly. Most of the machines, therefore, are provided with some type of chip-handling capability, either standard or optional. Figure 12.40 shows typical chip-removal facilities in turning centres and machining centres. In the case of turning centres, the slant bed allows for the chips to be accumulated at the bottom away from the machining zone. From there, the chips need to be transported by a conveyor into a collection bin, which can be removed periodically. A similar arrangement is made for machining centres as well. In the case of flexible manufacturing centres where a number of CNC machine tools are linked, some kind of integrated chip-handling facility needs to be provided which can collect the chips generated from all the machine tools in the system.

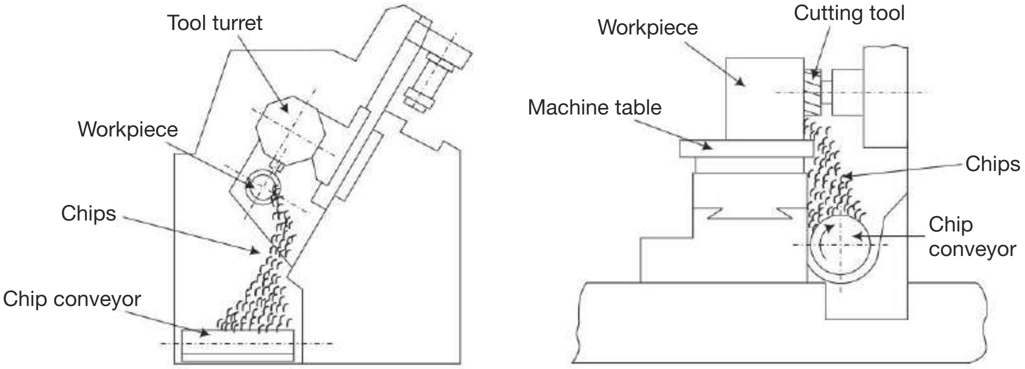


Fig. 12.40 *Chip conveyors used in CNC machine tools*

Work Support in Turning Centres In the case CNC turning centres work needs to be supported when it is relatively long. Many turning centres have a tailstock which can be manual or can be fully programmable under an M-code. The quill of the tailstock is generally a programmable hydraulic unit. For fully programmable tailstocks, it becomes necessary that the workpiece be in position with the centre to receive the tailstock quill. To ensure this, it may be necessary to have a steady rest as shown in Fig. 12.41.

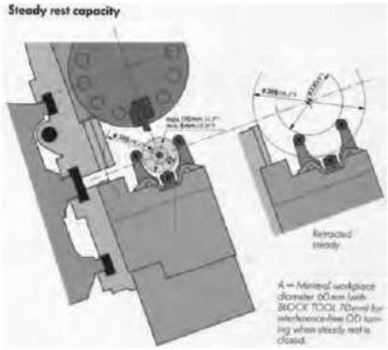


Fig. 12.41 *Programmable steady rest used in CNC turning centres (Courtesy, George Fischer, Switzerland)*

Chuck-Jaw Changer Another important support system required for unattended operation of a CNC turning centre is the chuck-jaw changer as shown in Fig. 12.42. This consists of a jaw magazine required for a variety of jobs arranged in a cylindrical drum above the chuck. The jaws needs to be pushed into the slots in the drum for replacement purpose.

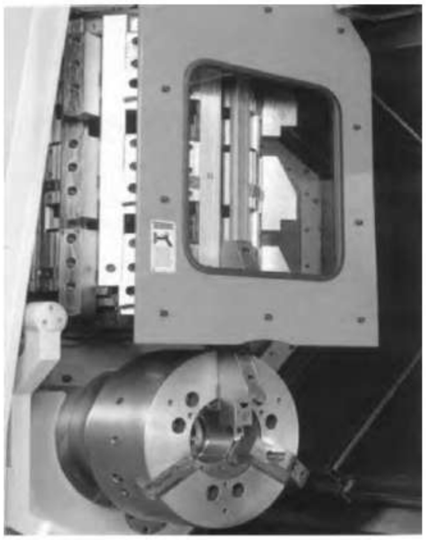


Fig. 12.42 *Automatic chuck-jaw changer for CNC turning centres (Courtesy, Yamazaki Mazak Corp., Japan)*

12.6 TOUCH-TRIGGER PROBES

The touch-trigger probe is basically a tool used for measurement. It consists of a spherical sapphire unit connected elastically to a rigid measuring unit as shown in Fig. 12.43. When the probe moves along a direction and touches a surface, the elastic connection deflects and triggers the measuring system. The trigger is basically an omnidirectional switch capable of detecting deflections in any direction. The moment the probe touches a workpiece surface, it is sensed and a signal is sent to the control to stop the movement of the axes. From that the physical position of the surface where the probe contacts gets recorded. Since the probe tip is spherical, the contact between the probe and the measured surface is a point contact. The controller automatically compensates for the radius of the probe. Though a signal is sent to the controller to stop the axes, the stoppage is not instantaneous in view of the inertia of the system. Thus, the probe shaft actually deflects since the axes overtravels. However, the overtravel is small depending upon the axes velocity. It is a good practice while using these probes to reach the surface slowly so that the error is small. The machine-tool manufacturers actually calibrate the machines using some kind of a ring to estimate this overtravel so that it can be automatically adjusted in the controller. These are the basic measuring units used in Coordinate Measuring Machines (CMM).

However, it is possible to use these probes in CNC machine tools (provided certain facilities exist in the CNC controller) for the purpose of measurement as well as tool setting. The CNC controller has a tool-probing instruction or cycle through which the measured position can be stored into parametric variables. These parameters may be computed to evaluate the necessary inspection function. This system not only enhances the utility of the CNC machine tool, but also improves the productivity by reducing the setting time for complex jobs, by measuring the tool offsets on the machine, etc. It improves the quality of the jobs produced by measuring the exact tool offsets just before the machining to start for each of the parts in a batch. Probing could also be used for the purpose of tool breakage monitoring as explained later.

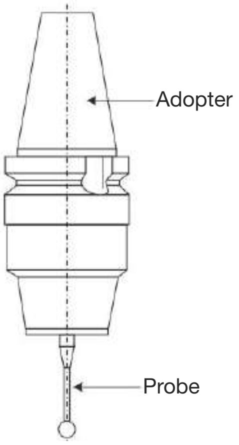


Fig. 12.43 *Touch-trigger probe used for inspection on a CNC machine tool*

The various types of transmission methods available for the measured value from the probe to the CNC controller are the following:

Direct Hard-wired Connection This may be used only for tool-setting probes since all the elements concerned with the measurement are stationary and fixed in space.

Inductive Transmission This system is used for measuring probes. A probe is treated like any other tool and is placed in the spindle. During the gauging cycle, the probe signals are transmitted across inductive modules as shown in Fig. 12.44.

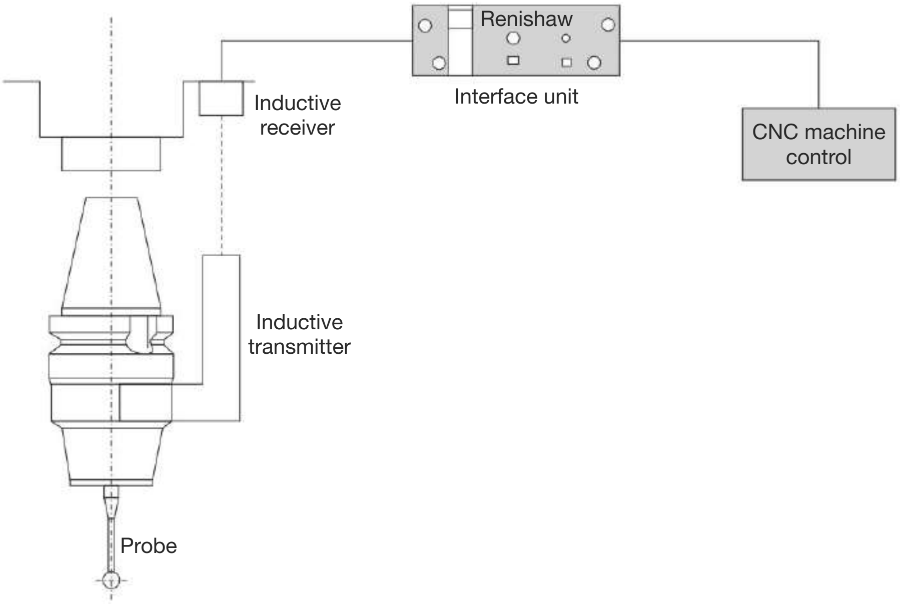


Fig. 12.44 *Inductive transmission systems used in machining centres*

Optical Transmission This system is also used for measuring probes. The optical system communicates by invisible infrared coded messages, which convey probe and control signals between the optical transmitter and receiver as shown in Fig. 12.45. These are the most convenient form since the optical receiver can be located outside the machining area and no specific location is required and are, therefore, widely used. They also use batteries, and hence require minimum wiring.

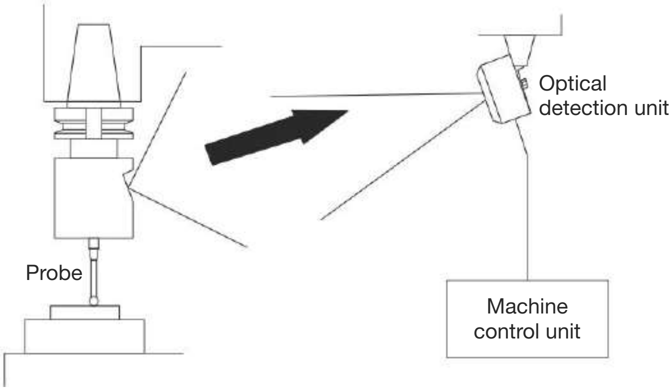
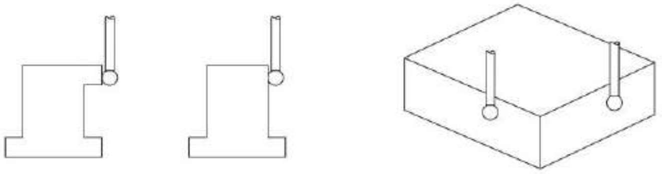


Fig. 12.45 *Optical transmission systems used in machining centres*

These probes can be used for a number of applications

- Datuming of the workpiece
- Workpiece dimension measurement
- Tool offset measurement
- Tool breakage monitoring
- Digitising

Datuming of the Workpiece When the component is to be clamped to the machine-tool table, it is necessary to establish the component datum before the part program can be activated. Traditionally, the dial gauges are used for the purpose. Inspection probes can be used for manual datuming of the components in case of machining centres. Before the component is clamped, the probe can be brought near the surface to be used for datum (set point/ surface) and accordingly, the set-point values entered in the controller registers. The probe can be made to touch the respective surfaces in order to obtain the datum as shown in Fig. 12.46. It is possible to touch at more than one point on the surface in order to compute a more accurate datum surface than otherwise possible.



Component identification

Workpiece datuming

Fig. 12.46 *Probing examples for workpiece set-up on the machine-tool table*

Workpiece Dimension Measurement Once a component is fully machined or after a critical feature is machined, probing may be carried out to check that the generated features are accurate. By making use of the machine-tool axis transducers, it is possible to measure any feature on the component. An example is shown in Fig. 12.47 for the measurement of the slot. First, the probe is made to touch one surface and the slide position is noted. Then the probe is made to move in the opposite direction and touch the opposite surface. The difference between the two readings gives the thickness of the slot.

Two examples of the kind of inspection that may be carried out are shown in Fig. 12.48. Similar arrangements may be made for connecting the probe systems in turning centres also.

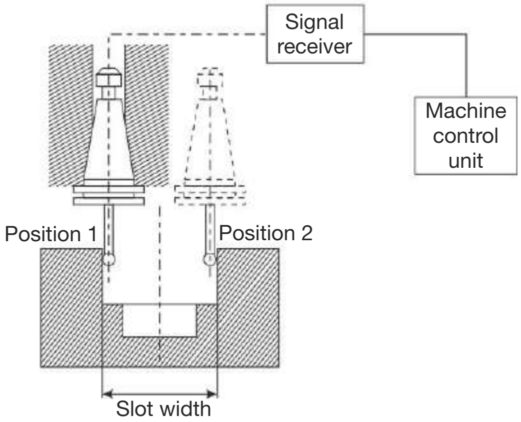


Fig. 12.47 *Use of probe for measuring the width of a slot*

The inspection can be *in-cycle* or *post-process*. In-cycle gauging refers to the technique of measuring the component using probes in between cuts. If the probing is done between the cuts, the last cut may or may not be carried out, depending upon the required dimension measured. The CNC part program is suitably written with probing cycles, and skipping the blocks based on the results of measurement. Also, if necessary an additional cut can be programmed to bring the part into the

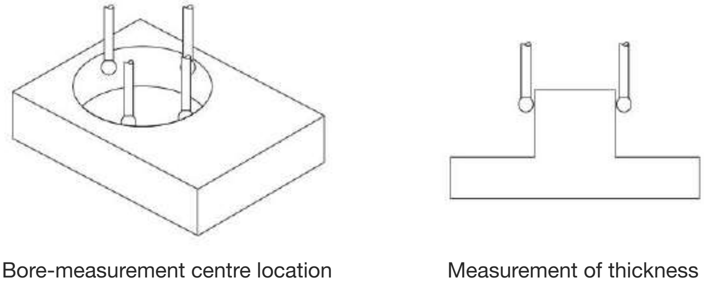


Fig. 12.48 *Probing examples: (a) Inspection of a bore for diameter and centre position, (b) Inspection of a web thickness*

tolerance level by compensating any of the wear present on the tool. However, the main problem in such cases is the quality of the surface in terms of the burrs present, and the cutting fluid traces on the surface makes the measurement a little difficult. Care, therefore, has to be exercised in programming to eliminate such surfaces. Thus, in-cycle probing allows for all the jobs to be controlled to closer tolerances required in the modern industry. However, the time taken is relatively large in the use of probes, since the probe movement when approaching the measured surface has to be relatively slow to reduce the overshoot error.

In post-process gauging, the inspection is carried out after completing all the machining, without removing the job from the work-holding fixture. In this case, it is established whether or not the part is within the tolerance limits. This allows for the remedial cuts to be taken to bring the job within the tolerance limits required for the critical dimensions. In this case, automatic correction of size is not possible, since the system can only give the results of inspection, and then the remedial action is to be manual.

Tool Probing The types of probes required for tool probing are different from the ones used for inspection. The probe head is flat faced with a round cross-section as shown in Fig. 12.49 for machining centres. The probe is clamped on the machine-tool table at one end not used for workpiece clamping. The tool can be brought in contact with the probe tip in both the directions to obtain the tool offsets (length and diameter).

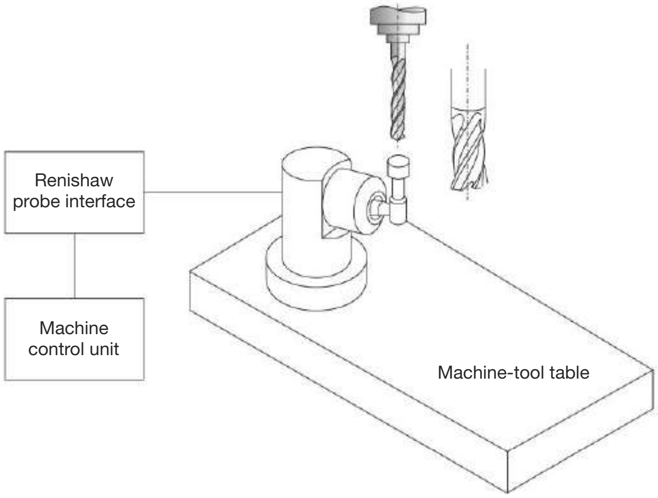


Fig. 12.49 *Probe type used for tool offset determination in machining centres*

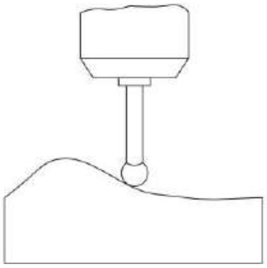


Fig. 12.50 *Probe used for digitising a profile*

The typical tool-setting times are of the order of 20 seconds including the automatic entry of offset values into the controller memory. As explained later, the same probe is also used for monitoring the tool breakages in between the cuts.

Digitising Digitising refers to the conversion of profiles into coordinate points at close intervals. This can be applied to curves as well as surfaces. Since the touch-trigger probe generates the point of contact with the help of the CNC control unit, it is also possible to use it for digitising complex profiles and surfaces. A typical example is shown in Fig. 12.50 for digitising a curve. The probe is programmed to move along the surface at small intervals and the obtained point is noted in the controller. For digitising a surface, it is necessary to move the probe tip through the two sets of curves in perpendicular directions as shown in Fig. 12.51. The points can be uploaded into a PC from the controller, where it can be used for converting into a CNC program by using appropriate interpolation techniques.

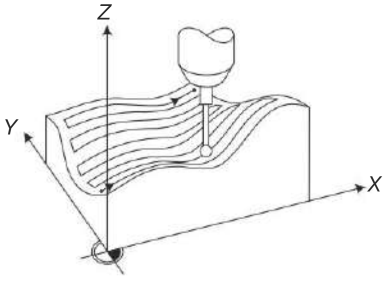


Fig. 12.51 *Probe used for digitising a surface*

Summary

- CNC machine tools have evolved from the basic machines in the beginning to the very sophisticated systems that provide lot of flexibility as well as productivity. The discussion can be done specifically in two broad categories as machining centres and turning centres, though there are some machines, which combine both these functions.
- Vertical-axis machining centres are used primarily because of their versatility in adopting for a large range of part geometries. These have a number of accessories that enhance their functionality. These machines also come with a large number of axes to cater to complex 3D profiles which otherwise cannot be machined with the 3-axes machines.
- Horizontal-axis machining centres are sturdy and are used for heavy machining applications. These, with a programmable rotary table, are able to completely machine four faces of boxlike components in a single setting.
- In view of the large majority of industrial components being axisymmetrical, there is a large variety of CNC turning centres in the market. They include turning centres with multiple spindles, opposing spindles, turrets with a large number of tools, tool turrets with driven tooling, etc. Many turning centres have additional axes such as *C* and *Y* to increase the range of machining surfaces that can be completed in a single setting.
- Vertical turning centres offer higher productivity by increasing the material-removal rate. Modification of the same machine by adding the multiple spindles allows for large volume production of industrial components.
- High-speed machining has been found to improve the machining accuracy and finish, thereby reducing the final finishing operations in many cases. This has now become the mainstream operation with spindle speeds up to 20,000 rpm and cutting feed rates of up to 100 m/min.
- Machine Control Unit (MCU) is the main brain behind the operation of a CNC machine tool. There are a number of functions that are done by an MCU. Most of the modern MCU are organised with modular components so that the enhancement of the functions can be done as required. From the programming point of view, modern control units offer a large number of enhancements.

- Adaptive control optimises the performance of the control, thereby improving the productivity of a CNC machine.
- There are a number of support systems that operate in conjunction with the machine tool. Some examples are the chip removal, work support, etc.
- A touch-trigger probe operates in the place of a tool in the spindle to provide a measurement of the coordinates of its tip. These are used to measure the dimensions of the finished part, so that the part program itself can complete inspection of the part. These are also used to determine the tool offsets and some digitising work as well.

Questions

1. Give a brief description of the CNC machining centres.
2. How do you select a CNC machining centre for machining the following components?
   (a) Aircraft fuselage section
   (b) Compressor rotor
   (c) Gearbox housing of an automobile
3. What are the ways in which 5-axes can be realised in CNC machining centres (vertical axis as well as horizontal axis)?
4. Briefly describe the automatic pallet changer as used in CNC machining centres.
5. Give a brief description of CNC turning centres.
6. Explain the reasons why turnmill centre is preferable to a 2-axis CNC turning centre?
7. Explain some of the multiple-axis (more than 2-axes) turning centres with applications.
8. What are the differences between horizontal- and vertical-axis CNC turning centres? Give their applications.
9. What are the variations available in vertical CNC turning centres? Give their applications.
10. How do you distinguish between high-speed machining and conventional machining?
11. What are the applications of high-speed machining?
12. What are the advantages to be gained by the use of high-speed machining?
13. What are some of the benefits achieved by the modern CNC controllers compared to the pre-microprocessor era controllers?
14. What are the various sub-systems present in a modern CNC controller unit?
15. Explain adaptive control as it pertains to numerical control.
16. What is adaptive control optimisation? How is it used in CNC machines?
17. What is adaptive control with constraints? How is it used in CNC machines?
18. Give a brief description of some of the support systems found in CNC machine tools.
19. Give a brief note on the touch-trigger probes.
20. Give a brief description of the data transmission method between the touch-trigger probe and the CNC controller. Give their relative advantages.
21. What are all the applications where the touch-trigger probes can be used in the shop floor? Give a brief description.
22. Write a short note on digitising using touch-trigger probes.

13

CNC PROGRAMMING

Objectives

CNC part program is a detailed list of instructions that need to be executed by the Machine Control Unit (MCU) to achieve the final component shape. The machining sequence needed to manufacture a given part is broken down into small elements and written in a specific format understood by the MCU. The programming language needs to be studied to develop meaningful part programs. After completing the study of this chapter, the reader should be able to

- Understand the fundamentals of part programming in terms of the various steps needed to be taken for completing a successful CNC part program
- Comprehend the elements of manual part-programming methods using word address format and the ISO G-coding systems
- Learn the part-programming fundamentals related to the use of various word addresses
- Appreciate different preparatory (G codes) and miscellaneous functions (M codes) as used in CNC part programs
- Write and prove sample part programs for CNC machining centres in planar milling operations using the word-address format
- Understand the concept of canned or fixed cycles for the hole-making operations
- Learn the use of cutter diameter and length compensation while using multiple cutting tools

13.1 PART-PROGRAMMING FUNDAMENTALS

To be a good CNC programmer, one should have a fair knowledge about machine tools, cutting tools, fixtures to be used and the manufacturing process. He should also have a good understanding of geometry, algebra and trigonometry. In fact, machine shop experience is the prerequisite for a good programmer as only careful process planning can lead to efficient and practical programs.

The total steps involved in the development of a part program and its proving is shown in Fig. 13.1. The following are some of the steps that are detailed below:

- Process planning
- Axes selection
- Tool selection
- Cutting-process parameters planning
- Job and tool set-up planning
- Machining-path planning
- Part-program writing
- Part-program proving

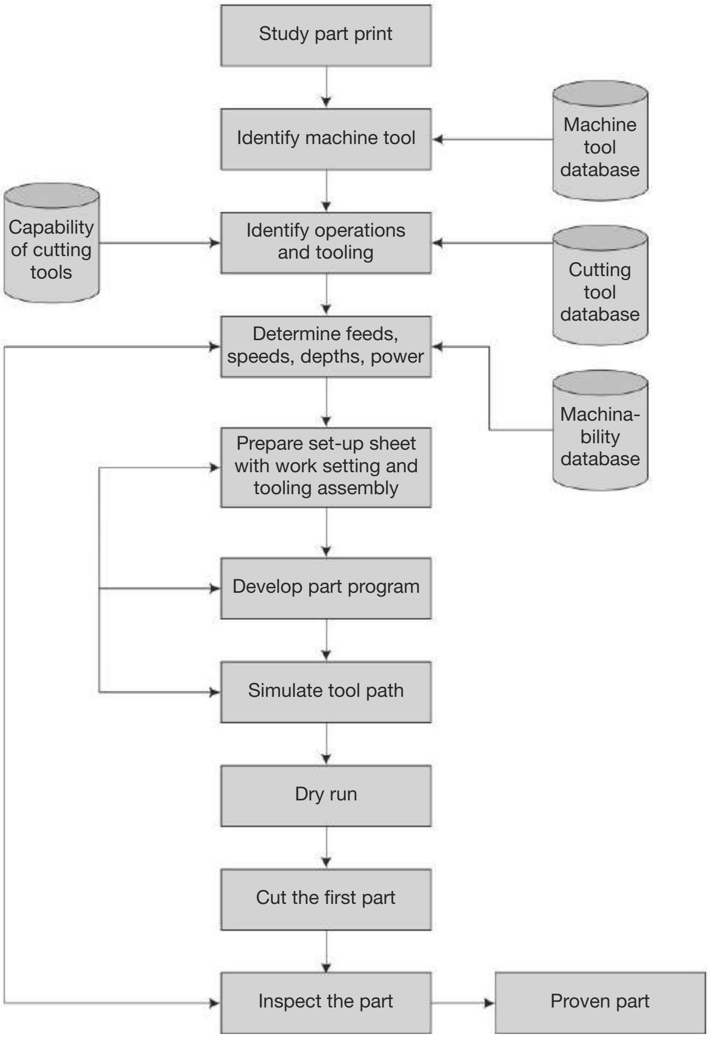


Fig. 13.1 *The steps involved in the development of a proven part program in NC machining*

13.1.1 Process Planning

Process plan is a detailed plan of the steps involved in manufacturing (machining) a given part. The following are the contents of a process plan:

- Machine tool used
- Fixture(s) required
- Sequence of operations
- For each operation
  - Cutting tools required
  - Process parameters

A programmer is supposed to carry out a careful study of the part drawing to prepare the process plan. The choice of the machine tool used depends upon the operations required, accuracy requirements, machine-tool capability and availability, cutting-tool availability and the shop practices. A careful choice of various options at this stage helps in deciding the final cost of manufacture of the part. A typical process plan is shown in Table 13.1 for the part to be machined as shown in Fig. 13.2.

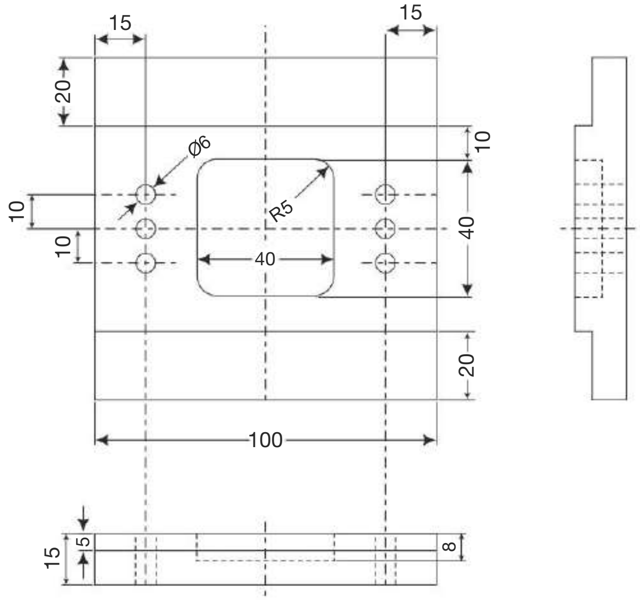


Fig. 13.2 *A typical component for NC machining*

Table 13.1 *Process plan for the component shown in Fig. 13.2*

Op. no.	*Description*	*Tools*
10	End mill the top face, 100 × 100 mm	Shell end mill, ϕ60 mm
20	End mill the steps, 20 × 100 × 5 mm	Shell end mill, ϕ60 mm
30	Mill pocket, 40 × 40 × 8 mm	HSS end mill, ϕ10 mm
40	Drill the six holes, ϕ6 × 15 mm	HSS twist drill, ϕ6 mm

13.1.2 Axes Selection

All the CNC machine tools rely on the axes system for describing the axes motion. To correctly describe the motion, it is therefore necessary to establish the axes system to be followed with the particular part. The ISO designation of axes was discussed in Chapter 10. In tune with that axes system, one has to choose the axes. However, it is also necessary for one to choose the axes system as appropriate to the machine tool coordinate system in question.

The axes system of all the CNC machine tools generally have a fixed datum position as designated by the machine-tool manufacturer. It may be called *reference position*, *fixed datum* or *home position*. This absolute datum position of the CNC machine tool may not be very convenient for setting the job. Hence most of the CNC machine tools come with 'floating datum'. In this case, the programmer can select the part datum anywhere in the machining limits of the machine tool based on the geometry of the part being machined.

The reference axes should be chosen so that coordinates for various features can be determined (Fig. 13.3). Here, X and Y are the reference axes. For the sake of convenience, the orthographic views of the component are shown.

The basis for choosing the axes system is more to do with the part geometry and the type of machine tool being used. When the operator is developing the program, it becomes extremely important to choose the right type of datum, since a careful selection eliminates a large amount of calculation process. Also, the part program becomes simple, being able to make use of the advanced software facilities such as mirror imaging, etc.

The first principle to be used while arriving at the datum is that if possible, keep all the parts in the first quadrant of the coordinate system. This helps in having all the coordinate values as positive. It helps the first-time programmer in eliminating as many errors as possible. Once enough experience is gained, it is possible for the programmer to carefully adjust the values. Similarly, touching the two sides of the pre-machined workpiece can easily do the setting of the tool.

The Z-axis datum is kept generally to match with the top surface of the workpiece. This helps in two ways. First, all positive values of the Z coordinate keeps the tool away from the workpiece so that the collision of tool with the work is avoided. Secondly, when the tool is to be set, the tool tip can be easily matched with the workpiece top surface.

Sometimes, the datum could be chosen as the geometric centre of the workpiece if all the geometry is symmetrical as shown in Fig. 13.3. In such a choice, the geometry calculation effort reduces to a minimum. Also, the mirror image facility available in the controller can be effectively exploited. The reader can notice the small number of dimensions required to describe the component. All the other dimensions are symmetrical about the axes system chosen. For the same component, choosing the left-side bottom corner as the datum causes a large number of dimension calculations needed as shown in Fig. 13.4.

13.1.3 Tool Selection

The choice of cutting tools is a very important function, since for a given operation many tools are feasible, but some of them are more economical than others. Therefore, in the economy of manufacture, it is essential to choose the right tool for the job. As a rule, we will only select the regular cutting tools for using in CNC machine tools. No special tooling is generally suggested, since the geometry can very well be generated by the CNC control.

As an example, when a contour is being milled, the choice can be an end mill or a slot drill. An end mill is stronger and can take deeper cuts than a corresponding slot drill. However, a slot drill can enter into a solid material, but an end mill cannot, in view of the fact that the cutting edge in the bottom does not extend to the

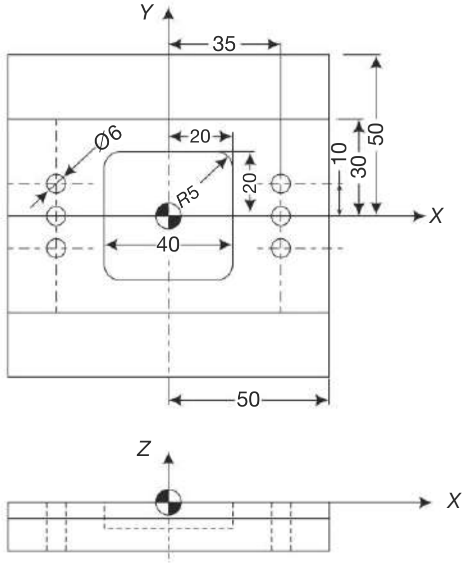


Fig. 13.3 *Part for NC machining shown with axes system at the centre*

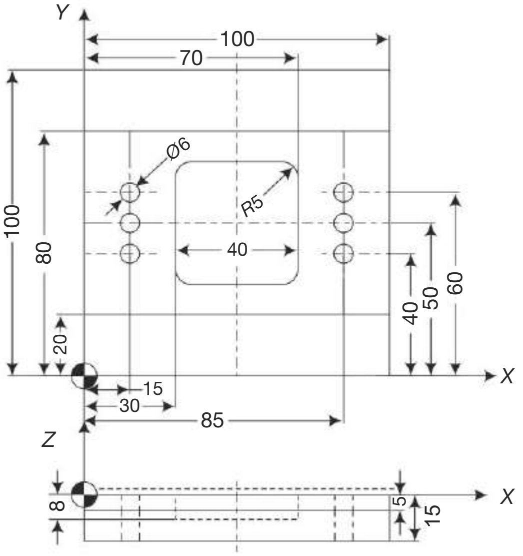


Fig. 13.4 *Same part as in Fig. 13.3 but with axes system at the bottom-left corner*

centre of the tool. As a result, an end mill should always approach the workpiece from the side while the slot drill can approach from the side or from the top.

Also, the size of an end mill or slot drill depends on the contour to be machined. You should choose the largest size of end mill available for better surface finish and higher material-removal rate. However, the tool radius may be limited often by the radius of curvature being generated.

For example, sometimes a tapered end mill or slot drill makes the machining very simple in generating the draft surfaces of dies and moulds. Otherwise, generating a draft angle may take a large amount of programming. Details of tooling are covered in Chapter 11.

13.1.4 Cutting-Process Parameters Planning

For a given tool and the operation selected, the appropriate process parameters are to be selected. These are to be generally taken from the handbooks supplied by the cutting-tool manufacturers or based on the shop experience. It is important in the context of CNC manufacture that the feeds and speeds selected should be as high as possible to reduce the machining time consistent with the product quality achieved. Details of process parameter selection are covered in Chapter 11.

13.1.5 Job and Tool Set-up Planning

This aspect is covered in a greater detail in the next section. This basically is aimed at setting the job on the machine tool and adjusting the cutting tool to the correct position. This is important since the accuracy of the geometry generated by the CNC machine tool is dependent on the initial position carefully defined.

13.1.6 Machining-Path Planning

This is a very important aspect of programming wherein the knowledge of machining operations plays a vital role. A careful planning of the tool path ensures that the requisite manufacturing specifications are achieved at the lowest cost. With the availability of complex canned cycles with many of the present-day controllers, this aspect has been simplified for the programmer by the careful choice of the cycles. The details of stock-removal cycles are covered in Chapters 14 and 15.

13.1.7 Part-Program Writing

This aspect deals with the actual writing of the part programs undertaking the format and syntax restrictions into account.

13.1.8 Part-Program Proving

This is another aspect, which the programmer should very carefully do before the part program is released to the shop. Once the program is made, it should be verified before it can be loaded on the machine-tool controller for the manufacture of the component. It is obvious that a faulty program can cause damage to the tool, workpiece and the machine tool itself. Sometimes, these accidents can prove grave for the operator and others around. One of the preliminary ways of avoiding such possibilities is to carry out a visual check of the program manuscript and of the displayed program on the VDU of the controller.

But this is understandably not enough in itself. A trial run can be carried out with or without the tool or workpiece to enable visualisation of movements taking place and of any collisions possible between the tool, the workpiece and the clamping device. At this stage, it is worthwhile stressing the point that while the program is being prepared, the positions of the clamps should have been taken into account and that they be clearly indicated in instructions to the operator. This is vital for eliminating the possibility of collisions occurring during machining.

During trial runs, feed and speed override control should be used so that the operator works at such values as enables him to exercise manual control comfortably and operate the emergency switch well in time. The program is run block by block, i.e., and after execution of each block, the machine waits till the operator manually presses the switch on the machine console for execution of the next block.

With the job and tool in position, dry runs are made, i.e., keeping a safe distance in between the tool and the job; the motions can be visualised for correctness. If during these trials, any mistakes are noticed, the program is examined and the necessary corrections made. After this, one component is made and checked. Based on this, speeds and feeds are modified and further corrections carried out so that correct profiles are obtained. Sometimes only one job, which may be quite complex and precise, needs to be made. This could even be of an expensive material. In such cases, the program is tried first on a cheap material, say wood, Perspex, etc. Only when the first trials are approved, the updated program is permitted to be used for further production.

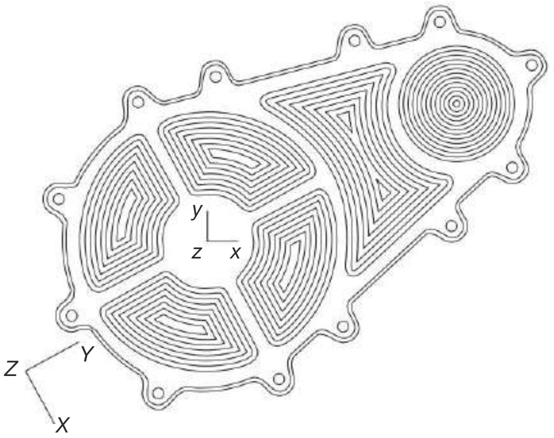


Fig. 13.5 *Tool path of the part for proving the NC part program*

Nowadays, graphical simulation packages are available on CNC systems, which make possible a graphical output on the VDU screen. This output shows the workpiece and the tool, the motion of the tool and the progressive removal of material as the program proceeds. Visualisation of this animation of the machining process helps to prove the part program before any actual machining is carried out. These verifications are carried out at a fast speed and thus the proving of the part program is done without much loss of time. A typical example is shown in Fig. 13.5, which shows in two dimensions the path taken by the cutting-tool centre.

It is also possible that the verification can be carried out on a microcomputer screen. Through these, it is possible to see how the tool path is programmed. A more advanced version is the programs, which can show how the material is being removed, so that the actual geometry generated can also be seen in these systems. This enables a fast detection of mistakes and their correction without loss of production time of the CNC machine tools. Many of these systems have the capability of showing the clamps and other elements, which are likely to interfere with the tool movement. Also, some systems have the capability of dynamically simulating the actual sized tool through the work material to make the simulation more realistic. A typical example is shown in Fig. 13.6, which shows visually the removal of material by the tool and consequent generation of the geometry from the blank geometry.

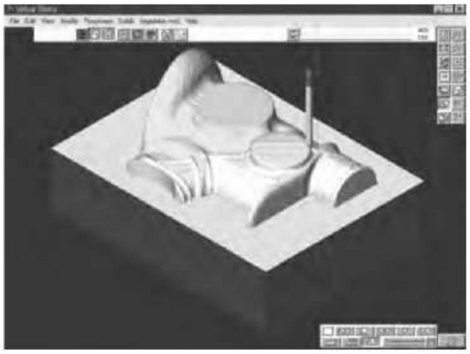


Fig. 13.6 *Shaded 3D image of the tool cutting the part for providing more realistic proving of the NC part program (Courtesy, Virtual Gibbs Inc.)*

Another simple method of verifying the program is that of plotting. However, it should be understood that this would give only a two-dimensional picture. The plotter is connected through an interface for obtaining the plot. The plot can be examined and compared with the component drawing for any error in the tool path.

13.1.9 Documentation for NC

It would now be amply clear that documentation is the most essential aspect of the CNC manufacturing practice. Therefore, it is worthwhile to list these as a checklist.

1. Component drawing.
2. *Process-planning sheet* As discussed earlier, this should contain details of the sequence of the operations, the machine tool used, the tools used with their numbers, speeds, feeds, etc.
3. *Tool cards* These should show each tool in assembled form with dimensions and identification numbers for each element (tool, collet, chuck, etc.).
4. *Set-up sheet* This would show all tools required, with their identification numbers, and the setting instructions for the component on the machine-tool table as shown in Fig. 13.7.
5. Programming sheets.
6. Punched paper tape, if this is the input form used.

The originals of these documents are kept in the programming room records cabinet while copies are sent to the shop floor as per the production planning. Whenever any changes are to be made, all issued copies are recalled and destroyed. The originals are updated (or made afresh) and copies are released accordingly.

13.2 MANUAL PART-PROGRAMMING METHODS

In the earlier days, a number of formats for NC part programs were used, such as fixed sequence or tab sequential. These systems required giving a large number of unwanted or duplicate information in each block of a part program. Thus, these are now replaced by means of a system called ‘Word Address Format’ in which

Set-up sheet

Op. no.	*Tool description*	*Catalogue number*	*Insert*	*Comment*
10	Slot drill HSS	VRT-123456		6 mm dia
20	Twist drill HSS	VRT 156756		12 mm dia
30	Twist drill HSS	VRT 234589		3 mm dia

Set-up instructions

The blank to be machined to the 80 × 80 mm size is to be mounted in a machine vice fixed to the machining centre table as shown.

Set-up sketch

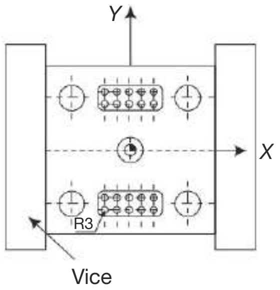


Fig. 13.7 *Set-up sheet for a machining centre part*

each of the information or data to be input in the form of numerical digits is preceded by a word address in the form of an English alphabet. For example, N105 means that N is the address for the numerical data 105. Thus, the controller can very easily and quickly process all the data entered in this format. A typical block of word address format may look as follows:

N115 G81 X120.5 Y55.0 Z-12.0 R2.0 F150 M3

13.2.1 ISO Standards for Coding

From the early years of development of numerical control, standardisation has been given due importance. As a result, many of the things that we use in NC are standardised and many of the manufacturers follow the standards to a great extent. One of the first things to be standardised is the word addresses to be used in programming. All the 26 letters of the English alphabet were standardised and given meanings as follows:

Character	**Address For**
A	Angular dimension around X axis
B	Angular dimension around Y axis
C	Angular dimension around Z axis
D	Angular dimension around special axis or third feed function*
E	Angular dimension around special axis or second feed function*
F	Feed function
G	Preparatory function
H	Unassigned
I	Distance to arc centre or thread lead parallel to X
J	Distance to arc centre or thread lead parallel to Y
K	Distance to arc centre or thread lead parallel to Z

L	Do not use
M	Miscellaneous function
N	Sequence number
O	Reference rewind stop
P	Third rapid traverse dimension or tertiary motion dimension parallel to *X**
Q	Second rapid traverse dimension or tertiary motion dimension parallel to *Y**
R	First rapid traverse dimension or tertiary motion dimension parallel to *Z**
S	Spindle speed function
T	Tool function
U	Secondary motion dimension parallel to *X**
V	Secondary motion dimension parallel to *Y**
W	Secondary motion dimension parallel to *Z**
X	Primary *X* motion dimension
Y	Primary *Y* motion dimension
Z	Primary *Z* motion dimension

* where D, E, P, Q, R, U, V, and W are not used as indicated, they may be used elsewhere.

The complete part program for a given component consists of a beginning code of % which signifies the start of the tape (in case of paper tapes) or beginning of a program if direct computer communication is involved such as in DNC mode. A part program consists of a large number of blocks (similar to sentences in a letter), each representing an operation to be carried out in the machining of a part.

Each block always starts with a block number used as identification and is programmed with an N word address. This must be programmed at the beginning of every block. As per ISO 2539, it has a minimum of three digits, e.g., N009, N028. However, some control manufacturers, notably Fanuc, dispense with this requirement. In their case, only those blocks which are to be specifically addressed as per the requirement of program flow would need to be given a block number. Other blocks can do away with this requirement. This saves valuable RAM space in the controller where the part programs are stored.

Each block can have one or more of the word addresses as explained above in a sequence. A typical ISO format for a block is shown below:

N5 G2 *X*±53 *Y*±53 *Z*±53 U..V..W..I..J..K..F5 S4 T4 M2 *

This shows a typical sequence in which the word addresses should occur in the block. However, it is not necessary that all these addresses should be present in each of the blocks or the sequence is important. The word addresses can occur in any sequence.

The numerical values immediately after word address indicate the maximum number of digits that are allowed for that particular address character. For example, G, the preparatory function is followed by two-digit information, say G00 to G99. The unsigned numbers indicate that the numerical value given would be without any sign. Also, a single digit indicates that the numerical value to be given is an integer.

When real values are to be given, two digits indicate them, the first one representing the number of digits before the decimal place while the latter is for those after the decimal place. For example, *X*±53 indicates that five digits before the decimal and three after it are needed to describe the word address *X*. The ± indicates that this address can be given with a sign. The + sign need not be given, since it is automatically assumed.

As per the standards followed, a decimal sign should not be given, its position being defined by the format specifications. However, many of the controllers allow a decimal point, and it is better for easy understanding to program directly the numbers with decimal point. In this book, all the dimensions are shown in decimal point deviating from the standard, but following the industry practice.

Since each function is indicated by its address character, the order of writing words in a block is not important except that the letter N should come right in the beginning and the end of the block should be placed where the information for that block is completed. Fanuc uses the end of block character as ';'. Others treat the 'Carriage Return' and 'Line Feed' as End of Block (EOB). In this book, we will use (*) as the end of block for easy understanding, though this is not required to be punched in the actual part program.

In the variable block format, the number of words and characters are variable, i.e., if any word is not required in any block then it need not be written and also if the value of any function remains the same in the next operation then it need not be repeated in the block. The following examples will clarify these details.

```
N110 G01 X-312.55 Y14.5 Z12.565 F200 S1500 T1103 M03 *
N115 Y187.0 Z0 *
```

In the block for the operation number 110, the value of functions G, X, F, S, T and M will be the same as in the operation number 115; only the values for the functions *Y* and *Z* will change to 187.0 and zero respectively. This feature of the word-address format, and also that the order of words is not important, makes the writing of programs very convenient.

Since the programming format for various control systems are not identical and may differ from ISO recommendations, it is important that the relevant programming manual should be consulted while preparing the program.

13.2.2 Coordinate function

As discussed above, the coordinates of the tool tip are programmed for generating a given component geometry. The coordinate values are specified using the word address such as X, Y, Z, U, V, W, I, J, K, etc. All these word addresses are normally signed along with the decimal point depending upon the resolution (at least 1 μm or less for precision CNC machine tools) available in the machine tool. Some examples are

```
X123.405 Y-34.450
```

13.2.3 Feed Function

Generally, the feed is designated in velocity units using the F word address. For example, F150 means that the feed rate is specified as 150 mm per minute. This is the actual speed with which the tool moves along the programmed path. However, depending upon the programmed path, there could be some deviations in the actual feed followed by the controller. Also, the controller calculates the actual feed rate of each of the axis.

Once the feed rate is programmed in a block, it remains in force in all the subsequent blocks till it is replaced by another F value, i.e., it is modal.

The feed rate programmed can be overridden by a setting on the controller console, in steps of 10% between 0 and 150%. However, in some situations this override will not work, for example, in case of thread cutting or thread milling.

By using an appropriate G code, it is also possible to change the feed rate units from mm per minute to mm per revolution or vice versa.

13.2.4 Speed Function

In some of the CNC machine tools, spindle speeds are set manually and so are not to be programmed. However, most of the CNC machines that are coming now have the capability for the stepless variation spindle speeds. Hence, they need to be programmed using the spindle speed word **S**.

The speed can be set directly in the revolutions per minute or RPM mode using the S word address as follows:

S1500 means, that spindle speed is to be set at 1500 rpm.

However, in some cases such as in turning centres, when the work surface controls the actual cutting speed then a direct RPM program would make the cutting speed vary whenever the work diameter changes. This is harmful from the surface finish as well as the tool-life point of view and hence another option for spindle-speed function is the constant surface speed. When this option is exercised, the spindle speed is specified not in RPM but in metres/minute or feet/minute depending upon the units chosen.

13.2.5 Tool Function

All NC machines are generally equipped with turrets or tool magazines with Automatic Tool Changers (ATC), which enable the positioning of the pre-set tools in a few seconds. Thus, the ratio of cutting time to total machine time is considerably increased.

The tool function is normally indicated by the word address T. This may have 2 or more digits depending upon the tool magazine capacity. Most general is 2 digits such as T15. This causes the tool magazine position 15 or tool number 15 to be brought into the spindle replacing the already present tool in the spindle. The tool replaced from the spindle is brought back to the empty position created when the tool 15 was loaded.

In machines where tool change is carried out manually, the word ‘T’ causes the stopping of the machine spindle and a light signal appears indicating to the operator that he has to carry out the tool change, the order of which he must already have been instructed about.

Tool offset, to be discussed later, can also be programmed by using the same T word, e.g., T1513 which means tool no. 15 (i.e., the tool located in the position 15 in the magazine) is to be loaded in the spindle and the value in offset register 13 is to be taken into account when this tool carries out the operation.

13.2.6 Comments

It is possible to add comments in the program to clarify the individual functions that are used in the program. For this purpose, parentheses are used. When the controller encounters the opening parenthesis then it ignores all the information till it reaches the closing parenthesis. An example is shown below to show the method used.

```
N010 G00 Z50.0 M05      (spindle stops and rapidly moves up)
N011 X0 Y0              (rapid move to start position 0,0)
N012 M30                (end of program and tape rewind)
```

13.3 PREPARATORY FUNCTIONS

This is denoted by ‘G’. It is a pre-set function associated with the movement of machine axes and the associated geometry. As discusses earlier, it has two digits, e.g., G01, G42, and G90 as per ISO specifications. However, some of the current-day controllers accept up to 3 or 4 digits. In this chapter, we will only discuss some of the regular functions. ISO has standardised a number of these preparatory functions, also popularly called G codes. The standardised codes are shown below:

CODE	FUNCTION
G00	Point-to-point positioning, rapid traverse
G01	Line interpolation
G02	Circular interpolation, clockwise (WC)

G03	Circular interpolation, anti-clockwise (CCW)
G04	Dwell
G05	Hold/Delay
G06	Parabolic interpolation
G07	Unassigned
G08	Acceleration of feed rate
G09	Deceleration of feed rate
G10	Linear interpolation for ‘long dimensions’ (10 inches–100 inches)
G11	Linear interpolation for ‘short dimensions’ (up to 10 inches)
G12	Unassigned
G13–G16	Axis designation
G17	*XY* plane designation
G18	*ZX* plane designation
G19	*YZ* plane designation
G20	Circular interpolation, CW for ‘long dimensions’
G21	Circular interpolation, CW for ‘short dimensions’
G22–G29	Unassigned
G30	Circular interpolation, CCW for ‘long dimensions’
G31	Circular interpolation, CCW for ‘short dimensions’
G32	Unassigned
G33	Thread cutting, constant lead
G34	Thread cutting, linearly increasing lead
G35	Thread cutting, linearly decreasing lead
G36–G39	Unassigned
G40	Cutter compensation—cancels to zero
G41	Cutter radius compensation—offset left
G42	Cutter radius compensation—offset right
G43	Cutter compensation—positive
G44	Cutter compensation—negative
G45–G52	Unassigned
G53	Deletion of zero offset
G54–G59	Datum point/zero shift
G60	Target value, positioning tolerance 1
G61	Target value, positioning tolerance 2, or loop cycle
G62	Rapid traverse positioning
G63	Tapping cycle
G64	Change in feed rate or speed
G65–G69	Unassigned
G70	Dimensioning in inch units

G71	Dimensioning in metric units
G72–G79	Unassigned
G80	Canned cycle cancelled
G81–G89	Canned drilling and boring cycles
G90	Specifies absolute input dimensions
G91	Specifies incremental input dimensions
G92	Programmed reference point shift
G93	Unassigned
G94	Feed rate/min (inch units when combined with G70)
G95	Feed rate/rev (metric units when combined with G71)
G96	Spindle feed rate for constant surface feed
G97	Spindle speed in revolutions per minute
G98–G99	Unassigned

Many of the control manufacturers follow these standard codes without altering the meaning. However, some manufacturers do change them to suit their way of programming.

It is generally possible to include more than one G address in one block, provided these functions are not mutually exclusive. For example, G02 and G03 (see details given later in this section) together in one block are normally not permissible. If they are given, the latter G code will become operational overriding the earlier from the same category. In Fanuc, controls up to 5 G codes can be given in one block. However, in MAHO Philips 532 system only one G code needs to be given. Though this makes the reading of the program easier, it unnecessarily increases the number of blocks in a program and the subsequent increase in the size of the part program.

Another aspect that one should normally remember is that some of the G codes are modal, which means that they behave as settings to the control. Once given they remain operational till cancelled by another G code from the same group. A few other G codes are non-modal, which means that they remain operational in the block in which they are programmed. When we are describing the G codes, it is made clear as to which is modal and not.

Also, some of the G codes are default or 'turn-on' codes. This means that they are operational when the controller is started. Also, when the program is completed, generally the controller is reset back to the original default settings. Hence care has to be taken by the programmer to understand the default codes in operation. This is also mentioned along with the G code description.

A few of the usual preparatory functions which are generally present in all machining centres and are uniformly followed by all controller manufacturers are given below:

Motion group

* G00	Rapid positioning
G01	Linear interpolation
G02	Circular interpolation clockwise
G03	Circular interpolation counter-clockwise

Dwell

G04	Dwell

Active plane selection group

* G17	*XY* plane selection
G18	*XZ* plane selection
G19	*YZ* plane selection

Cutter compensation group

* G40	Cutter compensation, cancel
G41	Cutter radius compensation left
G42	Cutter radius compensation right

Units group

* G70	Inch units
G71	Metric units

Hole-making canned cycle group

* G80	Canned cycle cancel
G81–G89	Canned cycles definition and ON

Coordinate system group

* G90	Absolute coordinate system
G91	Incremental coordinate system

Pre-set

G92	Absolute pre-set, change the datum position

The * sign indicates the generally accepted default or turn-on code in operation. However, some control manufacturers allow this to be modified to whatever suits them. The above is only a possible indication but not in any way standardised by ISO.

In the following, we will give a description of the way to use these G codes.

13.3.1 Coordinate System Group, G90 and G91

The input of dimensional information can be done either in the absolute or in the incremental system. The preparatory function G90 is used for absolute programming. In the absolute system, the dimensions are given with respect to a common datum chosen by the programmer. It must be programmed and can be cancelled by the function G91 (and also when the program statement has the word M02 or M30). In Fig. 13.8, *OX* and *OY* are the datum.

Whatever may be the route of the move, the programmed *X* and *Y* values of each position remain the same. Suppose the route is 0-*A*-*B*-*C*. Then

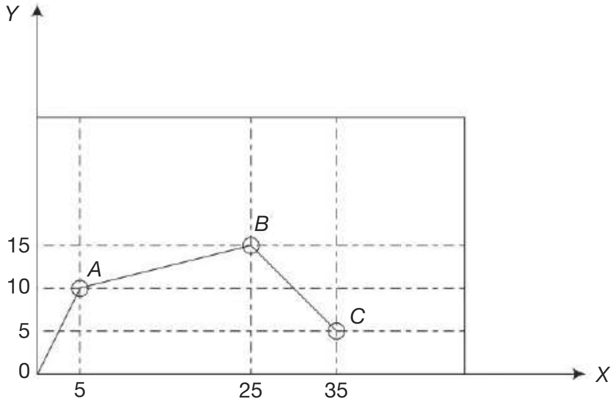


Fig. 13.8 *Absolute (G90) and incremental (G91) systems*

N007 G90 G01 X5.0 Y10.0 F200
N008 X25.0 Y15.0
N009 X35.0 Y5.0

This system is generally advisable for programming, as there are few chances of errors. When the tool is in a particular quadrant, such as the first, then all the coordinate values are positive; similarly, with all the other quadrants of geometry. Thus, it is suggested that the new programmers should always make use of the absolute system till they become familiar with the programming system.

The incremental type, denoted in the program by the word G91, is also available on all NC equipments. This is generally the 'Turn-on mode' and can be cancelled by the word G90. The end of the program words M02 or M30 also sets it. In the incremental system, the data is incremental to the previous block. Unlike as with G90, the programmed data changes only if the route of the move is altered. Referring to the same figure (Fig. 13.8), the program for route 0-*A*-*B*-*C* is

N007 G91 G01 X5.0 Y10.0 F200
N008 X20.0 Y5.0
N009 X10.0 Y-10.0

If the route is changed then the program changes accordingly. Incremental programming is useful when the features are dimensioned in a continuous chain, e.g., 5 holes, 31.250 mm apart along the *X* axis would simply be programmed for each feature as X31.25; while in absolute programming, one would have to calculate the value for each position, i.e., X31.25, X62.5, X93.75,... It is also important while one follows the incremental programming system, to take care of the direction in which the movement is taking place, irrespective of the quadrant in which the tool is moving.

In a program, both the systems may be followed but it should be done carefully. In the incremental system, any error done in a single block is carried forward and no correction can be done. Also, the errors in the transmission system result in having the errors accumulated, while that does not happen in the case of an absolute coordinate system.

13.3.2 Units Group, G70, G71

This group of codes specifies the units in which the program is to be interpreted. G70 stands for programming in inch units while G71 stands for programming in mm units. Any one of these can be made as turn-on code depending upon the default units likely to be used. Most of the controls destined for areas other than North America generally have a default G71. This can be easily changed when necessary. In any case, it is a better practice to make the habit of giving this code as the very first code in the part program.

A given program should be written only in either inch or mm units, but not both. Hence, only one of the two codes should be present in one program. In Fanuc controls, normally G20 and G21 are used for the units in place of G70 and G71. This can be changed optionally to 70 and 71 using the programmable functions of the control system.

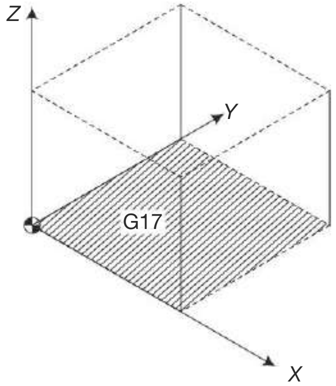


Fig. 13.9 *XY plane selection for vertical-axis milling machines*

13.3.3 Active Plane-Selection Group, G17, G18, G19

Some of the functions in NC control can only work in a plane rather than in all the 3 possible coordinate axes. This, therefore, requires the selection of an active plane. This can be done by using these codes. The typical coordinate system and the corresponding plane labelling are shown in Fig. 13.9.

G17 XY Plane Selection This is the default turn-on code. This allows for the working to be carried out in the horizontal plane in case of vertical-axis milling machines as shown in Fig. 13.9. In the case of 2.5 axes machines, in a given block only *X* and *Y* coordinates are to be specified while the *Z* coordinates are to be specified in a separate block. For a horizontal-axis machine, the working plane is the vertical plane perpendicular to the spindle axis. Similarly, *XY* plane selection for horizontal-axis machining centres is shown in Fig. 13.10.

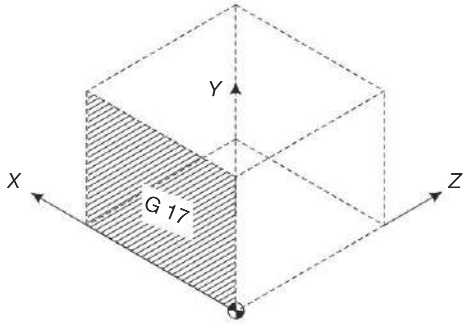


Fig. 13.10 *XY plane selection for horizontal-axis milling machines*

G18 XZ Plane Selection This allows the working to be carried out in the *XZ* plane (Fig. 13.11). In the case of 2.5 axes machines, in a given block only *X* and *Z* coordinates are to be specified while the *Y* coordinates are to be specified in a separate block.

G19 YZ Plane Selection This allows the working to be carried out in the *YZ* plane (Fig. 13.12). In the case of 2.5 axes machines, in a given block only *Y* and *Z* coordinates are to be specified while the *X* coordinates are to be specified in a separate block.

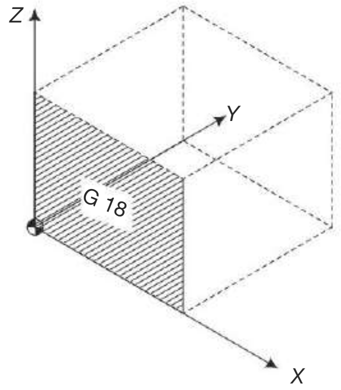


Fig. 13.11 *XZ plane selection for horizontal-axis milling machines*

13.3.4 Preset, G92

As described earlier, each of the machine tools has a separate machine reference point. However, this point is not very convenient to use as a coordinate datum for the part. Most of the NC machine tools allow for a 'floating datum' to be fixed anywhere in the machining envelope of the machine tool. As a result, the programmer can choose a convenient position on the part as datum, which may be referred to as 'program zero point'. The same will have to be communicated to the NC controller as datum. The choice of the datum as explained earlier is to suit either the setting of the component or to simplify the coordinate calculations.

It is necessary in the beginning to make the system understand the coordinate datum position of the part, which is different from the machine reference point. To do this, we make use of the G92 code. The part, which was pre machined, is clamped at a suitable position on the machine table. A known tool or a setting mandrel of known diameter is kept in the machine spindle. This tool is then brought to a known position near or on the workpiece blank, called the set point. The same is then programmed in the part program using the G92 code. For example, in Fig. 13.13, the workpiece of dimensions 200 × 170 × 30 mm is located on the machine bed with the longer edge along the *X*-axis. The tip of

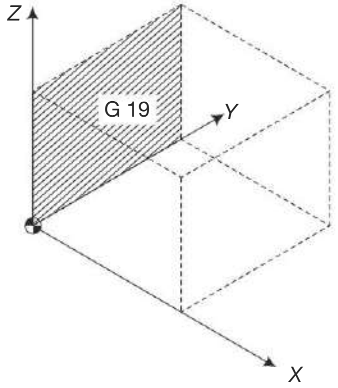


Fig. 13.12 *YZ plane selection for horizontal-axis milling machines*

the setting tool, held in the spindle, is made to touch the point *A*, i.e., the setting position. At that instant, the program block entered is

N015 G92 X200.0 Y170.0 Z50.0

Depending upon the point being touched, the coordinate can be specified, taking the diameter of the probe touching the workpiece. The tool tip is to be set at a distance of 50 from the top surface by means of a suitable gage.

G53 to G56 are the other codes used for setting the programmable datum positions. These allow for fixing a number of positions on the machine table whose coordinates can be entered into the controller as a permanent memory. When required, their positions can be simply called by giving the particular G code in the program. This is also be useful for machining a batch of components, all of which are located on the machine table each at the positions indicated by G53, G54, etc.

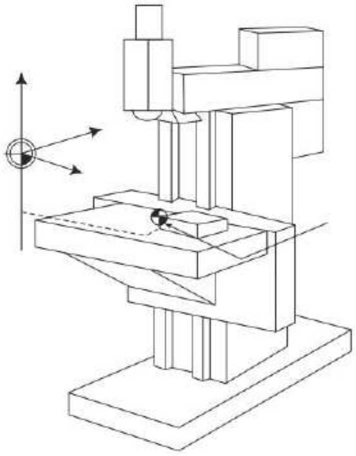


Fig. 13.13 *Setting the workpiece on the machine table*

13.3.5 Motion Group, G00, G01, G02, G03

This is the most important group of codes used in part programming. G00 is the turn-on code from this list. More explanation is given below:

Rapid Positioning, G00 This is used for moving the tool at a rapid rate (normally, the maximum available feed rate such as 8000 or 40 000 mm/min) along the axes involved for achieving the position programmed. The path taken by the tool to reach the programmed point is not important for this code.

This is a modal (stays active till cancelled by any other function of its family, i.e., G01, G02, G03) function and is also the 'turn-on mode' (available as soon as the system is switched on or when a new program starts). Referring to Fig. 13.14, from the position *A*, it is required to achieve the position *B*.

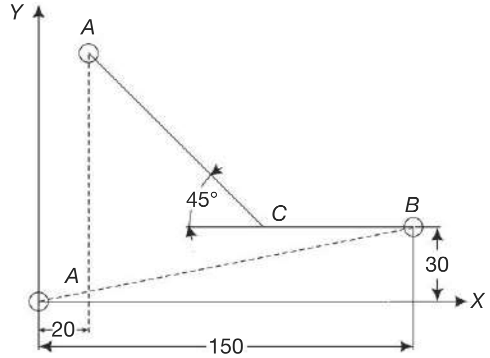


Fig. 13.14 *Positioning, preparatory function G00*

This is typical to all machining situations when the tool has to be brought close to the component before any cutting commences. It is obvious that this movement is in the air (cutting air) and so to minimise the idle time, it should take place at the maximum feed rate of various slides involved. For this, the program block would be

N105 G90 G00 X150.0 Y30.0

It will be noticed that the initial path is at 45 degrees because both the *X* and *Y* slides move at the same feed rate (assuming the motors are of the same rating) till the required *Y* ordinate value is achieved, after which only the *X* slide moves till the position *B* is achieved. This is one way of achieving the final position. There could be other possible methods implemented by different controllers.

The rapid positioning is actually a 3D positioning, such that positioning can be achieved simultaneously in all the 3 axes as shown in Fig. 13.15.

Absolute programming A to B

```
N110 G90 G00 X50.0 Y45.0 Z 40.0
N120 X90.0 Y90.0 Z70.0
```

Incremental programming A to B

```
N110 G90 G00 X50.0 Y45.0 Z 40.0
N120 G91 X40.0 Y45.0 Z30.0
```

Incremental programming B to A

```
N110 G90 G00 X90.0 Y90.0 Z 70.0
N120 G91 X-40.0 Y-45.0 Z-30.0
```

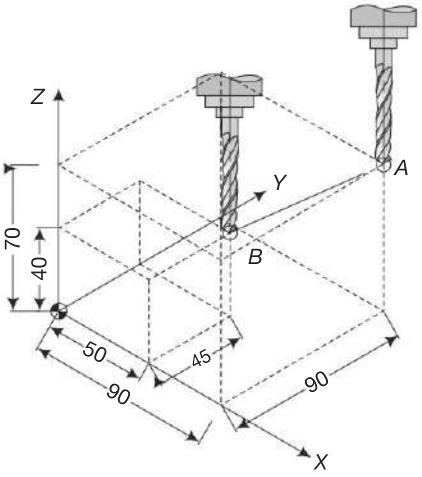


Fig. 13.15 *Positioning, preparatory function G00 in 3 dimensions*

Linear or Straight line Interpolation, G01 This code is generally used when the material is to be cut using a feed rate. When the motion is desired along a straight line at a given feed rate, this function is used. It is modal. If a cut has to be made from *D* to *E* (Fig. 13.16) at a feed rate of 250 mm per minute, then the block would be

```
N115 G01 X110.0 Y30.0 F250
```

In this case, the controller moves all the three axes at a rate such that the resultant velocity along the line matches the programmed feed rate.

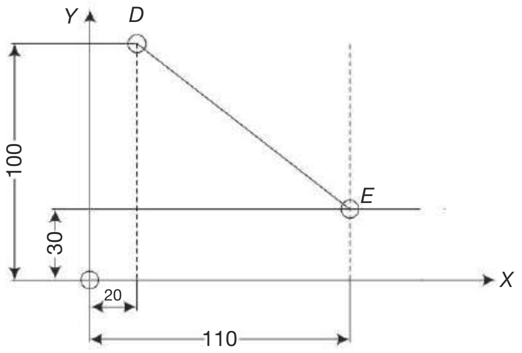


Fig. 13.16 *Linear interpolation, preparatory function G01*

Similarly, for the motion command shown in Fig. 13.15,

Absolute programming A to B

```
N110 G90 G00 X50.0 Y45.0 Z 40.0
N120 G01 X90.0 Y90.0 Z70.0 F350
```

Incremental programming A to B

```
N110 G90 G00 X50.0 Y45.0 Z 40.0
N120 G91 G01 X40.0 Y45.0 Z30.0 F350
```

Incremental programming B to A

```
N110 G90 G00 X90.0 Y90.0 Z 70.0
N120 G91 G01 X-40.0 Y-45.0 Z-30.0 F350
```

Circular Interpolation, G02 / G03 When an arc is to be traversed in a plane, the function G02 or G03 is used if the direction of the motion is clockwise or anti-clockwise respectively, looking in the negative direction of the axis perpendicular to the plane. Referring to Fig. 13.17, when the motion is from the *F* to *G* in the *XY* plane, the program block would be, as per ISO,

```
N125 G02 X65.0 Y60.0 I35.0 J-10.0 F250
```

But, if the motion were from *G* to *F* then it would be

```
N130 G03 X15.0 Y30.0 I-15.0 J-40.0 F250
```

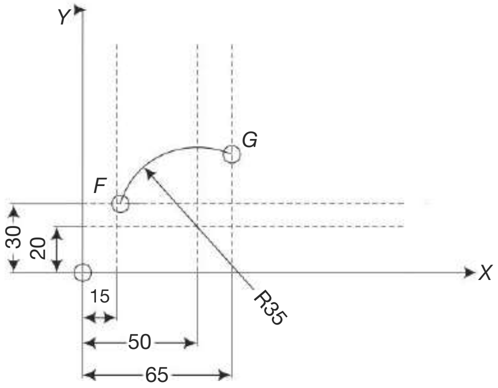


Fig. 13.17 *Circular interpolation, preparatory function G02/G03*

Here, (X, Y) are the coordinates of the destination and (I, J) the distances, along the reference axes of the centre of the arc from the starting point of the arc. It is essential that the coordinates of the destination should be correct and within the prescribed limits. In some systems, (I, J) are the coordinates of the centre of the arc.

Some systems carry out circular interpolation when the value of the arc radius is given, being positive if the angle subtended by the arc at the centre is less than 180 degrees, and negative, if otherwise. Assuming the radius and the angle subtended to be 40 mm and 100 degrees respectively then

N130 G02 X65.0 Y60.0 R40.0 F250

It is possible to draw a complete circle which would mean that the destination coincides with the starting point. Then

N310 G02 I35.0 J-10.0 F250

It will be noted that the destination need not be stated, since its coordinates are the same as that of the starting point already entered in the previous block. One should check up from the controller programming manual as to which procedure is to be followed. It may be noted that a full circle cannot be obtained with R-value.

The circular interpolation is 2D interpolation and can only be carried out in any plane. For example, in Fig. 13.18 is shown a typical circular slot in the XY-plane which can be machined using a slot drill in a vertical-axis milling machine. This will be the most common form of usage. However, sometimes it may be necessary to machine circular profiles in other planes as well. An example is shown in Fig. 13.19 for the XZ plane. In such cases, it becomes necessary to specify the plane to be used with G17, G18 or G19 codes.

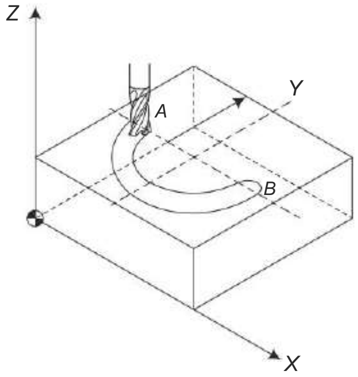


Fig. 13.18 *Circular interpolation in XY plane using G17 plane selection*

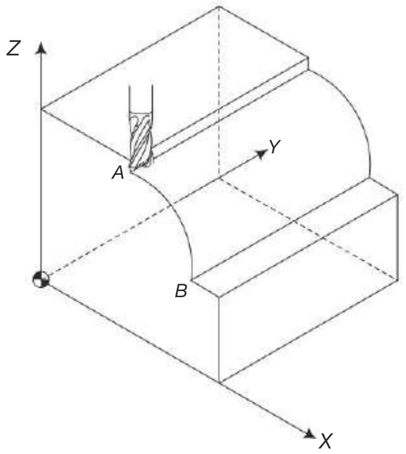


Fig. 13.19 *Circular interpolation in XZ plane using G18 plane selection*

Dwell, G04 This is to give a delay in the program. When the G04 code is encountered, the controller stops at that particular point for a specified time mentioned in the block. After that time, the controller continues to execute the next block in the program. The delay time is normally mentioned in seconds using the X word address. In some controls, a word other than X may also be used. For example, in Fanuc controls, the P word address is used to specify the dwell time in milliseconds. An example is shown below:

N045 G04 X3.0	This calls for a stoppage of the control for a period of 3 seconds.
N045 G04 P3000	This calls for a stoppage of the control for a period of 3 seconds. No decimal point programming with P word address in Fanuc controls.

13.4 MISCELLANEOUS FUNCTIONS, M

These functions actually operate some controls on the machine tool and thus affect the running of the machine. Generally, only one-M code is supposed to be given in a single block. However, some controllers allow for two or more M codes to be given in a block, provided these are not mutually exclusive, e.g., coolant ON (M07) and OFF (M09) cannot be given in one block.

Less number of M codes have been standardised by ISO compared to G codes in view of the direct control exercised by these on the machine tool. The ISO standard M codes are shown below:

CODE	FUNCTION
M00	Program stop, spindle and coolant off
M01	Optional programmable stop
M02	End of program—often interchangeable with M30
M03	Spindle on, CW
M04	Spindle on, CCW
M05	Spindle stop
M06	Tool change
M07	Coolant supply No. 1 on
M08	Coolant supply No. 2 on
M09	Coolant off
M10	Clamp
M11	Unclamp
M12	Unassigned
M13	Spindle on, CW + coolant on
M14	Spindle on, CCW + coolant on
M15	Rapid traverse in + direction
M16	Rapid traverse in - direction
M17–M18	Unassigned
M19	Spindle stop at specified angular position
M20–M29	Unassigned
M30	Program stop at end tape + tape rewind
M31	Interlock by-pass
M32–M35	Constant cutting velocity
M36–M39	Unassigned
M40–M45	Gear changes; otherwise unassigned
M46–M49	Unassigned
M50	Coolant supply No. 3 on
M51	Coolant supply No. 4 on

M52–M54	Unassigned
M55	Linear cutter offset No. 1 shift
M56	Linear cutter offset No. 2 shift
M57–M59	Unassigned
M60	Piece part change
M61	Linear piece part shift, location 1
M62	Linear piece part shift, location 2
M63–M67	Unassigned
M68	Clamp piece part
M69	Unclamp piece part
M70	Unassigned
M71	Angular piece part shift, location 1
M72	Angular piece part shift, location 2
M73–M77	Unassigned
M78	Clamp non-activated machine bed-ways
M79	Unclamp non-activated machine bed-ways
M80-M99	Unassigned

Some of the common miscellaneous functions often found in many a controller are the following:

M00 This terminates the auto-operation of the machine after completing the instructions in the block in which it has been programmed. This is called ‘program stop’ and if it is required to continue with the rest of the program, the ‘start’ button on the console is to be pressed. This is useful for changing the clamp position or to carry out inspection of a particular dimension after a machining cut is taken. This being a pause function, and calls for the attention of the operator, delays the completion of the program and therefore should be avoided as far as possible.

M01 This is ‘optional stop’ and stops the machine, as in the case of M00, only if the ‘optional stop’ switch on the controller console is ‘ON’. This is useful when inspection is to be carried out on some components and not all in a given batch.

M02 This is ‘end of program’ and it causes the stopping of the machine and clearing of all the control registers. Another code M30 also does the same function.

M03 The miscellaneous function for machine spindle control for clockwise rotation. This starts the spindle to move in the clockwise direction at the speed set earlier using the S word address. When it is given in a block, it is the first code to be executed before all the other codes in a block are acted upon.

M04 The miscellaneous function for machine-spindle control for counter-clockwise rotation. This starts the spindle to move in the counter-clockwise direction at the speed set earlier using the S word address. When it is given in a block, it is the first code to be executed before all the other codes in a block are acted upon.

M05 The miscellaneous function for stopping the machine spindle. When it is given in a block, it is the last code to be executed after all the other codes in a block are acted upon.

M06 Is for tool change.

M07 Are for ‘Coolant 1 On’

M08	Are for 'Coolant 2 On'
M09	Is for 'Coolant Off'.
M13	The miscellaneous function for machine-spindle control for clockwise rotation and the starting of the coolant simultaneously. This starts the spindle to move in the clockwise direction at the speed set earlier using the S word address. When it is given in a block, it is the first code to be executed before all the other codes in a block are acted upon.
M30	It is similar to M02. It indicates 'end of tape' and 'tape rewind'. If a paper tape is used, the tape is rewind till the % sign is encountered. For machines working with RAM, the active program comes to the beginning. Many a times, M02 and M30 are synonymous in operation in modern-day controllers.

13.5 PROGRAM NUMBER

In many of the latest CNC systems, there is a provision for labelling the program at the start itself which facilitates searching from stored programs. The symbol used for the program number in Fanuc controls is 'O' or ':', followed by its number. For example, O238 or :238. Such information does not interfere with the NC program.

Invariably, in most of the components there are a number of repetitive features, e.g., pattern of holes, profiles, etc. Instead of writing blocks for each of them repeatedly in the program as per process plan, the facility exists for writing the subprograms for each feature and entering them with labels at the end of the main program. In the main program, where these are required, they are called by an appropriate block, e.g., in Fanuc controls, M98 P1001, i.e., miscellaneous function 98 is call for subprogram (also called subroutine), the number after 'P' referring to the subprogram being called. The subprograms are ended with the word 'M99' in Fanuc controls.

Example 13.1 The component to be machined is shown in Fig. 13.20. It is assumed that the pocket is through and hence only the outside is to be machined as a finish cut of the pocket. The tool to be used is a 20 mm diameter slot drill. If an end mill is to be used, the program should be modified with a hole to be drilled at B first before the end mill is used. The setting is done with point the *A* as reference (0, 0, 0) and the reference axes are along *X* and *Y* directions. A typical program, as per ISO (except the decimal point), for this would be

```
N001 G92 X0 Y0 Z0                     absolute presetting at A
N002 G90                              absolute programming
N003 G00 X25.0 Y25.0 Z2.0 T01 S3000 M03   tool brought rapidly at B, 2 mm above XY plane
N004 G01 Z-12.0 F120                  tool goes down to full depth
N005 Y75.0                            proceeds to C
N006 X65.0                            proceeds towards right to D
N007 G02 Y25.0 I0 J-25.0              cuts curved profile till E
N008 G01 X25.0                        proceeds to B
N009 Z2.0                             tool moves 2 mm above the XY plane
N010 G00 Z50.0 M05                    spindle stops and rapidly moves up
N011 X0 Y0                            rapid move to start position 0,0
N012 M30                              end of program and tape rewind
```

The graphical simulation of the above is shown in Fig. 13.21, simulating the material-removal process on the machine tool.

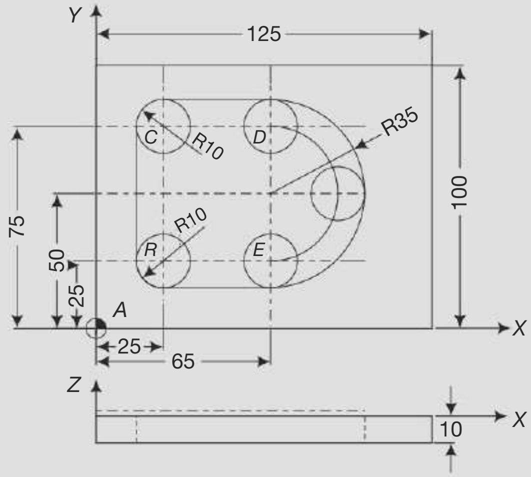


Fig. 13.20 *Example 13.1*

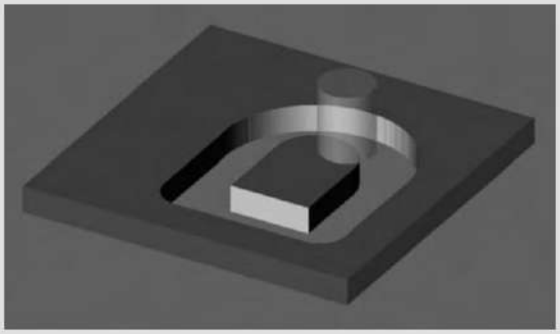


Fig. 13.21 *Simulation of the above part program for the component shown in Fig. 13.20*

Example 13.2 The component to be machined is shown in Fig. 13.22. The outer profile needs to be machined using a slot drill of ϕ16 mm.

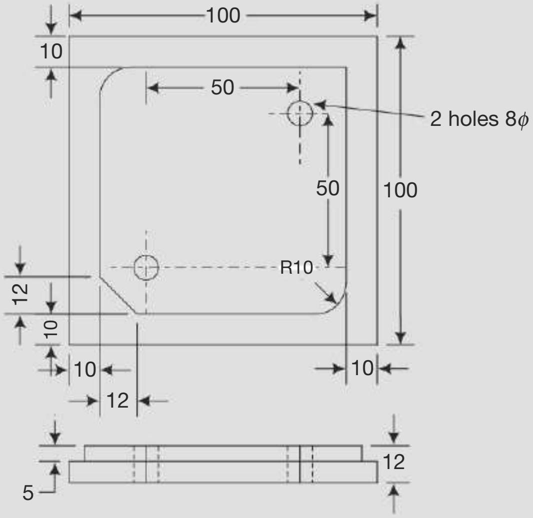


Fig. 13.22 *Component for Example 13.2*

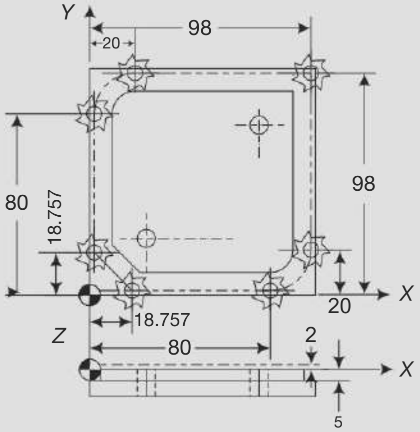


Fig. 13.23 *Tool path for machining the component for Example 13.22*

The axes system chosen for the component is given in Fig. 13.23. The path to be taken by the tool is shown as the centreline. The complete part program with all the necessary M codes is given below:

```
%
01002                                    (Program number)
N010 G71 G92 X0 Y0 Z50                   (Set point)
N015 G90                                 (Absolute programming)
N020 T01 S500 M06                        (Tool change speed setting)
N025 G00 Z2.0 M03                        (Rapid move to clearance plane)
N030 G01 Z-5.0 F100                      (Feed to the required depth)
N035 Y80.0 F120                          (Cut along straight line)
N040 G02 X20.0 Y98.0 R20.0 F100          (Circular move)
N045 G01 X98.0                           (Cut along straight line)
N050 Y20                                 (Cut along straight line)
N055 G02 X80.0 Y2.0 R18.0                (Circular move)
N060 G01 X18.757                         (Straight line to the intersection point)
N065 X2.0 Y18.757                        (Straight line to the intersection point)
N070 Z2.0 M05                            (Feed to clearance plane stop spindle)
N075 G0 X0 Y0 Z50                        (Rapid to set point)
N080 M02                                 (End of program)
```

The reader would have noticed that the tool has been moved along a path which is offset from the original contour by a value equal to the radius of the cutter. Further, it became necessary to calculate the new intersection points which sometimes require the use of trigonometry. For example, see the path taken by a cutter of radius *R* as shown in Fig. 13.24. To calculate the actual cutter path, it is necessary to evaluate the ΔX and ΔY using the following formulae.

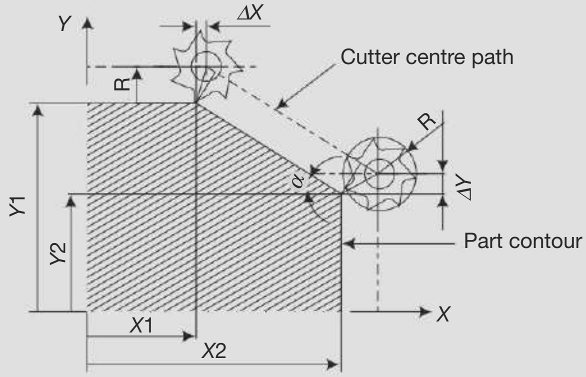


Fig. 13.24 *Offset Tool path for machining contours that are not parallel to the principal axes*

$$\Delta Y = R \tan\left(\frac{\alpha}{2}\right)$$

$$\Delta X = R \tan\left(\frac{90 - \alpha}{2}\right)$$

It is possible to derive similar formulae for other contour situations as well. Further, most of the new controllers are provided with cutter radius compensation options. In such cases, it is not necessary to calculate such intersection values. The details are given later.

The graphical simulation of the above is shown in Fig. 13.25, simulating the material-removal process on the machine tool.

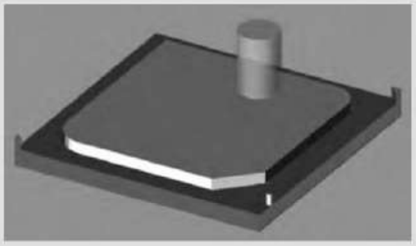


Fig. 13.25 *Simulation of the above part program for the component shown in Fig. 13.22*

13.6 TOOL-LENGTH COMPENSATION

In the programs discussed so far, only one tool was used to perform the machining function. Hence, the workpiece setting is performed with the single tool. However, in cases where there is more than one tool, programming becomes cumbersome, if the programmer has to take care of the individual tool lengths for the purpose of programming the *Z* depth in each case.

In NC practice, all tools are measured in the assembled state using a tool pre-setter as explained in Chapter 11, and this information is always kept up to date (Fig. 13.26). For the tools being used, the difference in length, with respect to the pre-setting tool, is recorded and is manually entered and stored with the associated tool number. Whenever these tools are called into action by programmed instruction, the respective compensation values are activated and automatically taken into account in the tool motion. The following program example explains how tool-length compensation is automatically taken care of.

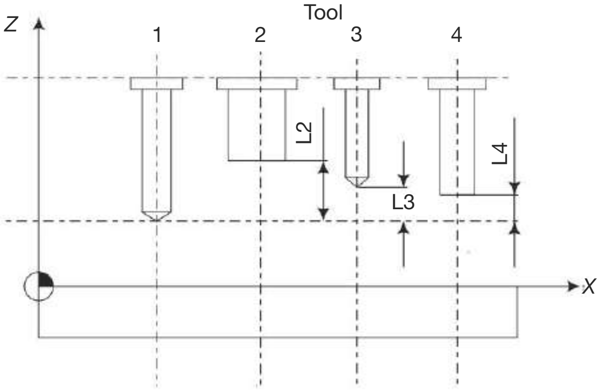


Fig. 13.26 *Tool-length compensation*

```
N003 M06 T01

N006 M06 T02
```

In these program blocks, M06 refers to tool change and T01, T02, ... refer to the tools to be loaded. Whenever the tool is ground or replaced, the new values are entered to replace the earlier ones and thus the program remains unchanged. This is an essential facility, without which the multiplicity of the programs for each job/tool combination is enormous and futile. It will be understood that the values entered compensate for the difference in lengths and thus all tools 'effectively' become independent of tool dimensions, if the dimensions of all the tools are stored. When programming, the tool dimensions are not considered since the compensation values are calculated by the control system itself during manufacturing.

13.7 CANNED CYCLES

It is found many a times that a series of motions are to be repeated a number of times, many of which are fairly common to all the positions. For example, in the case of a drilling operation, the tool (twist drill) has to position a little above the hole in rapid position, then move to the required depth with the given feed rate and then the tool has to return to the top of the hole as shown in Fig. 13.27.

The same actions are to be repeated for each of the holes. For each of the operations, 3 NC blocks to be written, out of which two blocks need to be repeated without any change for each of the holes to be drilled in the same plane. It, therefore, is possible to define a canned cycle or fixed cycle which can repeat all these motions without having to repeat the same information for each of the holes. The most common cycles that are useful are for the hole-making operations such as drilling, reaming, tapping, etc. The advantages to be derived in using the canned cycles can be gauged by looking at the part programs below for the component shown in Fig. 13.28.

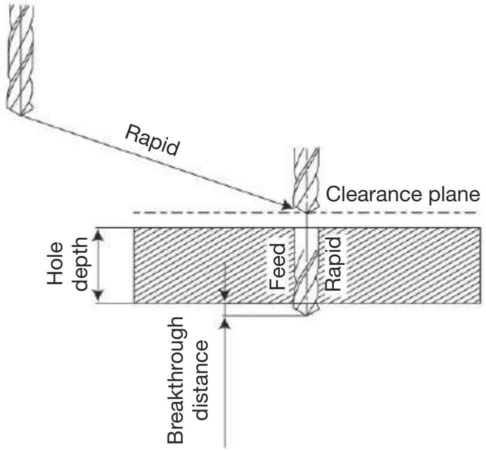


Fig. 13.27 *Typical motions embedded in G81 canned cycle*

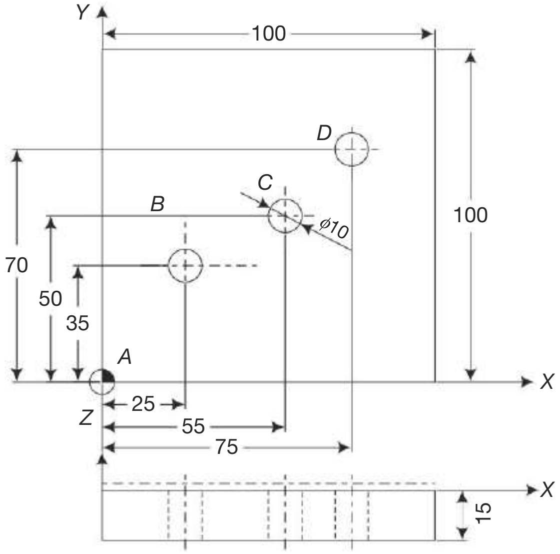


Fig. 13.28 *Example for canned cycles*

For the component shown in Fig. 13.28, the NC program for drilling the three holes without using canned cycles is shown below:

```
N010 G00 X25.0 Y35.0 Z2 *
N015 G01 Z-18.0 F125 *
N020 G00 Z2.0 *
N025 X55.0 Y50.0 *
N035 G01 Z-18.0 F125 *
N040 G00 Z2.0 *
N045 X75.0 Y70.0
N050 G01 Z-18.0 F125 *
N055 G00 Z2.0 *
N065 X0 Y0 Z50 *
```

For the same component, the NC program using canned cycles is shown below:

```
N010 G81 X25.0 Y35.0 Z-18.0 R2.0 F125 *
N015 X55.0 Y50.0 *
N020 X75.0 Y70.0 *
N025 G80 X0 Y0 Z50 *
```

In the canned cycles, the additional data such as the clearance plane position has to be specified. For example, the G81 canned cycle is used for carrying out the drilling operations for through holes. The actual operations embedded in the G81 canned cycle are shown in Fig. 13.27. The format to be used as follows:

N G81 X Y Z R

X, Y	Refer to the centre coordinates of the point where the drilling is to be carried out.
Z	Refer to the final depth of the hole to be drilled
R	Refer to the position of the clearance plane (same as the *Z*-axis position of the clearance plane)

Table 13.2 *Standard canned cycle motions*

Canned cycle number	*Feed from surface*	*At programmed depth (end of feed point)*			*Used for*
		Dwell	*Spindle speed*	*Spindle return motion*	
G80	Off	—	Stop	—	Cancel canned cycle
G81	Constant	—	—	Rapid	Drilling, centre drilling
G82	Constant	Yes	—	Rapid	Counter sinking, counter boring
G83	Intermittent	—	—	Rapid	Deep-hole drilling
G84	Constant	—	Reverse	Feed	Tapping
G85	Constant	—	—	Feed	Reaming
G86	Constant	—	Stop	Rapid	Boring
G87	Constant	—	Stop	Manual	Multiple boring
G88	Constant	Yes	Stop	Manual	Boring
G89	Constant	Yes	—	Feed	Boring

G81 is a canned cycle, which is modal. As a result, for any point specified when G81 is in force the drilling action will be repeated. As a result, to cancel the canned cycle, it is necessary to use the code G80. The effect of G80 is to cancel the existing canned cycle in force and make G00 operational. Any coordinates to which the spindle should move can be programmed in the G80 block.

There are a number of canned cycles identified by ISO in this category. The typical motions embedded in various canned cycles as per ISO are shown in Table 13.2.

Example 13.3 The component to be machined is shown in Fig. 13.29. Write a program using canned cycles to drill all the holes shown in Fig 13.29.

```
%
O1303
N010 G71 G92 X0 Y0 Z50
N015 G90
N020 T01 S800 M06
N025 G82 X25 Y25 Z-5 R2 P2 F125 M03      (Centre drill all holes)
N030 X25 Y75
N035 X75 Y25
N040 G80 X-50 Y0 Z50 M05
```

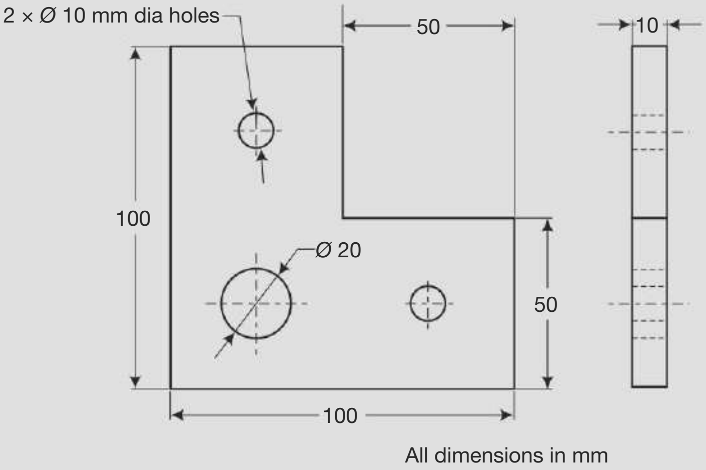


Fig. 13.29 *Component for NC program in Example 13.3*

```
N045 T02 S800 M06                        (Finish drill the φ 20 hole)
N050 G81 X25 Y25 Z-13 R2 F125 M03
N055 G80 X-50 Y0 Z50 M05
N060 T03 S800 M06                        (Finish Drill the φ 10 holes)
N065 G81 X25 Y75 Z-13 R2 F125 M03
N070 X75 Y25
N075 G80 X-50 Y-50 Z50 M05
N080 T00 M05
N090 M02
```

The graphical simulation of the Example 13.3 is shown in Fig. 13.30, simulating the material-removal process on the machine tool.

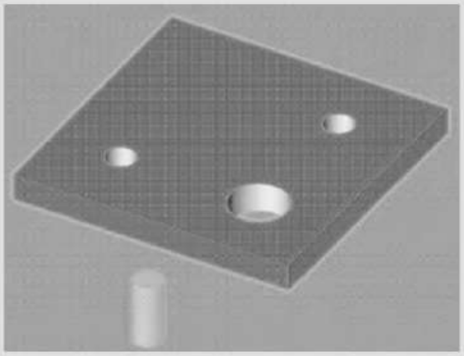


Fig. 13.30 *Simulation of the program for Example 13.3*

Example 13.4 The component to be machined is shown in Fig. 13.31. Write a program using canned cycles to drill all the holes shown in Fig 13.31.

```
%
O1303
N010 G71 G92 X0 Y0 Z50
N015 G90
N020 T01 S800 M06
N025 G82 X15 Y15 Z-5 R2 P2 F125 M03     (Centre drill all holes)
N030 X135 Y15
N035 X75 Y50
```

```
N030 X135 Y85
N035 X15 Y85
N040 G80 X-50 Y0 Z50 M05
N045 T02 S800 M06                     (Finish drill the  20 hole)
N050 G81 X75 Y50 Z-13 R2 F125 M03
N055 G80 X-50 Y0 Z50 M05
N060 T03 S800 M06                     (Finish drill the  10 holes)
N065 G81 X15 Y15 Z-13 R2 F125 M03
N070 X135 Y15
N035 X135 Y85
N030 X15 Y85
N075 G80 X-50 Y-50 Z50 M05
N080 T00 M05
N090 M02
```

The graphical simulation of the Example 13.4 is shown in Fig. 13.32, simulating the material-removal process on the machine tool.

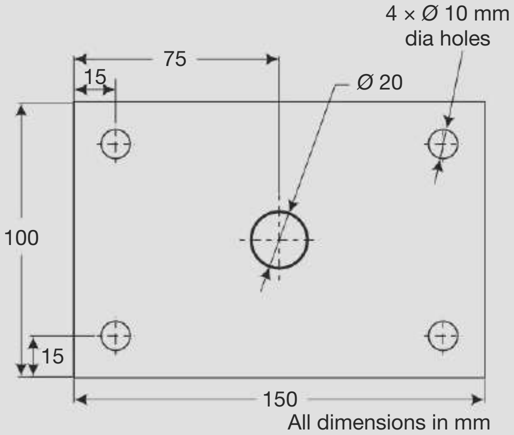


Fig. 13.31 *Component for NC program in Example 13.4*

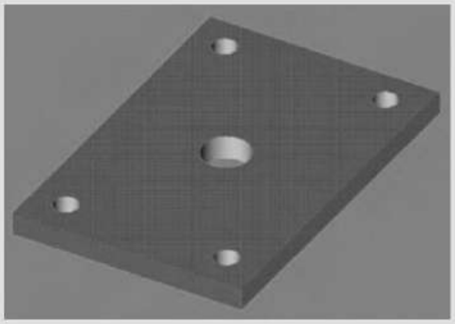
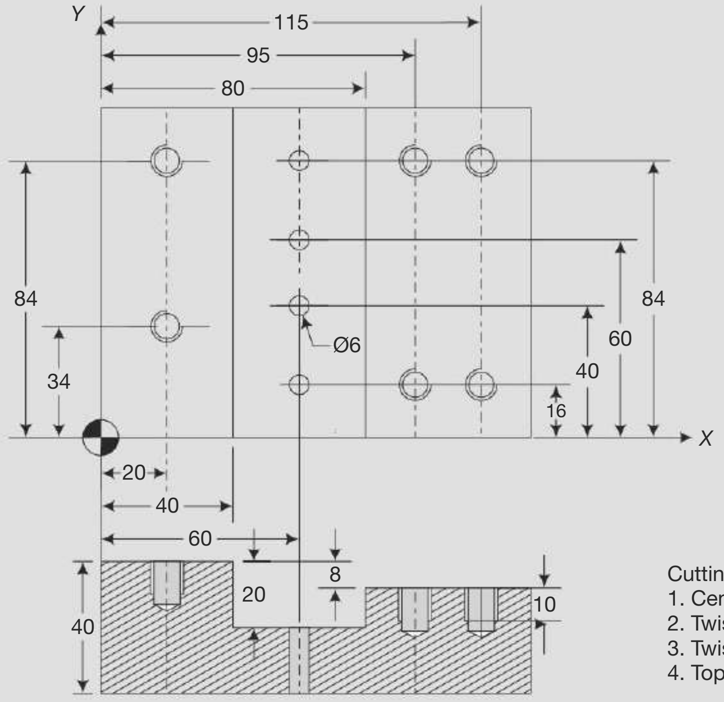


Fig. 13.32 *Simulation of the program for Example 13.4*

Example 13.5 The component to be machined is shown in Fig. 13.33. All the hole-making operations are to be machined using the tools specified.

The point to be noted in the above component is that the top surface is at three different levels as such, the clearance plane will be different for each of these planes. Hence, the part program will have to take this into account while using the appropriate canned cycle. The following part program machines the part completely.



Cutting tool:
1. Center drill
2. Twist drill Ø 6 mm
3. Twist drill Ø 8.5 mm
4. Top M10

Fig. 13.33 *Component for NC program in Example 13.5*

```
%
O1245
N010 G71 G92 X0 Y0 Z50
N015 G90
N020 T01 S800 M06                   (Centre drill all holes)
N025 G82 X20 Y34 Z-5 R2 F125 M03
N030 X20 Y84
N035 G80 X60 Y16
N040 G82 X60 Y16 Z-25 R18 F125
N045 Y40
N050 Y60
N055 Y84
N060 G80 Z6
N065 G82 X95 Y16 Z-13 R6 F125
N070 Y84
N075 X115
N080 Y84
N085 G80 Z2 M05
```

```
N090 T02 S800 M06                    (Finish drill the ϕ 6 holes)
N095 G81 X60 Y16 Z-43 R18 F125
N100 Y40
N105 Y60
N110 Y84
N115 G80 Z2 M05
N120 T03 S800 M06                    (Pre-drill the tapped holes)
N125 G81 X20 Y34 Z-15 R2 F125 M03
N130 X20 Y84
N135 G80 X95 Y16
N140 G81 X95 Y16 Z-23 R-6 F125
N145 Y84
N150 X115
N155 Y16
N160 G80 Z2 M05
N165 T04 S200 M06                    (Tap all the holes)
N175 G84 X20 Y34 Z-15 R2 F300 M03
N180 X20 Y84
N185 G80 X95 Y16
N190 G84 X95 Y16 Z-23 R-6 F300
N195 Y84
N200 X115
N205 Y16
N210 G80 Z2 M05
N215 T00 M06
N220 G0 Z50
N225 X0 Y0
N230 M02
```

The graphical simulation of the above is shown in Fig. 13.34, simulating the material-removal process on the machine tool.

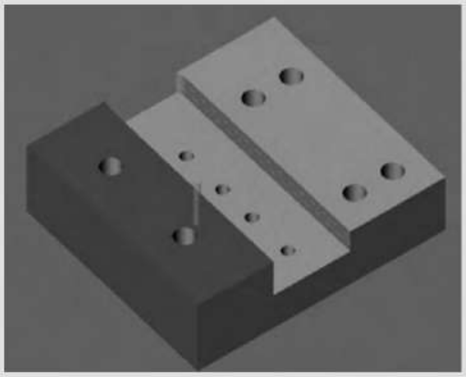


Fig. 13.34 *Simulation of the above part program for for the component shown in Fig. 13.33*

13.8 CUTTER-RADIUS COMPENSATION

In contouring operations, it becomes necessary to calculate the tool path for preparing the program by offsetting the contour by an amount equal to the radius of the cutter. Figure 13.35 shows the part contour and the tool path for a typical component. Apart from the problem of calculating, one should realise that whenever the cutter size changes, the program would need editing. However, if a compensation equal to the radius of the cutter is entered and stored in the control system, then the program could be written for the component profile and thus no change in program would be required. It is as if the program is written with a cutter of zero radius.

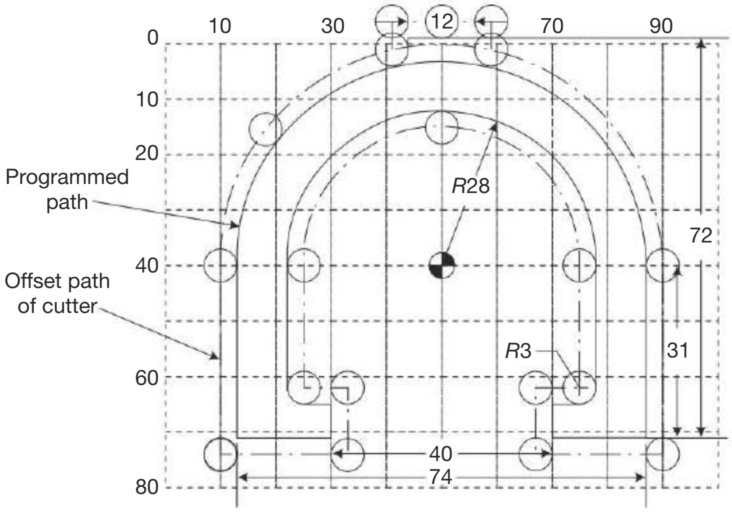


Fig. 13.35 *Cutter radius compensation*

The preparatory functions G40, G41 and G42 are used for radius compensation and form one group. These are modal and the one programmed in any block remains active till cancelled by the other.

G40	Compensation 'off'.
G41	Used when the cutter is on the left of the programmed path when looking in the direction of the tool movement, i.e., the radius compensation is considered to the left of the programmed profile.
G42	Used when the cutter is on the right of the programmed path when looking in the direction of the tool movement, i.e., radius compensation is to the right of the programmed profile (in some systems, cutter diameter compensation is possible and in these cases, the value of the diameter is entered as the compensation value). The tool-radius entry is always positive. If the programmed path is determined for a particular size of the cutter, the compensation value is '+' or '–', depending on whether the cutter used for machining is oversized or undersized.

During the operation with cutter-radius compensation on, the control automatically calculates the tool path as offset in the correct direction by the radius of the tool in the spindle. However, during the start-up and cancelling of the compensation, the actual position is dependent upon the next or previous move.

Consider the cutter moves as shown in Fig. 13.36. The program actually shows the actual coordinates given in the part print. The actual motion of the tool during the machining process is shown with a centreline.

```
N030 G42 G00 X110 Y140      (Startup command A – 1)
N035 G01 X0 F120            (Move 1 – 2)
N040 Y0                     (Move 2 – 3)
N045 X60                    (Move 3 – 4)
N050 X110 Y50               (Move 4 – 5)
N055 Y140                   (Move 5 – 6)
N060 G40 G00 X160 Y170      (Cancel compensation, 6 – A)
```

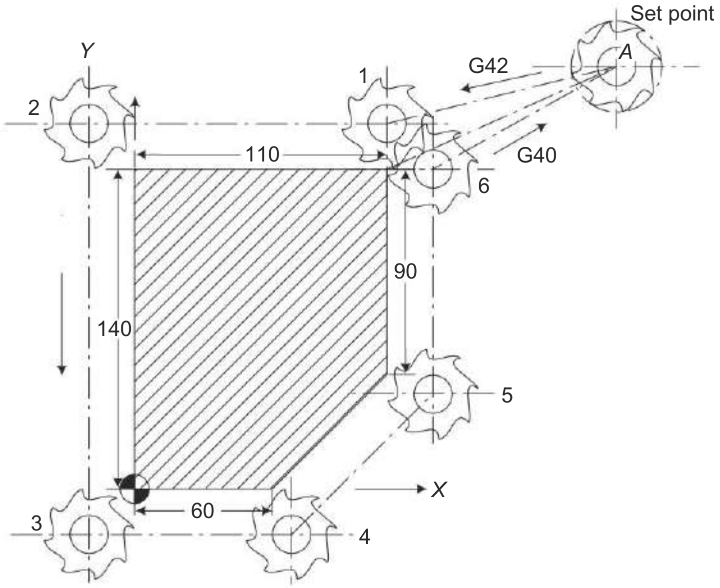


Fig. 13.36 *Example showing the cutter-radius compensation using the G codes G42*

During the start-up move (A – 1), the tool is actually positioned at a point which is normal to the next move, shown in Fig. 13.36 as 1. This move is called the *ramp-on move*. Similarly, when the compensation is cancelled, the tool will be positioned at a point normal to the previous move shown as 6 in Fig. 13.36. This is termed the *ramp-off move*.

Example 13.6 The following program will illustrate the use of these preparatory functions. Figure 13.33 shows the top view and the top of the component surface is taken as $Z = 0$.

```
%
N010 G71 G92 X0 Y0 Z50.0     (Absolute pre-set at A, 50 mm above work surface)
N020 G90                     (Absolute programming)
N030 T01 M06
N040 G00 Z2 M03 S600         (Rapid plane positioning)
N050 G41 G01 X100 Y-60 F100  (cutter compensation ON; tool moves along the path left of the
                             programmed contour)
N055 Z-5.0                   (Move to required depth)
N060 X300.0
N070 Y-260.0
N080 X100.0
N090 Y-60.0
N100 G40 G00 X0 Y0           (cutter radius compensation OFF)
N110 Z50.0 T0 M05            (compensation value for tool T1 cancelled)
N120 M06 T02                 (Second tool for internal profile)
```

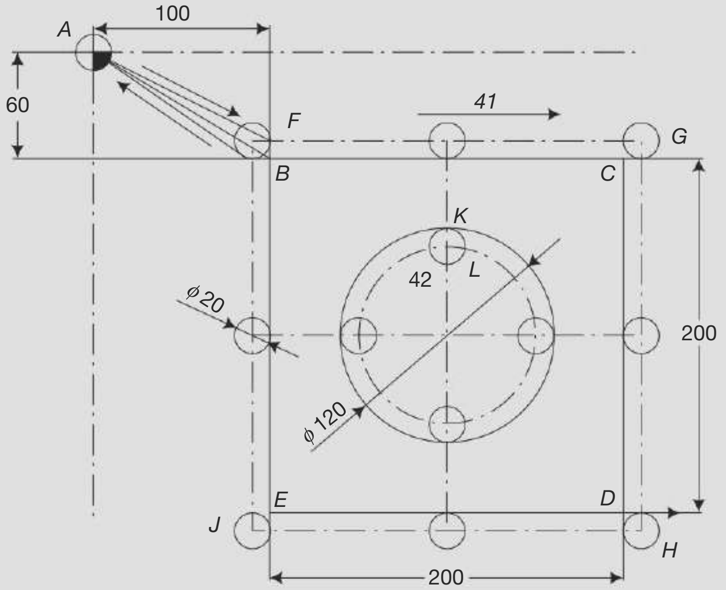


Fig. 13.37 *Example showing the cutter radius compensation using the G codes G41 and G42*

```
N130 X200.0 Y-120.0 Z2.0 S600 M03
N140 G01 Z-5.0 F100
N150 G42 Y-100.0                (cutter compensation ON; tool path right of programmed contour)
N160 G02 I0 J-60.0
N170 G00 Z50.0
N180 G40 X0 Y0 M05
N190 T0 M06
N200 M02
```

In the block N050, the programmed position is *B*, but the cutter centre comes to the position *F* such that *BF* is perpendicular to the next move *BC* (Block N060), programmed in the next block. *AF* is sometimes called 'ramp-on-move' in view of its shape. Similarly, in the block N150, the programmed position is *L* but the cutter centre attains the position *M*, such that *LM* is perpendicular to the direction of the next move, i.e., tangent to the circle (Block N160). In the block N100, the compensation is deactivated by the word G40, which is entered after the programmed position *B*. The cutter stops at *K*, such that *KB* is perpendicular to the direction of the previous move, i.e., *EB*. *KA* is called `ramp-off move'.

In the blocks N110 and N180, T0 causes cancellation of the compensation values for the tool in action at that time. Radius compensation is also used when similar profiles are to be cut with different depths of cut, e.g., rough cut and finish cut. When a rough cut is made, a compensation equal to the thickness of the material to be left for finish is entered and during the finish cut, this compensation is taken off. In view of identical programmed paths in roughing and finishing, such programming is done by using subroutines. For the roughing cut, the subroutine is called in the main program using the compensation and for the finish cut, the same subroutine is called using the same tool without compensation.

Compensation for tool length and radius can be specific to a tool and if so, when the word T03 occurs in a block, it calls the tool number 3 into action with the compensation pre-registered for it. However, in many CNC systems, the compensation values (MDI entry) are stored separately, irrespective of the tools being used. This helps in calling different compensation values even with the same tool when used on different occasions. For example, T0104 word would mean that tool number 01 is used with the compensation value, entered in the register against the identifier 04. Before commencing work on the machine, the operator must examine the compensation values stored and verify them with the list of values supplied to him. Negligence on this count could be very serious.

In some of the popular control systems, the pre-registered compensation values are called in the program block by the words D ... and H... which refer to the tool radius and length compensations respectively. For example,

```
N017 M06 T02
N018 G81 X170.0 Y100.0 Z65.0 R48.0 H07 F100 M03
```

means that the drilling operation takes place with the tool number 02 with a length compensation corresponding to the entry against the identifier 07. Similarly,

```
N074 M06 T06
N075 G01 X70.0 D03 F150 M03
```

means that milling takes place with the tool number 06 with a radius compensation corresponding to the entry in the register against the identifier 03.

Example 13.7 A complete part program using the ISO codes for the following component for the external contour for the component is shown in Fig. 13.38 using the cutter radius compensation.

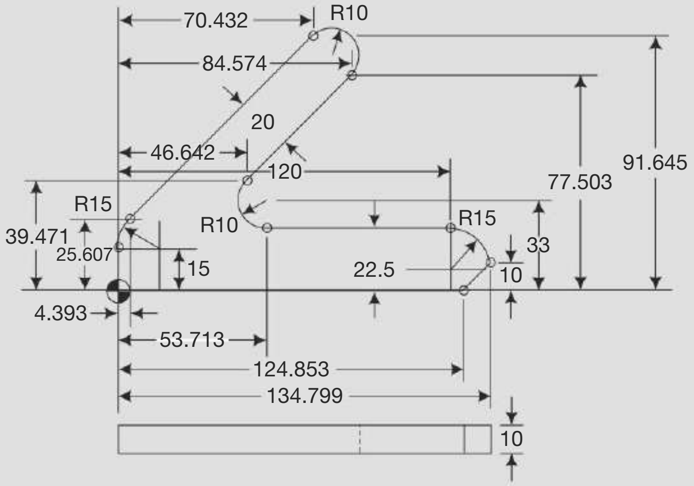


Fig. 13.38 *Example for contour programming using the cutter-radius compensation*

```
%
N010 G71 G92 X-100 Y-100 Z50
N015 G90
```

```
N020 T01 M06
N025 G42 G00 X0 Y0 Z2 D1
N030 G01 Z-10 F120 M03
N035 G01 Y15
N040 G02 X4.393 Y25.607 R15
N045 G01 X70.432 Y91.645
N050 G02 X 84.574 Y77.503 R10
N055 G01 X46.642 Y39.571
N060 G03 X53.713 Y22.5 R10
N065 G1 X120
N070 G02 X134.8 Y10 R15
N075 G01 X124.853 Y0
N080 X0 Y0
N085 Z2
N090 G40 G00 X-100 Y-100 Z50 M05
N095 T00 M06
N100 M02
```

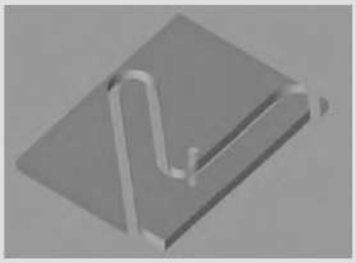


Fig. 13.39 *Simulation of the above part program for the component shown in Fig. 13.38*

The graphical simulation of the above is shown in Fig. 13.39, simulating the material-removal process on the machine tool.

Here, a number of examples for CNC programs from simple to a little more complex are given for practice.

Example 13.8 A complete part program using the ISO codes for the following component for the different holes present in the component is shown in Fig. 13.40. The cutting speed to be used is 50 m/min and the feed rate is 0.08 mm/rev. Use the lower left-hand corner of the part as the datum.

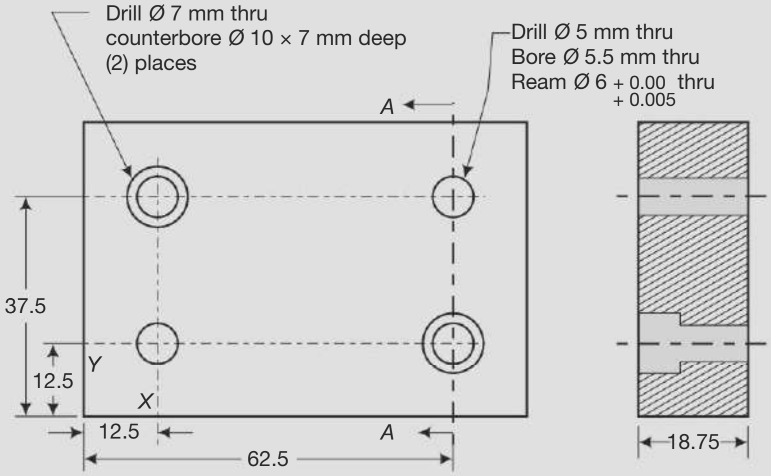


Fig. 13.40 *Part drawing for Example 13.8*

The part requires multiple operations based on the information given in the part drawing. The operations and tools required are the following:

Op. no	*Description*	*Tool required*	*Spindle, RPM*	*Feed, mm/min*
10	Drill two holes	Twist drill 7 mm dia	2200	175
20	Counterbore two holes	End mill 10 mm dia	1590	125
30	Drill two holes	Twist drill 5 mm dia	3180	250
40	Bore the two holes	Boring bar 5.5 mm dia	2890	230
50	Ream two holes	Reamer 6 mm dia	2650	210

Given $V = 50$ m/min and $f = 0.08$ mm/rev

$$\text{Spindle speed, } N = \frac{1000\ V}{\pi D} = \frac{1000 \times 50}{\pi \times 7} = 2273.64 \text{ rpm}$$

Feed to be used = 0.080 × 2200 = 176 mm/min

Similar calculations can be done for all other tools and the values are filled in the above table which are used for writing the part program as shown below:

```
%
01308                                        (P N Rao)
N010 G71 G92 X0 Y0 Z50                       (September 22, 2008)
N015 G90                                     (Stock 75 x 50 x 20 mm)
N020 T01 S2200 M06                           (Drill 7 mm diameter)
N025 G81 X12.5 Y37.5 Z-22.0 R2.0 F175 M03    (Start drilling)
N030 X62.5 Y12.5
N035 G80 X-50.0 Z50.0                        (End of cycle)
N040 T02 S1590 M06                           (End mill 10 mm diameter)
N045 G82 X12.5 Y37.5 Z-7.0 P1.0 R2.0 F125
N050 X62.5 Y12.5
N055 G80 X-50.0 Z50.0                        (End of cycle)
N060 T03 S3180 M06                           (Drill 5 mm diameter)
N065 G81 X12.5 Y12.5 Z-22.0 R2.0 F250
N070 X62.5 Y37.5
N075 G80 X-50.0 Z50.0                        (End of cycle)
N080 T04 S2890 M06                           (Bore 5.5 mm diameter)
N085 G86 X12.5 Y12.5 Z-22.0 R2.0 F230        (Boring cycle)
N090 X62.5 Y37.5
N095 G80 X-50.0 Z50.0                        (End of cycle)
N100 T05 S2650 M06                           (Reamer 6 mm diameter)
N105 G85 X12.5 Y12.5 Z-22.0 R2.0 F210        (Reaming cycle)
N110 X62.5 Y37.5
```

```
N115 G80 X-50.0 Z50.0 M05                    (End of cycle)
N120 G00 X-50.0 Y-50.0
N130 M02                                     (End of program)
```

The graphical simulation of the Example 13.8 is shown in Fig. 13.41, simulating the material-removal process on the machine tool.

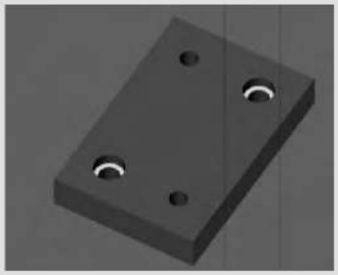


Fig. 13.41 *Simulation of the program for Example 13.8*

Example 13.9 A complete part program using the ISO codes for the following component for the different holes present in the component is shown in Fig. 13.42.

The part requires multiple operations based on the information given in the part drawing. The operations and tools required are given below.

Op. no	*Description*	*Tool required*
10	Drill one hole	Twist drill 5 mm dia
20	Counter bore one hole	End mill 8 mm dia
30	Drill three holes	Twist drill 5 mm dia
40	Tapping the three holes	Machine tap M6

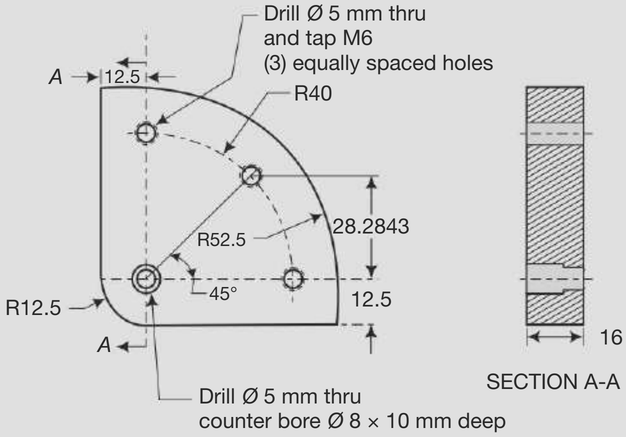


Fig. 13.42 *Part drawing for Example 13.9*

Choose the datum as the centre of the bottom left-hand hole.

```
%
O1309                                (P N Rao)
N010 G71 G92 X0 Y0 Z50               (September 22, 2008)
N015 G90                             (Stock 65 x 65 x 16 mm)
N020 T01 S800 M06                    (Drill 7 mm diameter)
```

```
N025 M03
N030 G81 X0 Y0 Z-20.0 R2.0 F80            (Start drilling)
N035 G80 X-50.0 Z50.0                     (End of cycle)
N040 T02 S800 M06                         (End mill 8 mm diameter)
N045 G82 X0 Y0 Z-10 P1 R2 F80
N050 G80 X-50.0 Z50.0                     (End of cycle)
N055 T03 S800 M06                         (Drill 5 mm diameter)
N060 G81 X40 Y0 Z-20.0 R2.0 F80
N065 X28.284 Y28.284
N070 X0 Y40.0
N075 G80 X-50.0 Z50.0                     (End of cycle)
N080 T04 S200 M06                         (M6 Machine tap)
N085 G84 X40 Y0 Z-16.0 R2.0 F200
N090 X28.284 Y28.284
N095 X0 Y40.0
N100 G80 X-50.0 Z50.0 M05                 (End of cycle)
N110 G00 X-50.0 Y-50.0
N120 M02                                  (End of program)
```

The graphical simulation of the Example 13.9 is shown in Fig. 13.43, simulating the material-removal process on the machine tool.

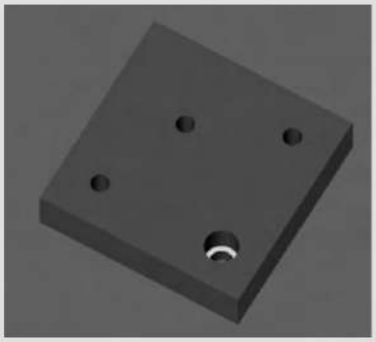


Fig. 13.43 *Simulation of the program for Example 13.9*

Example 13.10 A complete part program using the ISO codes for the following component for the different holes present in the component shown in Fig. 13.44.

The part requires multiple operations based on the information given in the part drawing. The operations and tools required are given below

Op. no	*Description*	*Tool required*
10	Drill four holes	Twist drill 7 mm dia
20	Counter bore four holes	End mill 10 mm dia
30	Drill four holes	Twist drill 5 mm dia
40	Tapping the four holes	Machine tap M6

Choose the datum as the centre of the circle in view of the symmetry of the geometry. The coordinates can then be easily calculated since they are all symmetrical about the datum.

```
%
O1310                                          (P N Rao)
N010 G71 G92 X0 Y0 Z50                         (September 22, 2008)
N015 G90 M03                                   (Stock 65 x 65 x 12.5 mm)
N020 T01 S800 M06                              (Drill 7 mm diameter)
```

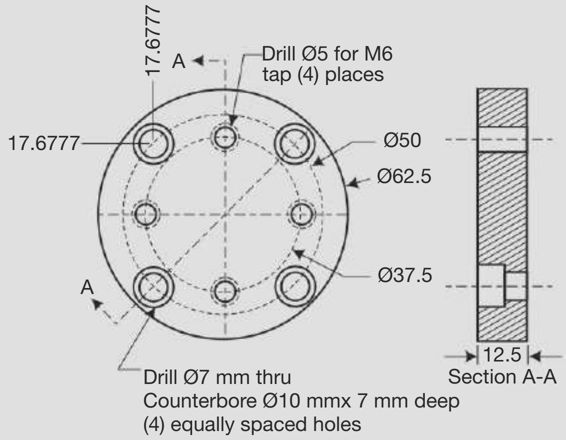


Fig. 13.44 *Part drawing for Example 13.10*

```
N025 G81 X17.678 Y17.678 Z-16.0 R2.0 F80     (Start drilling)
N030 X-17.678 Y17.678
N035 X-17.678 Y-17.678
N040 X17.678 Y-17.678
N045 G80 X-50.0 Z50.0                        (End of cycle)
N050 T02 S800 M06                            (End mill 10 mm diameter)
N055 G82 X17.678 Y17.678 Z-7 P1 R2 F80
N060 X-17.678 Y17.678
N065 X-17.678 Y-17.678
N070 X17.678 Y-17.678
N075 G80 X-50.0 Z50.0                        (End of cycle)
N080 T03 S800 M06                            (Drill 5 mm diameter)
N085 G81 X0 Y18.75 Z-16.0 R2.0 F80
N090 X0 Y18.75
N095 X-18.75 Y0
N100 X0 Y-18.75
N105 X18.75 Y0
N110 G80 X-50.0 Z50.0                        (End of cycle)
N115 T04 S300 M06                            (M6 Machine tap)
N120 G84 X0 Y18.75 Z-16.0 R2.0 F80
N125 X0 Y18.75
N130 X-18.75 Y0
N135 X0 Y-18.75
N140 X18.75 Y0
```

```
N145 G80 X-50.0 Z50.0 M05                    (End of cycle)
N150 G00 X-50.0 Y-50.0
N160 M02                                     (End of program)
```

The graphical simulation of Example 13.10 is shown in Fig. 13.45, simulating the material-removal process on the machine tool.

Example 13.11 A complete part program using the ISO codes for the following component for the external contour for the component is shown in Fig. 13.46. The cutting speed to be used is 30 m/min and the feed rate is 0.08 mm/rev. Use the lower left-hand corner of the part as the datum.

Taking datum at the left-hand bottom part of the part, all the dimensions where the cutter has to move in order to generate the part are shown in Fig. 13.47. Based on these dimensions, the part program is developed with three tools, the details of which are given in the program.

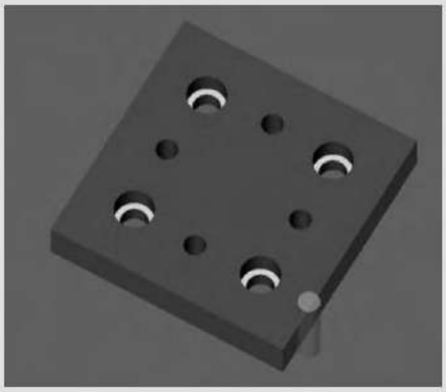
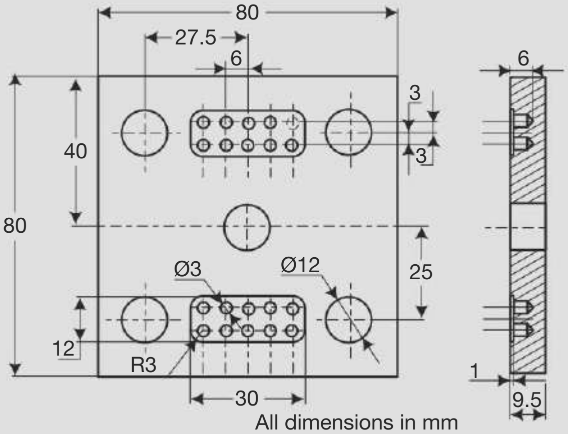


Fig. 13.45 *Simulation of the program for Example 13.10*



All dimensions in mm
AISI 1045 steel

Fig. 13.46 *Part drawing for Example 13.11*

Given $V = 30$ m/min and $f = 0.08$ mm/rev

$$\text{Spindle speed, } N = \frac{1000\,V}{\pi D} = \frac{1000 \times 30}{\pi \times 12} = 1591 \text{ rpm}$$

Feed to be used = 0.080 × 1590 = 127 mm/min

Similar calculations can be done for all other tools and the values are filled in the tool table which are used for writing the part program as shown below:

Op. no	*Description*	*Tool required*	*Spindle, RPM*	*Feed, mm/min*
10	Mill the pocket	End mill 6 mm dia	1590	125
20	Drill five holes	Twist drill 12 mm dia	800	65
30	Drill twenty holes	Twist drill 3 mm dia	3180	250

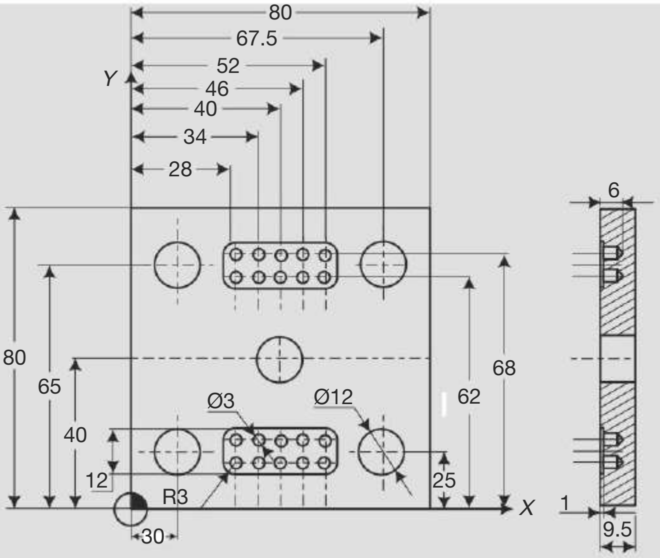


Fig. 13.47 *Part drawing shown with cutter for Example 13.11*

```
N10 (P N Rao 22 September 2008)
N20 (USE 80 x 80 x 10 mm AISI 1045 steel)
N30 (INITIALIZE STOCK TO X=0, Y=0, Z=0)
N40 (T1 IS A 6 mm END MILL)
N50 (T2 IS A 12 mm twist drill)
N60 (T3 IS A 3 mm twist drill)
N70 G71 G90
N80 M06 T01 (Tool Change 6 mm End Mill)
N90 (Pocket milling Operation, Contour 1)
N100 M3 S1590.0
N110 G0 Z10.0 (Retract)
N120 G0 X28.0 Y12.0
N130 G0 Z2.0 (Retract)
N140 G1 Z-1.0 F125.0 (Plunge)
N150 G1 X52.0 F125.0
N160 Y18.0
N170 X28.0
N180 Y12.0
N190 G0 Z2.0 (Retract)
N200 G0 Y62.0 (Position for the second contour)
N210 G1 Z-1.0 F100.0 (Plunge)
N220 G1 X52.0 F200.0
```

```
N230 Y68.0
N240 X28.0
N250 Y62.0
N260 G0 Z2.0 M05 (Retract)
N270 M06 T02 (Tool change 12 mm twist drill)
N280 (Drill 5 holes)
N290 S800.00 M03
N300 G0 X12.5 Y15.0
N310 G81 G99 Z-14.0 R1.0 F65.0
N320 X67.5 Y15.0
N330 X40.0 Y 40.0
N340 X67.5 Y65.0
N350 X12.5 Y 65.0
N360 G80
N370 G0 Z10.0 (Retract)
N380 M06 T03 (Tool change 3 mm twist drill)
N390 (Drill 20 holes)
N400 S3180.00
N410 G0 X28.0 Y68.0
N420 G81 G99 Z-6.0 R1.0 F250.0
N430 X34.0 Y68.0
N440 X40.0 Y68.0
N450 X46.0 Y68.0
N460 X52.0 Y68.0
N470 X52.0 Y62.0
N480 X46.0 Y62.0
N490 X40.0 Y62.0
N500 X34.0 Y62.0
N510 X28.0 Y62.0
N520 X28.0 Y18.0
N530 X34.0 Y18.0
N540 X40.0 Y18.0
N550 X46.0 Y18.0
N560 X52.0 Y18.0
N570 X52.0 Y12.0
N580 X46.0 Y12.0
N590 X40.0 Y12.0
N600 X34.0 Y12.0
N610 X28.0 Y12.0
```

```
N620 G80 M05
N630 G0 Z10.0 (Retract)
N640 M02 (END OF FILE)
```

The graphical simulation of Example 13.11 is shown in Fig. 13.48, simulating the material-removal process on the machine tool.

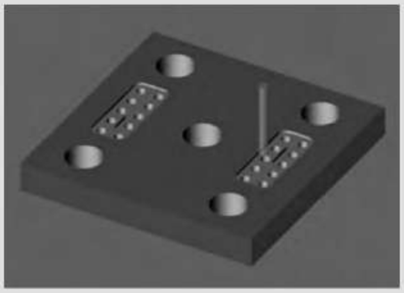


Fig. 13.48 *Simulation of the program for Example 13.11*

Example 13.12 A complete part program using the ISO codes for the following component to completely machine the component shown in Fig. 13.49.

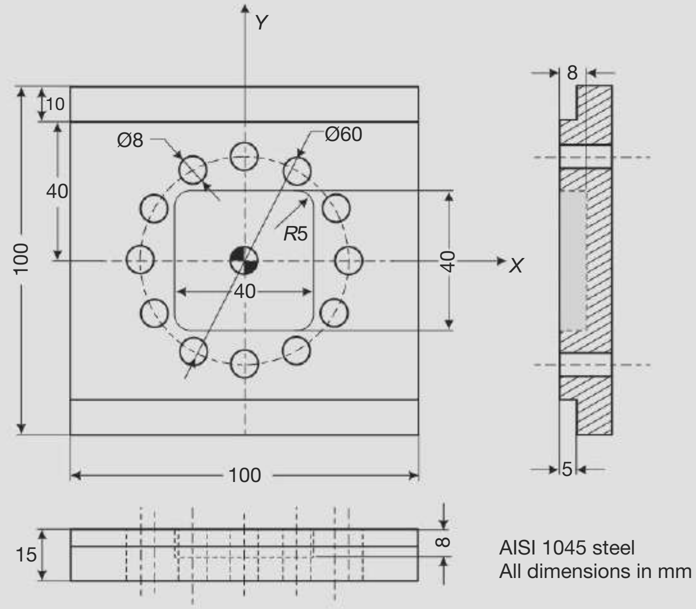


Fig. 13.49 *Part drawing for Example 13.12*

```
N10 (Class exercise Sept 25, 2008 - P N Rao)
N20 (Stock: 100 x 100 x 15 mm - SAE 1045 steel)
N30 (Tool 1 - End mill 10 mm dia)
N40 (Tool 2 - Drill 8 mm dia)
N50
N60 G71 G90 (Metric, Absolute)
N70 T1 M06 (End mill)
N80 S500 M03
N90 G00 X-55.0 Y-50.0 Z2.0 (Position the tool above the start point of cut)
N100 G01 Z-5.0 F75.0 (Mill to depth)
```

```
N110 X55.0
N120 Y-45.0
N130 X-55.2
N140 G0 Y50.0
N150 G1 X55.0 F75.0
N160 Y45.0
N170 X-55.0
N180 G0 Z2.0
N190 X-15.0 Y-15.0 (Position above the pocket)
N200 G1 Z-8.0 F75.0 (Mill to depth)
N210 X15.0 (Mill pocket)
N220 Y-7.0
N230 X-15.0
N240 Y1.0
N250 X15.0
N260 Y9.0
N270 X-15.0
N280 Y15.0
N290 X15.0
N300 Y-15.0
N310 X-15.0
N320 Y15.0
N330 X15.0
N340 G0 Z2.0 (Lift the tool)
N350 T2 M06 (Drill)
N360 S850 M03
N370 G81 X30.0 Y0 Z-19.0 R2.0 F110 (Drill the first hole)
N380 X25.98 Y15
N390 X15.0 Y25.98
N400 X0 Y30.0
N410 X-15.0 Y25.98
N420 X-25.98 Y15.0
N430 X-30.0 Y0
N440 X-25.98 Y-15.0
N450 X-15.0 Y-25.98
N460 X0 Y-30.0
N470 X15.0 Y-25.98
N480 X25.98 Y-15.0
```

```
N490 G80 Z25.0 M05 (Cancel fixed cycle, stop the
spindle)
N500 G00 X-75.0 Y-75.0 Z50.0
N510 M02
```

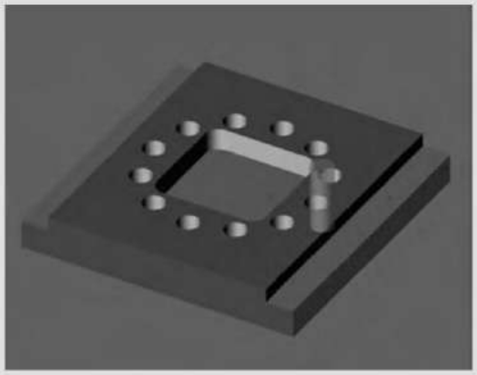


Fig. 13.50 *Simulation of the program for Example 13.12*

Example 13.13 A complete part program using the ISO codes for the following component for the external contour for the component is shown in Fig. 13.51 without using the cutter radius compensation.

Since the arc radius is given as 7.5 mm, if an end-mill size of 15 mm diameter is used then it is possible to reduce a few cuts, since that radius will be formed by the cutter size itself. The resulting tool path is shown in Fig 13.52, with the cutter positions being sequentially given. Also, the cutter location data as ordinates is given in the above drawing taking the bottom left-hand corner as the datum. It is now possible for the user to write a part program with these coordinate values.

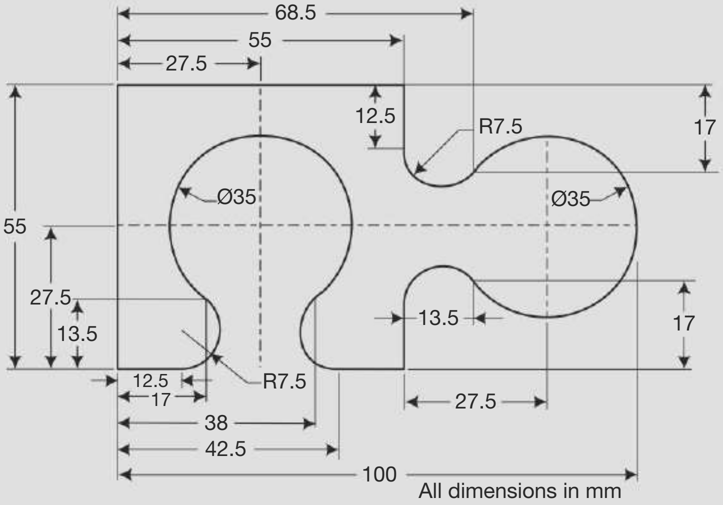


Fig. 13.51 *Part drawing for Example 13.13*

```
%
O1313                          (P N Rao, September 22, 2008)
N010 G71 G92 X0 Y0 Z50         (Stock 105 x 60 x 5 mm)
N015 G90                       (End mill 15 mm diameter)
N020 T01 S800 M06
N025 G00 X-7.5 Y-7.5 Z2.0 M03  (Position above the start point 1)
N030 G01 Z-6.0 F100.0          (Mill to depth)
N035 X-7.5 Y62.5               (Point 2)
```

```
N030 X62.5 Y62.5                (Point 3)
N035 X62.5 Y42.5                (Point 4)
N040 G02 X62.5 Y12.5 R-25.0     (Point 5)
N045 G01 X62.5 Y-7.5            (Point 6)
N050 X42.5 Y-7.5                (Point 7)
N055 G02 X33.5 Y19.5 R15.0      (Point 8)
N060 G03 X21.5 Y19.5 R10.0      (Point 9)
N065 G02 X12.5 Y-7.5 R15.0      (Point 10)
N070 G01 X-7.5 Y-7.5            (Last Point 1)
N075 G00 Z2.0 M05               (Lift tool to clearance plane)
N080 X-50.0 Y-50.0 Z50.0        (Position tool away from workpiece)
N090 M02                        (End of program)
```

The graphical simulation of Example 13.13 is shown in Fig. 13.53, simulating the material-removal process on the machine tool.

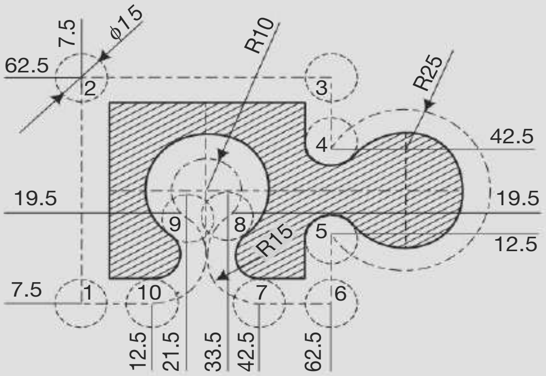


Fig. 13.52 *Part drawing shown with cutter for Example 13.13*

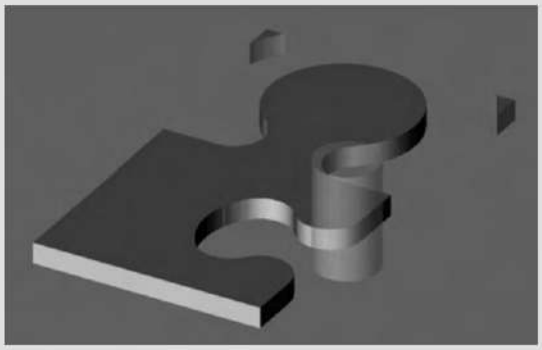


Fig. 13.53 *Simulation of the program for Example 13.13*

Example 13.14 A complete part program using the ISO codes for the following component for the external contour for the component shown in Fig. 13.51 without using the cutter radius compensation.

Since the part in Fig. 13.54 has a number of transition points between the arc segments, calculating the points through which the cutter centre should pass is going to be tedius if geometric calculations are used. Hence, it may be a better idea to use CAD to draw the geometry making sure that the part datum is kept exactly at the origin of the coordinate system as shown in Fig. 13.55. Select an end mill of large size (25 mm in this case) and offset the contour to be machined by an amount equal to the radius of the selected cutter (12.5 mm). Then using the ordinate facility in the CAD system, get the coordinates of all the transition points on the offset contour. From that prepare a table of coordinates through which the cutter should pass to get the required contour.

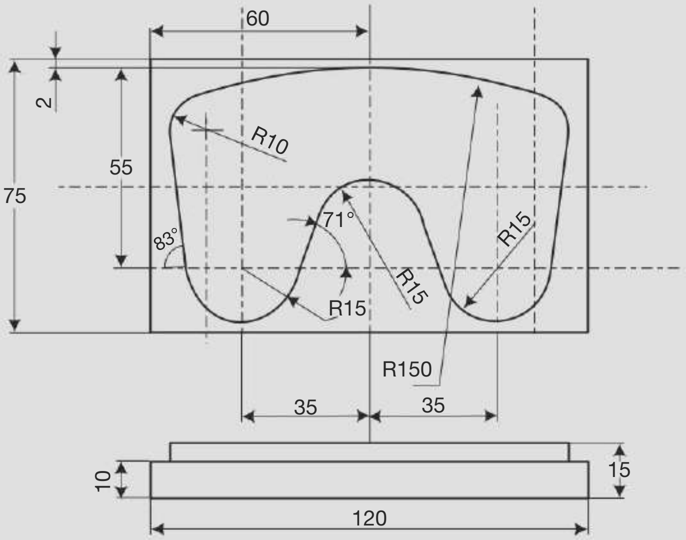


Fig. 13.54 *Part drawing for Example 13.14*

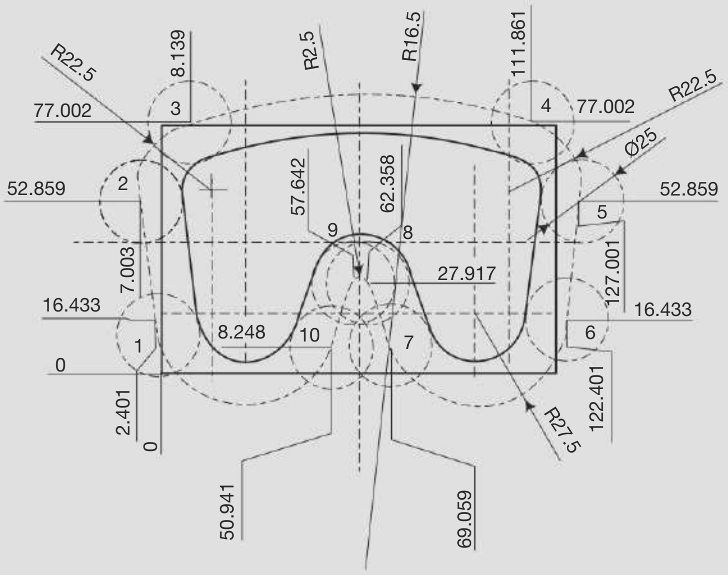


Fig. 13.55 *Part drawing shown with cutter for Example 13.14*

```
%
O1314                                (P N Rao, September 22, 2008)
N010 G71 G92 X0 Y0 Z50               (Stock 120 x 80 x 15 mm)
N015 G90                             (End mill 25 mm diameter)
```

Point no.	*X-coordinate*	*Y-coordinate*	*Radius*	*Arc direction*
1	–2.401	16.433	—	—
2	–7.003	52.859	22.5	CW
3	8.139	77.002	162.5	CW
4	111.861	77.002	22.5	CW
5	127.003	52.859	—	—
6	122.401	16.433	27.5	CW
7	69.059	8.248	—	—
8	62.358	27.917	2.5	CW
9	57.642	27.917	—	—
10	50.941	8.248	27.5	CW

```
N020 T01 S800 M06
N025 G00 X-2.401 Y16.433 Z2.0            (Position above the start point 1)
M03                                      (Mill to depth)
N030 G01 Z-5.0 F100.0                    (Point 2)
N035 X-7.003 Y52.859                     (Point 3)
N030 G02 X8.139 Y77.002 R22.5            (Point 4)
N035 X111.861 Y77.002 R162.5             (Point 5)
N040 X127.003 Y52.859 R22.5              (Point 6)
N045 G01 X122.401 Y16.433                (Point 7)
N050 G02 X69.059 Y8.248 R27.5            (Point 8)
N055 G01 X62.358 Y27.917                 (Point 9)
N060 G03 X57.642 Y27.917 R2.5            (Point 10)
N065 G01 X50.941 Y8.248                  (Last Point 1)
N070 G02 X-2.401 Y16.433 R27.5
N075 G00 Z2.0 M05                        (Lift tool to clearance plane)
N080 X-50.0 Y-50.0 Z50.0                 (Position tool away)
N090 M02                                 (End of program)
```

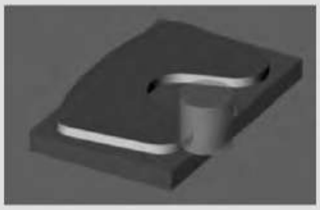


Fig. 13.56 *Simulation of the program for Example 13.14*

The graphical simulation of Example 13.13 is shown in Fig. 13.56, simulating the material-removal process on the machine tool.

Summary

- Part programming for CNC machine tools is an activity that requires knowledge of the machine-tool characteristics as well as the machining practices. A lot of previous knowledge and experience in these areas will improve the part programming ability.
- Part programming is a systematic activity and following the steps in proper sequence helps in developing efficient part programs.
- Selection of the part coordinate axes system with respect to the part geometry helps in reducing the mathematical calculations that are needed to write the part program.

- The ISO standard for part programming utilises all the 26 characters of the English alphabet as word addresses in a part program.
- Preparatory functions or G-codes control the geometrical nature of the data input as well as operation of some of the data in a program. A large number of these codes are standardised by ISO.
- Motion commands are the most important part of the G-codes used to control the motion in rapid (G00), linear interpolation (G01) and circular interpolation (G02 and G03).
- Miscellaneous functions, or M-codes, are used to specify the machine-tool functions in the program. Less number of codes are standardised in view of these being machine specific.
- For programs involving multiple tools, tool-length compensation can be activated by the use of tool offset registers.
- Canned, or fixed, cycles offer the facility to reduce the bulk of the program by canning the often repeated command into a single G-code. Examples are for the drilling of holes (G81).
- In milling of contours, large amount of complex calculations are often required to take multiple cuts. This can be reduced by the use of cutter diameter compensation (G41 and G42).

Questions

1. What is the know-how that a computer numerical control programmer should have for developing the programs?
2. What are the most important functions that are used for programming?
3. Explain the function of the preparatory functions. Give the functioning of any one G code used for the purpose.
4. What is dwell? Explain its function and how it is specified in a part program.
5. Explain the method used for specifying the tool specification in a CNC part program.
6. Explain the concept of ‘floating datum’ and ‘set point’ with reference to CNC part programming. What is their relationship? Explain how they are used in programming in ISO format.
7. How is the tool-length compensation specified in a machining centre?
8. Explain with a neat sketch the operation of the canned cycle G81 as per ISO.
9. What are the various functions embedded in the G82 canned cycle? Explain their use with examples in the ISO format.
10. What do you understand by the term ‘canned cycle’ in manual part programming? Explain with neat sketches the differences between the operation of the canned cycles G81 and G83.
11. How is cutter compensation given in the case of a machining centre? Explain with the help of an example how it is operational. Specify any of the limitations in using this facility.
12. What are the limitations of cutter-radius compensation?

Problems

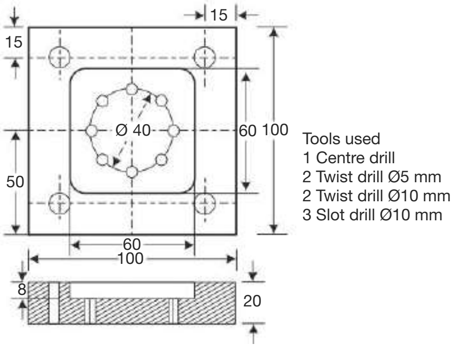
1. For the following component (Fig. 13.57), make a part program on a machining centre equipped with an ISO standard controller. The work material is AISI 1040 steel. Clearly show the set point and axes on the sketch of the part. Also, prepare the planning sheet as used in the laboratory. Show all the calculations that are necessary.

2. Examine the following CNC part program for a machining centre equipped with a controller following the ISO standard. Identify any errors found in the program and also explain the errors.

```
%
N7001 *
N1 G71 *
N2 G90 *
N3 T1 M6 *
N4 G0 X75.0 Y100.0 *
N5 G1 Z-3 *
N6 X175.0 F100 *
N7 Y25.0 *
N8 X75.0 *
N9 Y100.0 *
N10 M30*
```

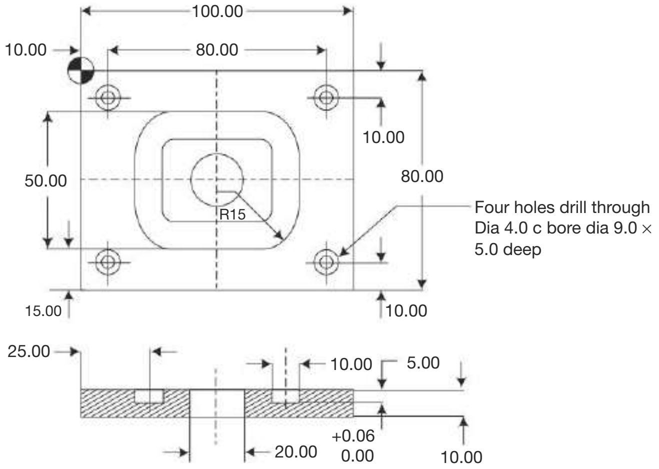
For the above program, prepare the geometry of the part generated, if the diameter of the slot drill used is 10.0 mm.

3. For the program shown in Question 2 above, prepare the geometry of the part generated, if the diameter of the slot drill used is 5 mm.
4. For the following component (Fig. 13.58), make a part program on a vertical-axis machining centre. Clearly show the set point and axes on the sketch of the part. Prepare also the planning sheet as used in the laboratory.



Tools used
1 Centre drill
2 Twist drill Ø5 mm
2 Twist drill Ø10 mm
3 Slot drill Ø10 mm

Fig. 13.57



Four holes drill through Dia 4.0 c bore dia 9.0 × 5.0 deep

Fig. 13.58

5. Examine the following CNC part programs. Give the errors in the programs and also explain the errors.

(a)
```
%
O7001 *
N2 G0 X3.0 Y4.0 *
N3 G1 X7.0 F100 *
N4 Y1.0 *
N5 X3.0 *
N6 Y4.0 *
N10 M02 *
```

(b)
```
%
O9401*
N1 G17 T1 M6 *
N2 G92 X90.0 Y70.0 *
N3 G81 Y2.0 Z-10.0 F200 S500 M3*
N5 G01 X4.0 Y12.0 F150 *
N10 M02 *
```

6. Explain the mistakes found in the following statements.

(a) N05 G01 X12.3 Y23.0 F120 *
(b) N25 G04 X2.0 O1234 *
(c) N45 G00 T1001 S400 *
(d) N75 G03 X0 Z0 F120 *
(e) N60 M01 T1000 Y0 *

7. For the component shown in Fig. 13.59, make a part program on a machining centre equipped with the ISO controller. Clearly show the set point and axes on the sketch of the part. Show all the necessary calculations.

8. For the components shown in Fig. 13.60 to 13.62, make part programs for machining on a CNC machining centre equipped with the ISO controller. Clearly show the set point and axes on the sketch of the part. Show all the necessary calculations.

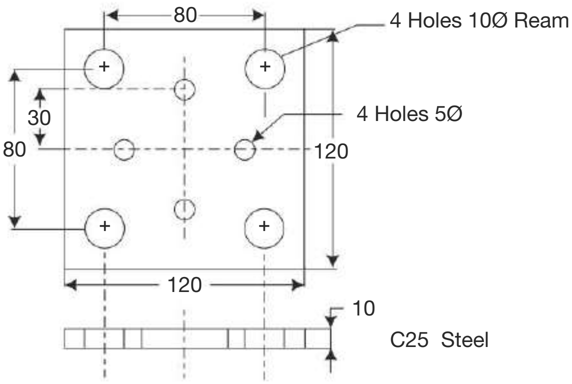


Fig. 13.59

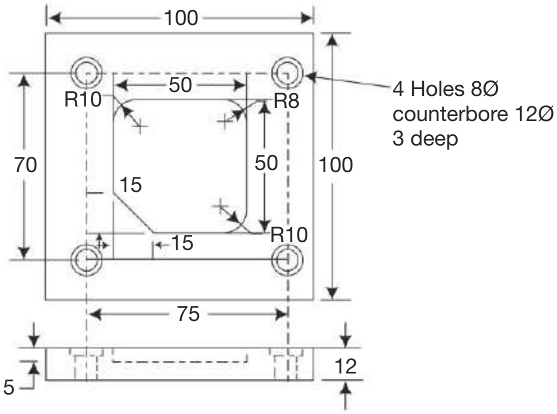


Fig. 13.60

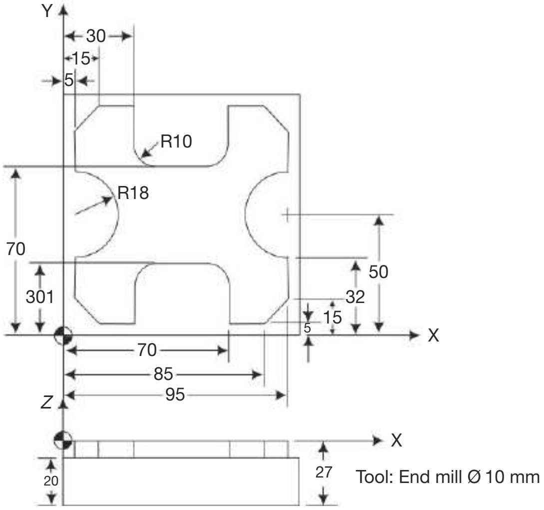


Fig. 13.61

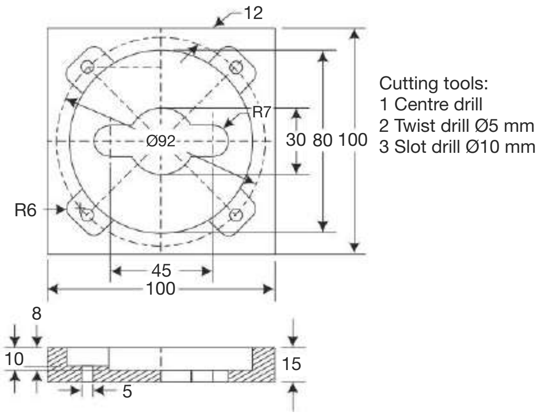


Fig. 13.62

9. For the components shown in Fig. 13.63 to 13.65 make part programs for machining on a CNC turret drill press equipped with the ISO controller using absolute programming. Choose the datum as the bottom left-hand corner of the part. Clearly show the set point and axes on the sketch of the part. The cutting speed to be used is 50 m/min and the feed rate is 0.08 mm/rev. Show all the necessary calculations.

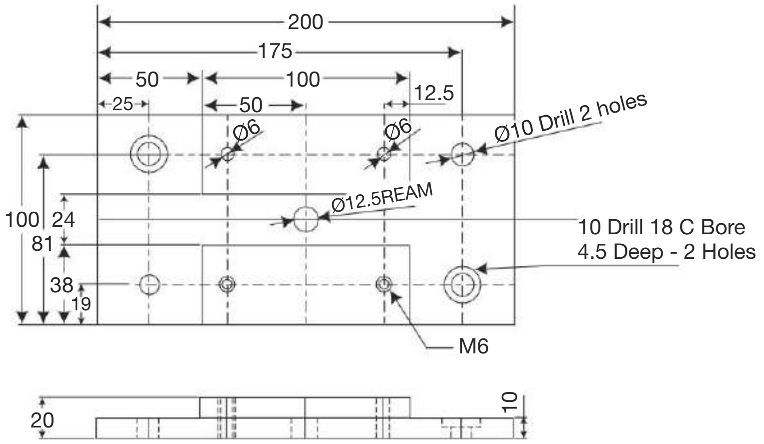


Fig. 13.63

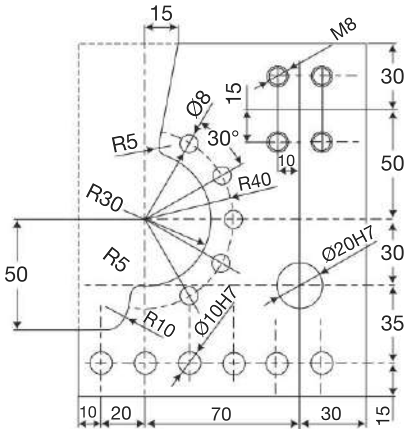


Fig. 13.64

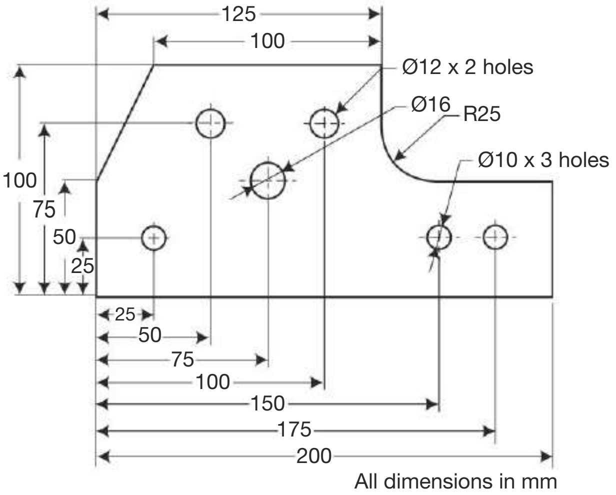


Fig. 13.65

10. For the components shown in Fig. 13.66 and 13.67, make part programs for machining on a CNC machining centre equipped with the ISO controller using absolute programming. Choose the datum as the bottom left-hand corner of the part. Clearly show the set point and axes on the sketch of the part. The cutting speed to be used is 50 m/min and the feed rate is 0.08 mm/rev. Show all the necessary calculations.

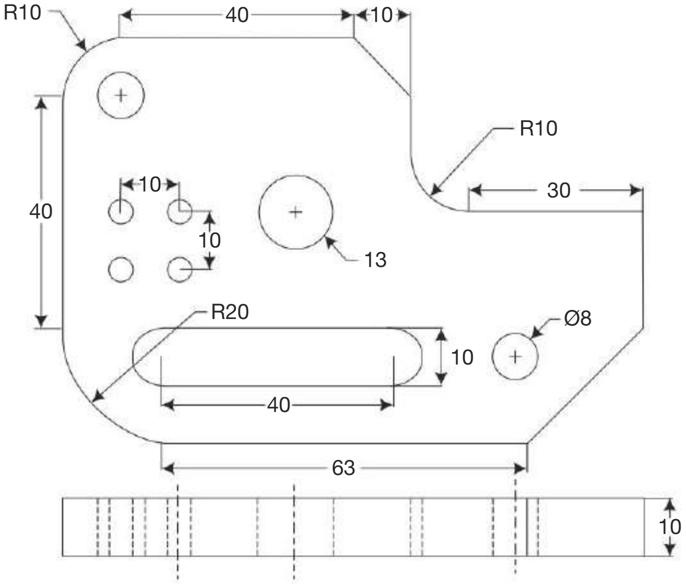


Fig. 13.66

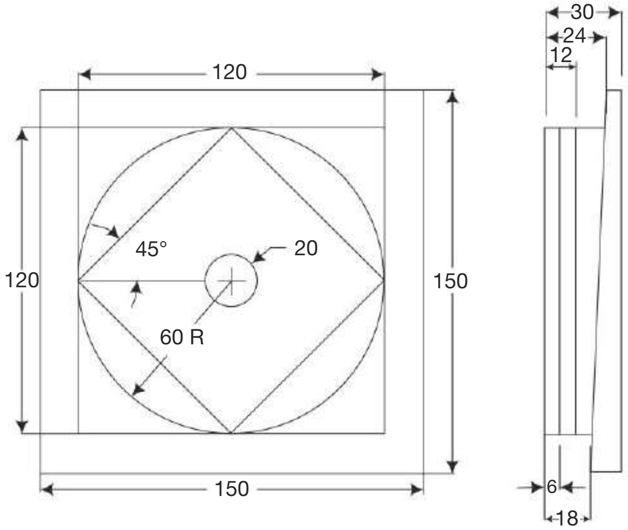


Fig. 13.67

14

TURNING-CENTRE PROGRAMMING

Objectives

Turning-centre programming, though similar to the machining centre programming, has certain differences and they need to be taken into consideration while programming. After completing the study of this chapter, the reader should be able to

- Comprehend the major differences between turning centres and machining centres that need to be taken care of during the part-program preparation
- Understand the tape formats and word addresses as used in turning-centre part programs
- Identify the different axes systems in turning centres
- Learn the basic formats of word addresses as used in part programs for turning centres
- Understand different usage of the G-codes and M-codes for use in part programs
- Use the tool-nose radius compensation effectively to reduce the machining errors
- Plan and develop part programs with complex contours
- Appreciate the various methods used to cut threads using turning centres
- Understand the use of special canned cycles used in turning centres

14.1 COMPARISON BETWEEN MACHINING CENTRES AND TURNING CENTRES

In the previous chapter, the method of programming for machining centres was discussed. The method to be followed for turning centres is also is similar, while taking into account the following differences from machining centres.

Axes System The axes system followed is basically X and Z with additional axes such as C, W and Y coming occasionally based on the structure of the machine tool (turning centre) as seen in Chapter 12.

Tools Used Generally, single-point cutting tools such as turning and boring tools are used. However, when the machine is equipped with C axis and power (or driven) tools in the tool turret, all the types of milling tools would also be used. These may be used on the turnmill centres, where the small milling operations such as keyway milling, flat milling, PCD drilling, etc., can be carried out.

Multiple Cuts Very often the blanks for turned jobs are normally solid materials (rolled stock) which require the removal of large amount of material in roughing cuts for clearing. This is in contrast to machining centres, where only a few cuts (often one or two) along the final contour to be generated are required. As a result, it is necessary to plan the optimum tool path for clearance in a number of turning jobs. This is called cut planning.

Internal and External Features There are identical features in these jobs with internal and external equivalents, such as turning and boring, etc. Programming for them is similar, however, care has to be taken to see that the tool while positioning for cutting internal features be carefully handled.

More than one Tool Post In the earlier machines such as Herbert Batchmatic, there used to be two tool posts. While programming, care has to be taken to see from where the tool has been brought into contact with the workpiece. However, the present-day turning centres mostly are with single tool turrets and therefore less problems are encountered.

Different G and M Codes The preparatory and miscellaneous codes used are to some extent same but some are different in view of the basic structure of the machine tool.

Tool-nose Radius In a machining centre, the programming is done with the centre of the spindle and its top end whereas in turning centres, the programming is to be done with the centre of the tool nose radius and compensation for the tool nose radius has to be done while programming.

Work Rotation The greatest simplification to programming of turning centres is because of the work rotation and as a result, the 3D profile gets transformed to a 2D profile. However, in some cases other axes may be involved, but simultaneously there can be only two axes.

Tool Indexing Normally, tool change in a turning centre is done by means of turret indexing, though some turning centres have tool magazines in the traditional sense. As a result, the tool-change position could be very close to the workpiece surface with enough clearance for the longest tool present in the turret.

Diameter Programming Very often the dimensioning practice in turned components is to give all diameters. However, the tool should approach for machining only radially and hence the provision of diameter programming facility in most of the turning centres. This means that all the movements along X axis should be doubled to represent the diametral movement rather than the radial movement.

14.2 TAPE FORMATS

A typical tape format for a turning centre GE FANUC 18T (metric) is presented here which is generally similar in the other machines as well.

N04	Block number, total four digits with leading zero suppression.
G03	Preparatory function, up to four preparatory functions in a block, however only one from a particular block (discussed later).

X(*U*) ± 053	Word address for *X* axis with absolute values specified by *X* and incremental values specified by *U*. Because of the separate word addresses used for absolute and incremental dimensions, it is possible to mix absolute and incremental dimensions in a single block without the use of G90 or G91 preparatory function. Again this facility is provided because traditionally in the turned components, diameters (*X*) is shown in absolute, whereas the lengths along the axis (*Z*) are shown in incremental form, i.e., steps.
Z(*W*) ± 0.53	Same as above for *Z* axis.

For axis dimensions, both decimal point and leading zero suppression are permitted. The controller automatically identifies a dimension as leading zero suppression if no explicit decimal point is present. Care has to be exercised while giving the axes coordinates.

R 053	Radius value for circular interpolation
I 053	-Centre coordinates in incremental format from the start point of
K ±053	-an arc for circular interpolation
F 032	Feed specification in mm/rev
F 050	Feed specification in mm/min
S 04	Spindle speed specification in direct rpm or constant surface speed in m/min
T 04	Tool function
M 02	Miscellaneous function, only one per block.

Besides these word addresses, certain other word addresses are also used while programming such as *A*, *B*, *C*, *E*, *P*, *Q*, *O*, etc. and they are discussed when actually encountered.

The End Of Block (EOB) character is a carriage return character in EIA (RS-224-B) format and a line feed character in ASCII (ISO) (RS-358-B) format. When programming from the keyboard, use the EOB key. This character will be displayed as a semicolon (;) on the control display screen. An End Of Record (EOR) character should be the first and last character in a program, which is to be uploaded to the machine control through the RS-232 serial port. If multiple programs are to be loaded from a single punched tape, punch the EOR character after the last program on the tape. The EOR character will be followed by an EOB character. The EOB character must be used after the last character in each data block of a part program that is to be loaded into the memory of the control. If the EOB character is omitted from a part program data block, the control will consider the next block to be a part of the block missing the EOB. This may cause undesirable machine behaviour.

Operator messages and comments can be included in a part program by enclosing them in parentheses. Any legal ASCII character can be used when writing a comment.

The Block Skip (/) code inserted at the beginning of a data block will cause that block of data to be ignored by the control when block skip is activated by the machine operator. When block skip is not active, the data block will be executed.

The words within a block in a word address format may follow any convenient sequence. However, it is generally recommended to follow the following sequence.

/, N, G, X, Z, U, W, B, C, I, K, P, Q, R, A, F, S, T, M

14.3 AXES SYSTEM

The general arrangement of a turning centre is shown in Fig. 14.1. The spindle axis is designated as Z and the radial axis perpendicular to the Z and away towards the principal tool post is termed as X axis. Here,

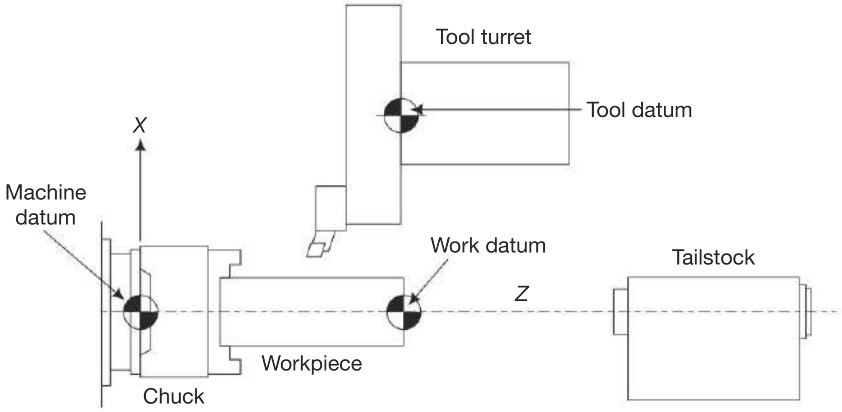


Fig. 14.1 *Axes system used for turning centres*

three datums have been defined. The first datum shown is a home or machine datum which is a prefixed position on the machine. This may generally be the intersection of spindle axis and clamping plane (collet face). At the start (power on) the controller display should show the axis positions with respect to home. A component datum is a datum position fixed by the programmer on the component for the convenience of part programming. At the beginning of the program, the position of the component datum with respect to home position would have to be specified (G50 or G92). Tool datum is the third datum, which is of interest in turning centres, which is essentially the point about which the turret indexes as well as the axes move.

Though programming is normally expected to be done by moving the tool tip around the component profile, in actual practice, it is the tool datum, which actually moves along the axes. The difference between the tool-tip position and the turret datum is termed *tool offsets*, as shown in Fig. 14.2. For each of the tool used in turning centre, a separate tool offset data along the *X* and *Z* axes as shown in Fig. 14.2 would have to be entered.

On a production method, the tool offsets for each of the tools can be obtained by a number of ways. A single method, most generally used is the use of a mandrel of known size into the work holding device and touch the end faces and diameter by the tool and noting down the corresponding display positions with respect to home position as shown in Fig. 14.3. Then the offset values can be easily obtained by noting down as shown in Fig. 14.3. The actual reference point on the tool, which is used for programming in turning centres is shown in Fig. 14.4. The other method is to use a tool-setting probe shown in Fig. 11.13.

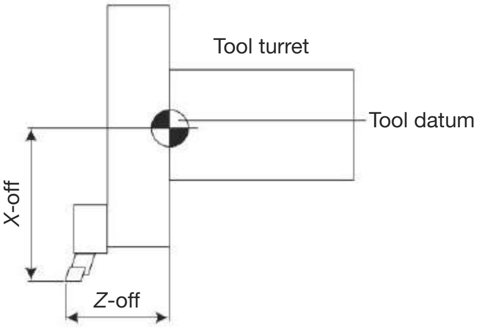


Fig. 14.2 *Tool offsets that are required for each tool in the turret*

14.4 GENERAL PROGRAMMING FUNCTIONS

14.4.1 Tool Function

Tools are selected in a program through the T word. The T word selects the turret station that is to be indexed to the cutting position and activates the tool offset register number. The tool offset register number selects the following from the tool offset file.

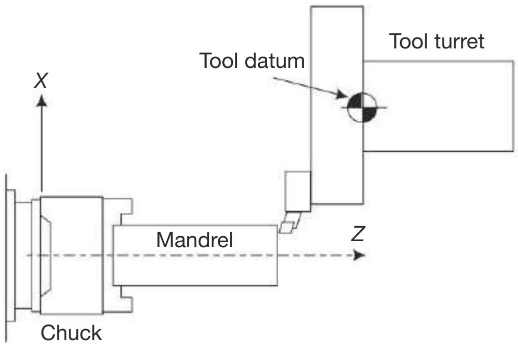


Fig. 14.3 *Measuring the tool offsets with a mandrel*

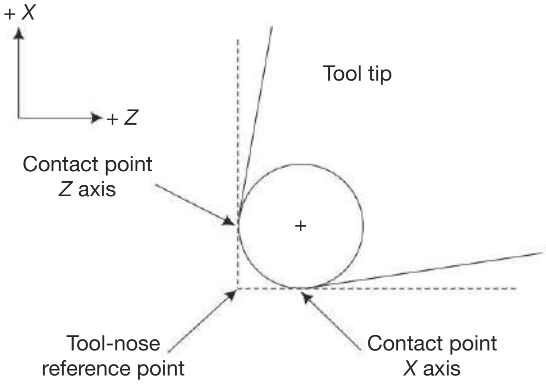


Fig. 14.4 *Measuring the tool offsets with a mandrel*

1. *X* and *Z* axis tool offsets
2. Tool nose radius value
3. Tool nose orientation number

The T word has the format T4, such as Tmmnn. The first two digits (mm) specify the turret station and the last two digits (nn) specify the location of the tool offsets. There is no need for any association between the tool number and the tool offset register number. The offset is cancelled by giving 00 in place of the offset register number. The register contains the following values. The register values can be entered through Manual Data Input (MDI) mode or through a separate tool offsets program.

Table 14.1 *Contents of the tool offset registers*

Offset register number	*X-axis offset amount*	*Z-axis offset amount*	*Tool nose radius*	*Tool nose radius direction*
01	120.040	15.20	0.8	0
02	121.410	18.65	0.8	0
.				
.				

The method of measuring the tool offset values is already presented earlier. Given below is an example of giving tool offset codes.

N080 M06 T0515;

Block N080 calls for turret station 5. Tool geometry offset register number is chosen as 15.

If no digits are programmed in the last two places, the turret will not index. Instead, the control will use the turret station number as an offset and activate that offset with present tool in place. This could result in a collision as the control is attempting the active tool with incorrect offsets.

14.4.2 Speed Function

The spindle speed can be specified in two ways, either direct rpm or in constant surface sped in m/min or ft/min. The constant surface speed option would be useful in turning, because the cutting speed is dictated by the diameter of the rotating workpiece which often varies during a turning process such as in taper turning, facing or when multiple steps are turned by the same tool, etc. Hence to maintain a constant cutting force and uniform surface finish it is necessary to vary the spindle rpm in direct proportion to the change in work diameter at the cutting-tool edge. This is achieved by constant surface speed option, which can be invoked by using the preparatory function G96. During power on, the default mode is the constant RPM, which can also be invoked by G97. G96 and G97 form a group of preparatory functions, which would cancel each other and otherwise would remain modal.

Since the spindle speed is continuously available with the diameter, there is a possibility that the rpm may go beyond the set rpm or a certain maximum limit for the equipment being used (for example limit on the chuck) though it is within machine tool's capability. In such situations, it is possible to specify an upper limit on the speed change by using the G50 command. The following example gives the program listing to demonstrate the use of S word.

```
N090   G50   S4000      Maximum spindle rpm is 4000
N100   G96   S200       Spindle speed is set at 200 m/min
N080   G97   S2500      Spindle speed is set at 2500 rpm
```

The S word is modal and once programmed it need not be programmed again until a different spindle speed is required. Do not program a decimal point with the S word.

14.4.3 Feed Specification

Similar to the speed, feed rate can be specified in two forms in the normal mm/min as well as mm/rev format, the conventional turning feed specification. These two formats can be obtained by using the following G-codes.

G98 feed in mm/min

G99 feed in mm/rev

During power on, G98 is the default mode. The feed rate remains unchanged until reprogrammed. One point of caution to be exercised with the two codes is that the format in which the values are read is dependent upon whether G98 is in force or G99. In G98 the format (metric) is F50 whereas in inch format it is F032. Thus in G98, feed values which are small are expected to be with a decimal point. Since the controller allows both decimal point and leading zero suppression, care has to be exercised for giving the correct value and format. When entering G98 mode, a new feed rate should be programmed. G98 is modal and cancels G99. The decimal point must be programmed. The following examples clearly explain the method.

```
G98   F   200   (200 mm/min)
```

The F word format for G99 is F1.6 in inch mode and F3.4 in metric mode.

```
G99    F1      (0.01 mm/rev)
G99    F1.0    (1.00 mm/rev)
G99    F100    (1.00 mm/rev)
```

The F word, which can be placed anywhere in the data block, remains unchanged until reprogrammed. If G00 is used to obtain the rapid traverse rate, be sure it is cancelled by another motion group G code after the rapid traverse move is completed. The Feed rate Override switch on the control panel modify the programmed feed rate from 0 per cent (Feed Hold) to 150 per cent. When **Dry Run** mode is active, the control causes all slide motion to take place at a feed rate selected with the Feed rate Override switches.

14.4.4 Units

Preparatory functions or G codes are used to specify the way the geometry needs to be generated. The G codes for units to be used in the programs are described here.

G20 Inch Input This is to be used when the dimensions are to be given in inch units. The command is modal and can be cancelled only by G21 (metric mode) command. G20 must be programmed in a block by itself. It is recommended that all programs written with inch dimensions should have the G20 code at the beginning of the program to ensure that the correct format is active. Here Fanuc deviates from the ISO standard of G70. The reason is that before ISO standardised G70/71, Fanuc was using these for special canned cycles, which are described in Chapter 15.

G21 Metric Input This is to be used when the dimensions are to be given in metric units.

14.4.5 Miscellaneous Functions

The M words convey action to the machine. They are known as miscellaneous functions and are designated by a programmed M word having the format M2. The M code may be placed anywhere in the data block. The miscellaneous functions generally used are similar to that of the machining centre.

M00 Program Stop The M00 command stops the program, stops the spindle and turns the coolant off. This function can be used for gauging the workpiece. Pressing Cycle Start on the control panel causes the program to continue. It is the programmer's responsibility to program for restarting the spindle or coolant pump when restarting the program after an M00 program stop.

M01 Optional Stop The functioning M01 command depends upon the condition of the Optional Stop push button on the control panel. If the optional stop has been activated before the block containing the M01 is read by the control then it functions similar to M00. If the optional stop push button has not been activated, the control will ignore the programmed M01 and will continue to execute the program. This function is useful when it is necessary to gauge the workpiece during set up. Pressing Cycle Start causes the program to continue.

M02 End of Program M02 indicates the end of a part program and should be programmed in the last block. It stops the spindle and turns the coolant off.

M03 Spindle Start in Clockwise Direction The M03 command causes the spindle to run in the clockwise direction at the programmed spindle speed (S word). M03 remains active until cancelled by other M codes or by pressing the Reset key or Emergency Stop push button.

M04 Spindle Start in Counter-Clockwise Direction The M04 command causes the spindle to run in the counter-clockwise direction at the programmed spindle speed (S word). M04 remains active until cancelled by other M codes or by pressing the Reset key or Emergency Stop push button.

M05 Spindle Stop The M05 command causes the spindle to stop and turns the coolant off, but will not stop axis motion unless G99 is active.

M08 Coolant ON M08 turns the coolant pump ON and remains active until cancelled by other M codes or by pressing the Reset key or Emergency Stop push button.

M09 Coolant OFF M09 turns the coolant pump off.

M13 Main Spindle Clockwise/Coolant ON The M13 command causes the spindle to run in the clockwise direction at the programmed spindle speed (S word) and turns the coolant pump on. M13 remains active until cancelled by other M codes or by pressing the Reset key or Emergency Stop push button. If M04 is programmed after M13, the spindle will run in the counter clockwise direction and the coolant pump will remain on.

M14 Main Spindle Counter-Clockwise/Coolant ON The M14 command causes the spindle to run in the counter clockwise direction at the programmed spindle speed (S word) and turns the coolant pump on. M14 remains active until cancelled by other M codes or by pressing the Reset key or Emergency Stop push button. If M03 is programmed after M14, the spindle will run in the clockwise direction and the coolant pump will remain on.

M30 End of Program and Rewind M30 indicates the end of a part program and should be programmed in the last block. It stops the spindle, turns the coolant off and rewinds the program to its beginning.

However, one point to be noted is that in a given block, only one M code is allowed. If more than one M code is programmed in a block from the keyboard or tape, the last M code entered will be the active M code.

14.4.6 Program Number

Each of the programs that is stored in the controller memory requires an identification, which is used while running and editing of the programs directly from the control console. This identification is specified in terms of a program number with O word address. The number can be a maximum of four digits. The program sequence should be as follows.

```
%;
O 3425;            Program start
N010 . . . . .
.......
.......
N100 M02;          program end
```

It is not necessary to program the leading zeros as these are automatically inserted by the control, when needed. The program number must be on the first line of the program. It may be programmed on a line by itself or it may be the first entry in the first data block. The part program numbers range from 1 to 8999.

End of program can also be designated by M30. Programs can also start without the program number (O word address). However, then the first block number encountered would be treated as the program number. Normally, when a new program is read by the controller, the old program already residing in the controller with the same name would be automatically erased without any warning to the user. Hence caution would have to be exercised by the user to check for the already existing program numbers in the memory of the controller.

14.4.7 Block Number

The N word provides an identification consisting of the letter N and up to four digits (0000 – 9999). It is not necessary to have a sequence number in any block. Though they can be placed anywhere in the block, it is customary to program them as the first word in the block, except when a Block Delete (/) is programmed. The N word does not affect machine operation. It is useful to identify a particular block for manual reference.

The numbering sequence can begin with any number, such as N0001. It is recommended that the programmer assign sequence numbers in intervals of five or ten so that additional blocks can be inserted into the program when necessary. This eliminates the necessity of reassigning sequence numbers after blocks are added to the program. Leading zeros may be omitted.

14.5 MOTION COMMANDS

The general motion commands given are similar in operation to that of the machining centre.

G00 Rapid Positioning G00 is used for rapid positioning simultaneously in all the axes. The feed rate programmed in the block would be overridden by the maximum allowed feed rate for the axes. The G00 command is modal. When it is programmed to move in both axes (*X* and *Z*), the axes execute a vectorial move at a traverse rate which is a result of the *X* and *Z* rapid traverse rates. An example program with reference to Fig. 14.5 demonstrates the usage.

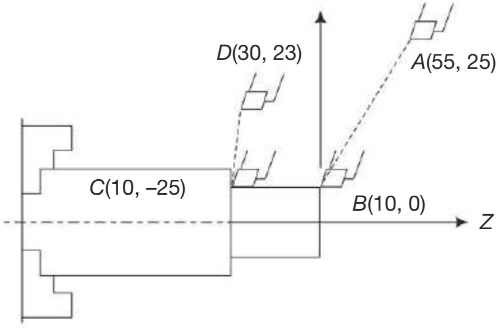


Fig. 14.5 *Rapid positioning*

Rapid Movement from A to B is

Absolute mode

```
N035    G00    X10.0    Z0
```

Incremental mode

```
N035    G00    U-45.0    W-25.0
```

Mixed mode

```
N035    G00    U-45.0    Z0
```

or

```
N035    G00    X10.0    W-25.0
```

G01 Linear Interpolation G01 is linear interpolation at a given feed rate. While programming a linear movement in one or two axes, the feed specified would be the vectorial along the resultant motion direction. As already explained, in a single block, the axes positions can be given either in absolute, incremental or absolute and incremental. An example is shown for all the cases in Fig. 14.6.

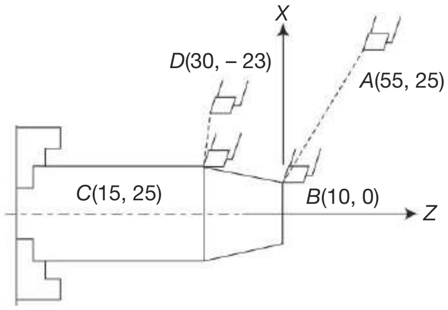


Fig. 14.6 *Linear interpolation with feed*

Programming from A–B–C–D (Radius Programming)

Absolute mode

```
N035    G00    X10.0    Z0
N040    G01    X15.0    Z-25.0
F120
N045    G00    X30.0    Z-23.0
```

Incremental mode

```
N035    G00    U-45.0    W-25.0
N040    G01    U5.0      W-25.0    F120
N045    G00    U15.0     W2.0
```

Mixed mode

```
N035    G00    U-45.0    Z0
N040    G01    U5.0      Z-25.0    F120
N045    G00    U15.0     Z-23.0
```

or

```
N035    G00    X10.0    W-25.0
```

```
N040    G01    X15.0    W-25.0    F120
N045    G00    X30.0    W2.0
```

The above example assumes that the programming being done is in the radius mode. However, it is also possible to programme the same using diameter mode. Here all the values given along X axis will have to be doubled even in the incremental mode. The selection of radius or diameter programming depends upon the system variables set during the integration of controller with the machine tool. The corresponding examples for diameter programming would be the following.

Programming from A–B–C–D (Diameter Programming)

Absolute mode

```
N035    G00    X20.0    Z0
N040    G01    X30.0    Z-25.0    F120
N045    G00    X60.0    Z-23.0
```

Incremental mode

```
N035    G00    U-90.0   W-25.0
N040    G01    U10.0    W-25.0    F120
N045    G00    U30.0    W2.0
```

G02 Circular Interpolation Clockwise

G03 Circular Interpolation Counter-Clockwise

G02 and G03 are used for specifying the circular interpolations. During the programming for circular interpolation there are two options: specification of the radius or centre coordinates. The radius can be directly specified using the R word address as shown in Fig. 14.7. Since in turning centres, the maximum arc that can be made is 180°, only positive radius values are to be specified.

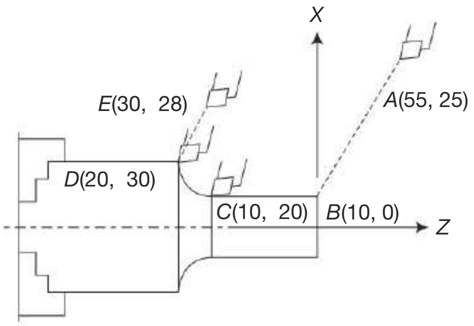


Fig. 14.7 *Circular interpolation with feed*

Programming from A–B–C–D–E (Radius Programming)

Absolute mode

```
N035    G00    X10.0    Z0
N040    G01    X10.0    Z-20.0    F120
N045    G02    X20.0    Z-30.0    R10.0    F100
N050    G00    X30.0    Z-28.0
```

When the centre coordinate is specified, its value is to be given in incremental form from the start point of the arc as shown in Fig. 14.8 with word addresses I and K. When radius is given, the arc can be machined with the given values, however when the centre coordinates are specified, there is a need to check for the validity of the data. With the specified centre coordinates, if the end point of the arc is not within a certain limit, (a tolerance value of 10 to 20 microns) normally the controller would reject the data.

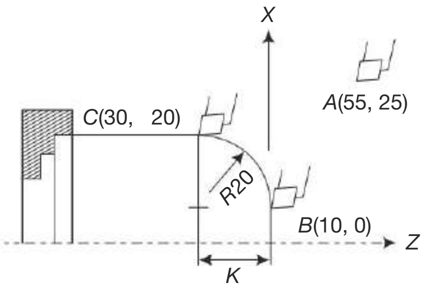


Fig. 14.8 *Circular interpolation using centre coordinates*

Programming from A–B–C (Radius Programming)

Absolute mode

```
N035    G00    X10.0    Z0
```

```
N045    G03    X30.0    Z-20.0    R20.0    F100
```

or

```
N035    G00    X10.0    Z0
N045    G03    X30.0    Z-20.0    I0       K-20.0    F100
```

G04 Dwell Dwell is used to specify a pause in axes motion for a specified time. This is a non-modal command, which causes the axes to stand still for the period of time specified in the block. The dwell time can be specified by using the X, U or P word address. However, with P word address no decimal point programming is allowed. Hence the value specified would be in milliseconds as the following example show for a delay of 4.5 s.

```
G04    X4.5
G04    X4500
G04    U4.5
G04    U4500
G04    P4500
```

14.5.1 Chamfer and Corner Radius

Chamfer and corner radius are more common in turning. They require an additional block to program normally. However, in Fanuc it is possible to program these using the linear interpolation code directly for the 45° chamfer and the 90° corner radius without an additional block, provided, motion prior to the chamfer or corner radius is along a single axis.

Chamfer The chamfer amount is specified by word address C, I or K as the case may be. The sign of C, I or K should be based on the direction in which the cutting tool has to move for making the chamfer. This should be clear from the examples shown in Fig. 14.9.

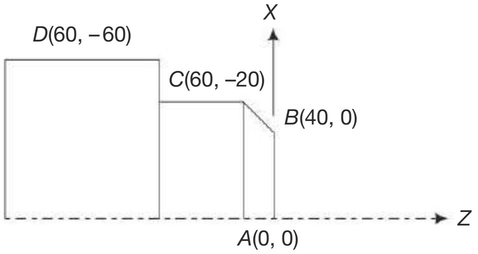


Fig. 14.9 *Example for chamfering*

Programming from A–B–C (Radius Programming)

Absolute mode

```
N045    G01    X60.0    Z0
C-20.0
```

or

```
N045    G01    X60.0    Z0    K-20.0
```

Programming from D–C–B (Radius Programming)

Absolute mode

```
N045    G01    X60.0    Z0    C-20.0
```

or

```
N045    G01    X60.0    Z0    I-20.0
```

Corner Radius Similar to the chamfer, corner radius both internal and external of 90° can be specified by combining R word address along with the G01 straight line interpolation as shown in Fig. 14.10. However, in this case also the corner radius can be specified in case where the tool movement prior to the radius formation should be along a single axis either *X* or *Z* only. The tool would move along the single axis to the programmed distance minus the corner radius. After that tool would move along the circle, with the given radius, till it reaches the programmed end point in the axis.

Programming from B–C–D (Radius Programming)

Absolute mode

```
N045    G01    X60.0    Z-40.0    R20.0
```

14.5.2 Tool-Nose Radius Compensation

While planning the final cut (finishing) it is necessary to take into consideration, the effect of tool-nose radius on the path to be taken by the turning tool. Imagine a situation similar to that of machining centre programming with a tool having a radius equal to the nose radius. Thus, the CLDATA to be generated is the actual path travelled by the tool-nose radius centre. The need for cutter radius compensation using the G codes 40, 41 and 42 was discussed with machining centre programming. Since the turning tool can be considered as a milling cutter with radius equal to the nose radius, it is imperative that the same will be applicable in turning as well. The use of these codes in turning is same as in milling and the same will be used in all the programming.

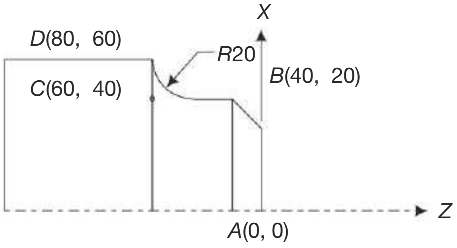


Fig. 14.10 *Example for corner radius*

Figure 14.11 shows the typical path to be taken for finish cutting of the profile. Most of the points on the cutter path can be easily derived by offsetting the nose radius. However, at the beginning and end of the inclined path it is necessary to make calculations based on simple trigonometry for the offset point from the original contour as explained in Chapter 13. Similarly, other cases can be easily derived.

However, by using the cutter compensation as shown in Fig. 14.11, the need for all complex calculations will be completely eliminated. The programming for the finishing cut will be the direct path of the actual contour to be machined. However, even after compensating the nose radius, the point of contact between the tool nose and the workpiece will still be along the nose radius periphery which will be changing depending upon the orientation of the tool with respect to the cut surface. For example as shown in Fig. 14.12(a), without the nose-radius correction, the tool will leave a small amount of material along the inclined path shown. For this purpose the turning centre controllers will provide the necessary correction. If the correction is active then the controller automatically compensates

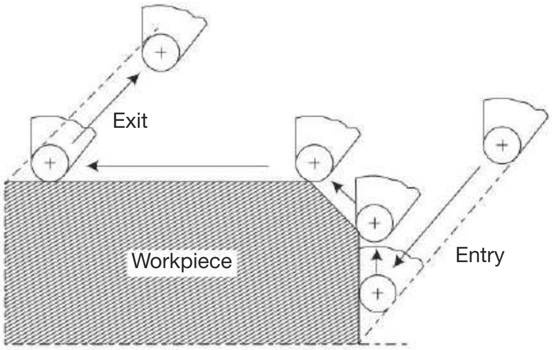
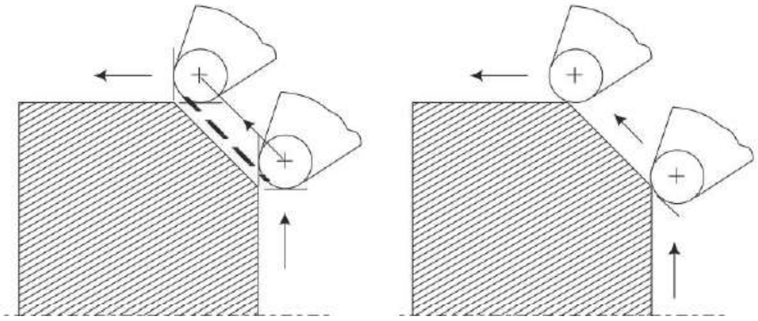


Fig. 14.11 *Offset path generated by the controller when using G42 for tool-nose radius compensation*



(a) Tool-nose radius compensation not active

(b) Tool-nose radius compensation active

Fig. 14.12 *Tool-nose radius correction* and removes the unwanted material as shown in Fig. 14.12(b). However, in order for the correction to be active, the controller will have to know the correct orientation of the nose radius with respect to the machining surface. For this purpose in the tool offset registers the tool-nose direction is included. The possible tool nose directions are identified and given numbers as shown in Fig. 14.13.

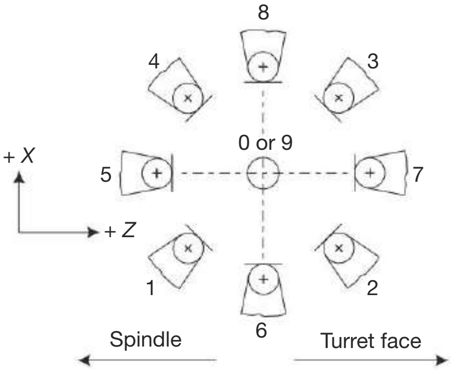


Fig. 14.13 *Tool-nose radius correction directions*

14.6 CUT PLANNING

In machining centres, most of the time the tool path is offset around the required geometry for the purpose of removing the extra material. However, in the case of turning centres, in particular when the blank happens to be rolled stock, it often requires a large amount of material to be removed before final contour is reached. It is possible to offset the final contour similar to the machining centre for generating the roughing cuts. However, this would be most inefficient since it would contain good amount of air cutting as the material may be missing. It is, therefore, necessary to plan roughing cuts not following the final profile but in a manner optimising the total tool movement to clear all the material. This is shown with the help of examples in Fig. 14.14 which are self-explanatory. A complete cut planning for external turning is shown in Fig. 14.14.

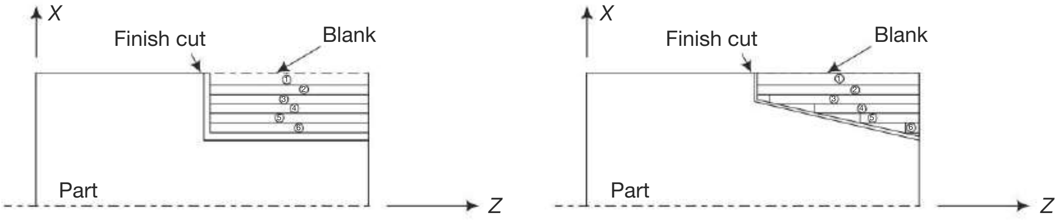


Fig. 14.14 *Cut planning for a turned component*

Example 14.1 Let us now have a look at an example for the complete planning of a component for machining on a lathe. The component drawing is given in Fig. 14.15. This part is to be machined from a rolled stock of 40-mm diameter.

First step in the planning process is the identification of complete process plan, which is shown in Table 14.1. After this a process drawing of the part is shown with the axes marked on the drawing (Fig.14.16).

Table 14.1 *Process plan for component shown in Fig. 14.15*

Operation	*Description*	*Tools*	*Cutting Speed, m/min*	*Feed, mm/rev*
10	Facing	T01, Facing tool	200	0.30
20	Rough turning	T02 Roughing tool	200	0.35
30	Finish turning	T03 Finishing tool	300	0.20

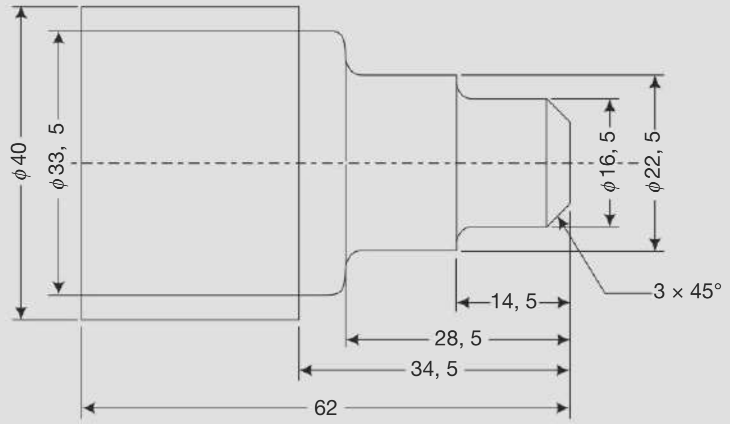


Fig. 14.15 *Part drawing to be used as an example for cut planning for a turned component*

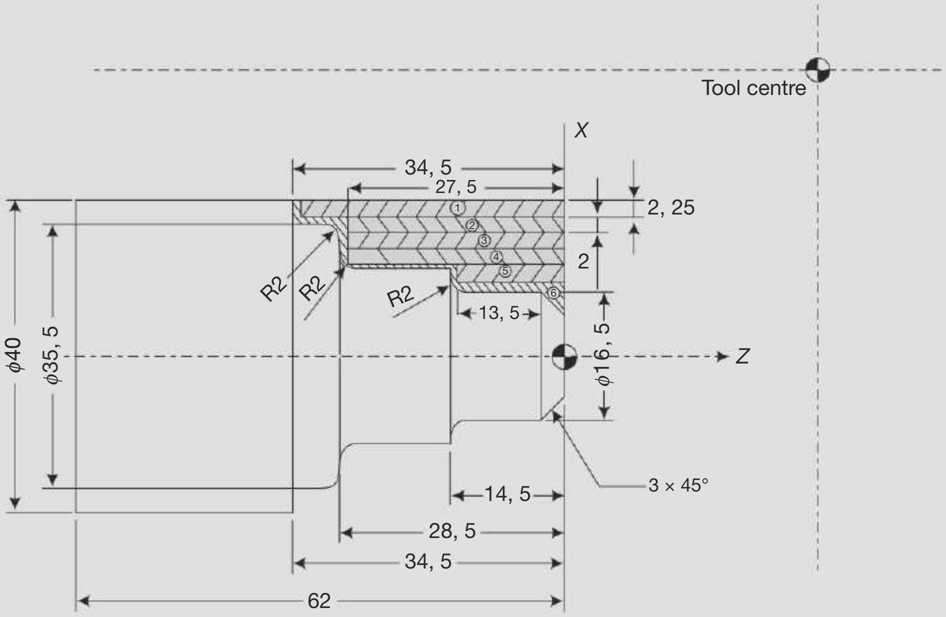


Fig. 14.16 *Cut planning of the part shown in Fig. 14.15*

In order to plan the cuts for rough turning by Tool No. T02, it is necessary to examine the total material to be removed after leaving a finish allowance of 1 mm all along the contour (shown with cross hatching in Fig. 14.16 next to part contour). Then the remaining part is divided into suitable cuts depending upon the maximum depth of cut allowed for the cutting tool chosen. In this case, assuming a depth of cut of 2 mm, it is possible to have 5 roughing cuts as marked in the Fig. 14.16.

To facilitate the calculation of the CLDATA the cut elements are shown separately in Fig. 14.17 for rough cutting and Fig. 14.18 for finish turning. From these the following program is developed with the necessary explanation provided as comments in parentheses in each block.

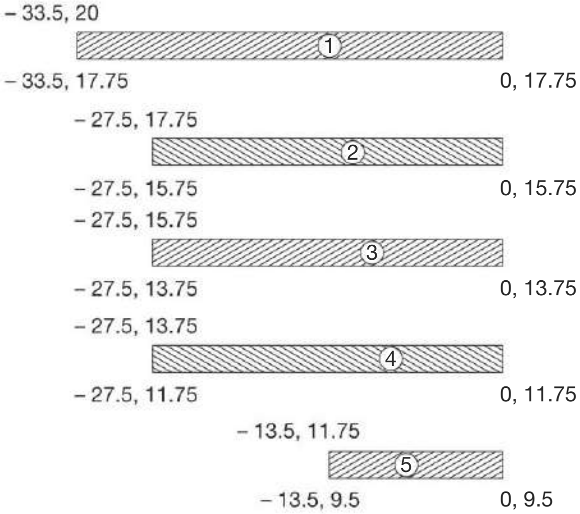


Fig. 14.17 *Rough cuts to be taken for the part shown in Fig. 14.15*

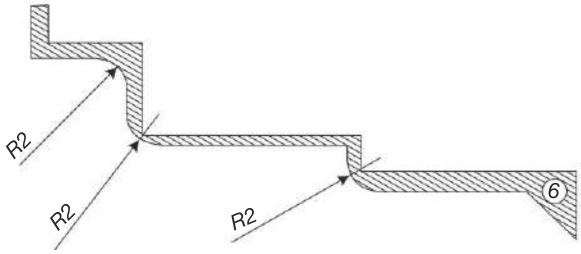


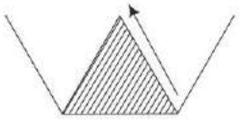
Fig. 14.18 *Finish cut to be taken for the part shown in Fig. 14.15*

```
%;
O8523;
N010   G21;                               (Metric units)
N015   G50    S4000;                      (Set maximum spindle speed at 4000 rpm)
N020   M06    T0101;                      (Facing tool)
N030   G96    S300;                       (Set the spindle speed)
N035   G00    X22.0   Z0      M03;        (Position at clearance plane for facing)
N040   G41;                               (Nose radius compensation left)
N045   G01    G98     X-0.5   F0.3;       (Feed to a small value beyond centre)
N050   G40    G00     X50.0   Z40.0;      (Position to the tool park position)
```

```
N055   S200   M06     T0202;                 (Rough turning tool)
N060   G42;                                  (Nose radius compensation right)
N065   G00    X17.75  Z2.0;                  (Position at clearance plane for rough turning)
N070   G01    Z-33.5  F0.35;                 (Turn till the end of cut 1)
N075   U-2.0  W-2.0;                         (Clear the tool away from the material)
N080   G00    Z2.0;                          (Rapid position till the clearance plane)
N085   X15.75;                               (Position for the next cut)
N090   G01    Z-27.5;                        (Turn till the end of cut 2)
N095   U-2.0  W-2.0;                         (Clear the tool away from the material)
N100   G00    Z2.0;                          (Rapid position till the clearance plane)
N105   X13.75;                               (Position for the next cut)
N110   G01    Z-27.5;                        (Turn till the end of cut 3)
N115   U-2.0  W-2.0;                         (Clear the tool away from the material)
N120   G00    Z2.0;                          (Rapid position till the clearance plane)
N125   X11.75;                               (Position for the next cut)
N130   G01    Z-27.5;                        (Turn till the end of cut 4)
N135   U-2.0  W-2.0;                         (Clear the tool away from the material)
N140   G00    Z2.0;                          (Rapid position till the clearance plane)
N145   X9.5;                                 (Position for the next cut)
N150   G01    Z-13.5;                        (Turn till the end of cut 5)
N155   U-2.0  W-2.0;                         (Clear the tool away from the material)
N160   G40    G00     X50.0   Z40.0;         (Position to the tool park position)
N165   S300   M06     T0303;                 (Finish turning tool-contour turning)
N170   G42;                                  (Nose radius compensation right)
N175   G00    X5.0    Z2.0;                  (Rapid position till the clearance plane)
N180   G01    X5.25   Z0      F0.15;         (Start the contour machining)
N185   X8.25  Z-3.0;
N190   Z-12.5;
N195   G02    X10.25  Z-14.5  R2.0;
N200   G01    X11.25;
N205   Z-26.5;
N210   G02    X13.25  Z-28.5  R2.0;
N215   G01    X14.75;
N220   G03    X16.75  Z-30.5  R2.0;
N225   G01    Z-34.5;
N230   X20.0;
N235   U-2.0  W-2.0   M05;                   (Clear the tool away from the material)
N240   G40    G00     X50.0   Z40.0;         (Position to the tool park position)
N250   M02;                                  (End of program)
```

14.7 THREAD CUTTING

Cutting screws is one of the most important tasks carried out in turning centres. There are a large number of thread forms that can be machined such as Whitworth, Acme, ISO metric, etc. While cutting threads, the feed is same as the lead of the pitch to be generated. The depth of cut in the case of thread cutting can be given in two ways: plunge cutting as shown in Fig. 14.19(b) or compound cutting as in Fig. 14.19(a).


(a) Plunge cutting

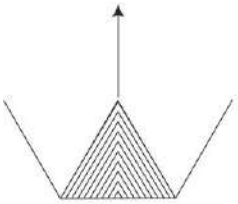

(b) Compound cutting

Fig. 14.19 *Depth of cut in thread cutting*

In case of plunge cutting, the cutting of the thread takes place along both the flanks of the tool. This would mean that the cutting tool would have to be provided with a zero or negative rake angle. In addition, the relief along the cutting edges cannot be provided in view of the form to be achieved. Cutting also takes place along a longer length of the tool. This creates difficulties in machining in terms of higher cutting forces and consequently chattering (violent vibrations). This results in poor surface finish and lower tool life, thus this method is not generally preferred.

With the compound feeding, the tool needs to be moved in both the directions (along the *X* and *Z* axes) simultaneously to position the tool tip along one flank of the thread. This configuration helps in more smooth flow of chips as the cutting takes place only along one cutting edge. This method therefore is much preferred compared to the earlier method.

For thread cutting operation, G33 code is used with uniform pitch. The function of G33 is to synchronise the spindle drive and the feed drive to achieve the necessary lead. However the start position need to be programmed carefully by the programmer to get the correct point depending upon the type of thread depth being given. The following example explains the use of G33 (Fig. 14.20).

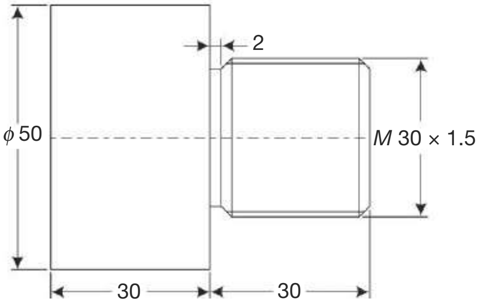


Fig. 14.20 *Example for thread cutting*

It is assumed that the plunge feed is being taken with depth of cut being varied as follows.

0.4 – 0.2 – 0.2 – 0.15 – 0.05 – 0.04 mm (Diameter programming)

```
N110   G97   S1000   M03;                  (Set the spindle speed as 1000 rpm)
N115   M06   T0202;                        (Get the thread cutting tool)
N120   G00   X29.2   Z4.0    M08;          (Position to the first cut)
N125   G33   Z-29.5  F1.5;                 (Complete the first cut)
N130   G00   U6.0;                         (Retract to a safe distance)
N135   Z4.0;                               (Go to the start point)
N140   X28.8;                              (Position to cut 2)
N145   G33   Z-29.5  F1.5;                 (Complete cut 2)
N150   G00   U6.0;
N155   Z4.0;
N160   X28.6;                              (Position to cut 3)
```

```
N165  G33   Z-29.5  F1.5;
N170  G00   U6.0;
N175  Z4.0;
N180  X28.4;                      (Position to cut 4)
N185  G33   Z-29.5  F1.5;
N190  G00   U6.0;
N195  Z4.0;
N200  X28.25;                     (Position to cut 5)
N205  G33   Z-29.5  F1.5;
N210  G00   U6.0;
N215  Z4.0;
N220  X28.2;                      (Position to cut 6)
N225  G33   Z-29.5  F1.5;
N230  G00   U6.0;
N235  Z4.0;
N240  X28.16;                     (Position to last cut)
N245  G33   Z-29.5  F1.5;
N250  G00   U6.0;
N255  Z4.0;
```

14.8 CANNED CYCLES

As noticed in the above examples, the length of the program becomes extremely long when a number of roughing cuts are involved. It may be noticed that a majority of the motions are actually repetitive in nature and therefore can be embedded into a canned cycle similar to the drilling series canned cycles seen in Chapter 13.

14.8.1 Turning Canned Cycle

A typical rough turning canned cycle is shown in Fig. 14.21. The tool is to be positioned at a point that just clears the work material. Typical distance of 1 to 2 mm along the *X* and *Z* axes is useful. The program then specifies the far corner of the turning motion to be taken using the *X* and *Z* coordinates. Then the canned cycle generates the following motions.

1. Move rapid to the *X* coordinate specified to take the depth of cut.
2. Move at feed rate (cutting metal) to the *Z* coordinate specified.
3. Retract at feed rate to the initial *X* position.
4. Retract rapid to the start point.

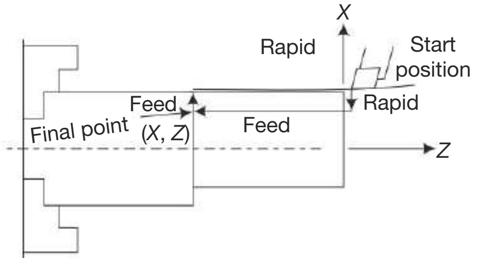


Fig. 14.21 *Rough turning canned cycle operation*

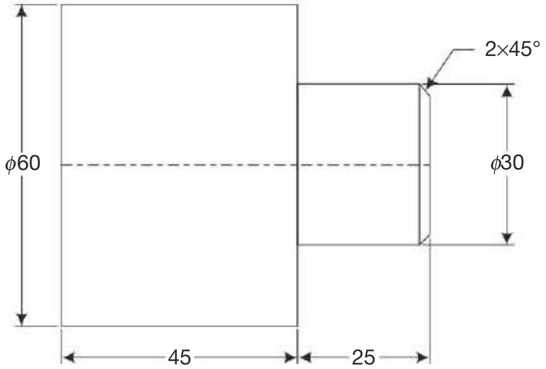


Fig. 14.22 *Rough turning canned cycle operation*

An example is shown (Fig. 14.22) below for the rough turning operation.

```
... ... .
N035   X32.0   Z1.0;                      (Position at start point of canned cycle)
N040   G97     F0.3;
N045   G90     X26.0   Z-25.0;            (Rectangular cut 1)
N050   X 22.0  Z-25.0;                    (Rectangular cut 2)
N055   X19.0   Z-25.0;                    (Rectangular cut 3)
N060   X16.0   Z-25.0;                    (Rectangular cut 4)
N065   G00     X40.0   Z50.0;
N070   M06     T0202;
N075   G42;
N080   Z2.0    X-1.0;                     (Finish contour)
N085   G01     Z0      F0.3;
N090   X13.0;
N095   X15.0   Z-2.0;
N100   Z-25.0;
N105   X31.0;
N110   G00     X40.0   Z50.0;
... ...
```

Part of the rough turning operation of the example shown in Fig. 14.14 is rewritten using the canned cycle below. It can be seen that there is a reduction in the amount of statements compared to the earlier case.

```
N055   S200    M06     T0202;             (Rough turning tool)
N060   G42;                               (Nose radius compensation right)
N065   G00     X21.0   Z2.0;              (Position at clearance plane for rough turning)
N070   G90     X 17.75 Z-33.5   F0.35;    (Turn till the end of cut 1)
N085   X15.75  Z-27.5;                    (Turn till the end of cut 2)
N105   X13.75  Z-27.5;                    (Turn till the end of cut 3)
```

```
N125    X11.75  Z-27.5;                  (Turn till the end of cut 4)
N145    X9.5    Z-13.5;                  (Turn till the end of cut 5)
N160    G40     G00     X50.0   Z40.0;   (Position to the tool park position)
```

It can be seen that this canned cycle will be able to take a rectangular cut as demonstrated. However, it is also possible to use it for tapered surfaces by making use of the I word address for movement along the *X* axis as shown in Fig. 14.23. In such cases the motion 2 as described above will be changed such that feeding will take place in *Z* axis as well as *X* axis (specified by incremental movement using the I word address). The I word address indicates the type of taper produced.

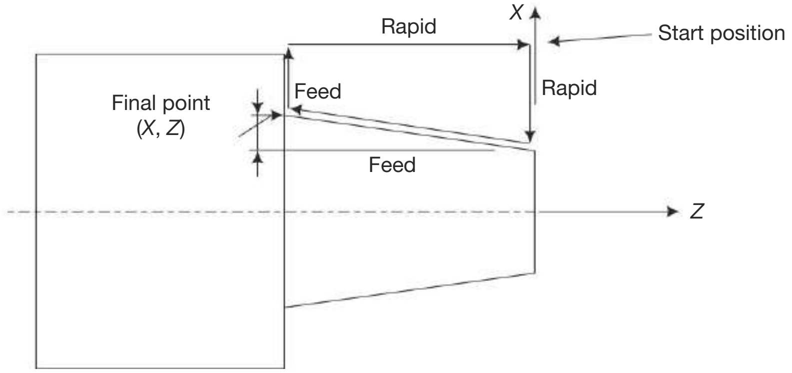


Fig. 14.23 *Rough turning canned cycle for tapered surfaces*

The same canned cycle can also be used for internal turning (boring) using a boring bar. No change in programming is required, but care should be taken to position the boring bar correctly at the clearance plane which is now inside the workpiece.

14.8.2 Facing Canned Cycle

Another canned cycle similar to the roughing cycle is available, except that the operation is to be done on the face of the workpiece rather than the longitudinal surface, as shown in Fig. 14.24. This is useful for taking a number of facing cuts. However, it is also possible to cut a tapered face using the K word address for movement along the *Z*-axis during the feeding motion. The actual usage is as follows.

```
G94     X ... .. Z... ...     K ... ...  F... ...
        Coordinates           For taper
        of cut                face
```

The actual motions performed are as follows.

1. Move rapid to the Z coordinate specified to take the depth of cut.
2. Move at feed rate (cutting metal) to the X coordinate specified.
3. Retract at feed rate to the initial Z position.
4. Retract rapid to the start point.

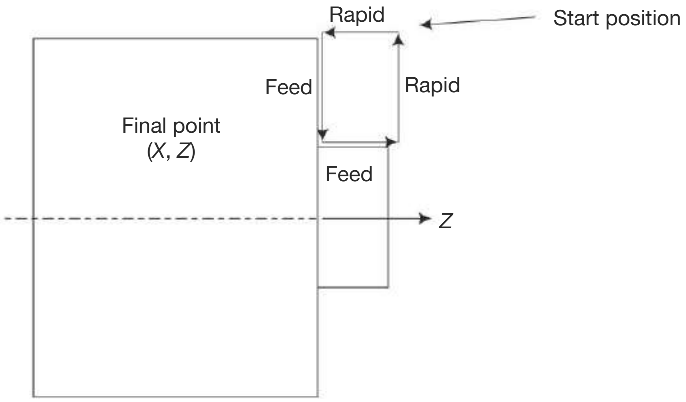


Fig. 14.24 *Face cutting canned cycle operation*

14.8.3 Thread Cutting Canned Cycle G92

The thread cutting cycle is similar to the rough turning cycle G90 in terms of the motions performed. The only difference is that during the cutting operation, G33 will be initiated in place of G01 for synchronising the spindle drive and the feed drive. The thread chasing needs to be handled by the programmer as explained earlier.

The thread cutting part shown in Fig. 14.20 is rewritten using the canned cycle below. It can be seen that there is a reduction in the amount of statements compared to the earlier case.

```
N110   G97    S1000  M03;                (Set the spindle speed as 1000 rpm)
N115   M06    T0202;                     (Get the thread cutting tool)
N120   G00    X29.2  Z4.0    M08;        (Position to first cut)
N125   G92    X29.2  Z-29.5  F1.5;       (Complete first cut)
N145   X28.8  Z-29.5;                    (Complete cut 2)
N165   X28.6  Z-29.5;                    (Complete cut 3)
N185   X28.4  Z-29.5;                    (Complete cut 4)
N205   X28.25 Z-29.5;                    (Complete cut 5)
N225   X28.2  Z-29.5;                    (Complete cut 6)
N245   X28.16 Z-29.5;                    (Complete last cut 5)
```

Summary

- Turning-centre programming follows the general pattern of programming approach as used in machining centre programming. However, there are certain fundamental differences between the two because of the nature of cutting tools used and the fact that the workpiece rotates rather than the tool in turning centres.
- Tape formats used are similar to that of the milling, with practically the same word addresses being used in turning centres as well.
- The axes system is predominantly two axes (*X* and *Z*) for most of the turning centres. The possibility of multiple turrets or multiple spindles some times adds more axes in turning centres.
- Programming the motion commands is similar to machining centres, but care has to be taken since axes system is *X* and *Z*, which means G18 is in operation.

- Tool nose radius compensation need to be effective to reduce the undercutting of the part contour depending upon the direction of motion of the tool.
- Since many turned parts may be starting from bar stock, substantial amount of material need to be removed in roughing and finishing cuts, that need proper cut planning to reduce the overall machining time.
- During thread cutting, synchronisation between speed and feed is achieved by using G33 command. The type of depth of cut to be used need to be planned by programmer.
- Turning canned cycle (turning and facing) reduce the length of the programs by embedding the repeated command into the canned cycle.

Questions

1. Give the differences in part programming to be considered between CNC turning centres and CNC machining centres.
2. Briefly explain about the datums to be considered in CNC turning centres.
3. How do you set the tool offsets in case of turning centres? Explain with an example.
4. How is the tool nose radius relevant in CNC turning centre programming? Explain with a simple example.
5. Describe with an example any one canned cycle in the Fanuc CNC turning controller.
6. The following components (Figs 14.25 to 14.36) are to be made using a CNC turning centre equipped with a FANUC controller. Prepare the part programs to completely machine the parts from rolled stock. The work material is AISI 1040 steel. Clearly show the set point and axes on the sketch of the part. Prepare the planning sheet also.

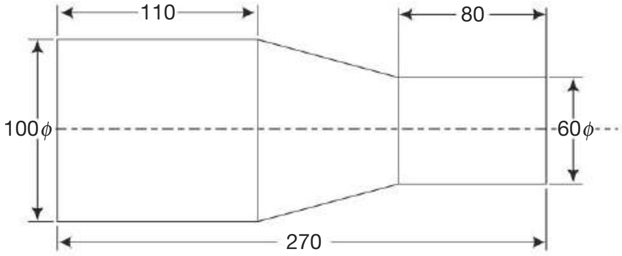


Fig. 14.25

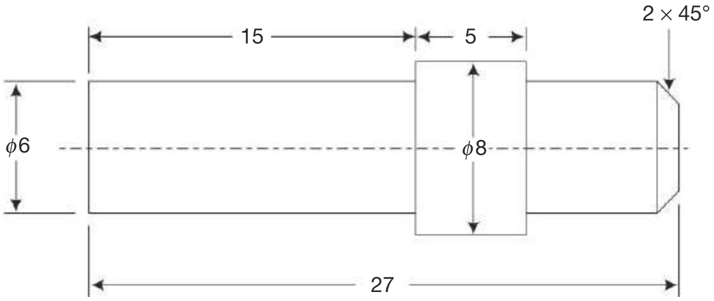


Fig. 14.26

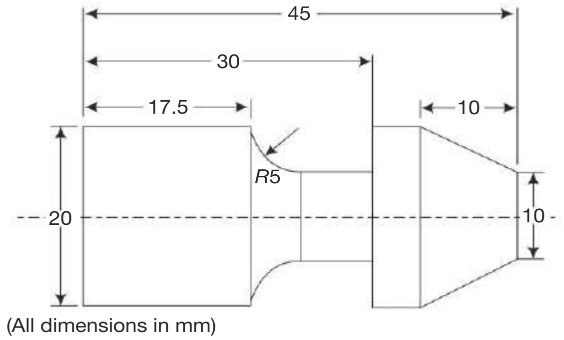


Fig. 14.27

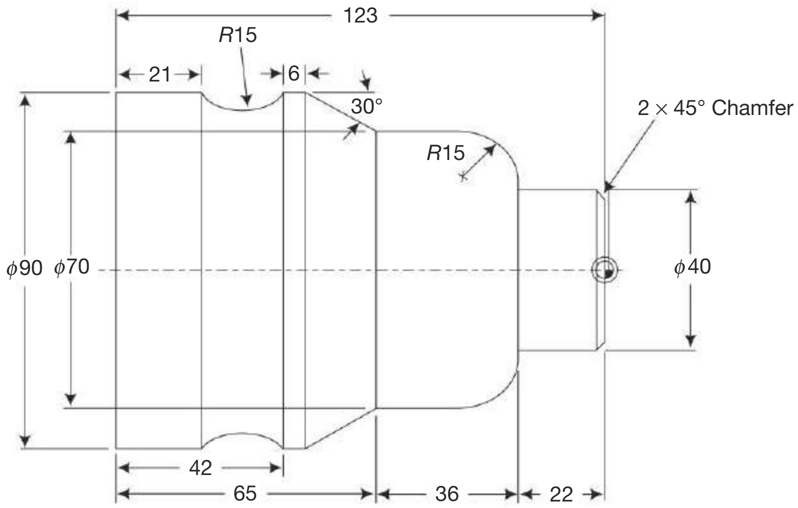


Fig. 14.28

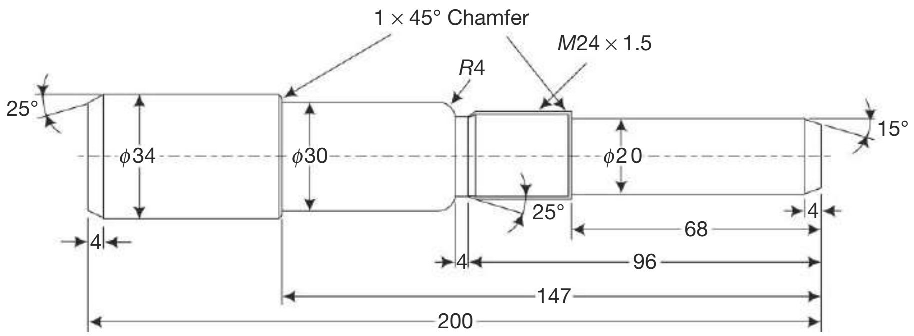


Fig. 14.29

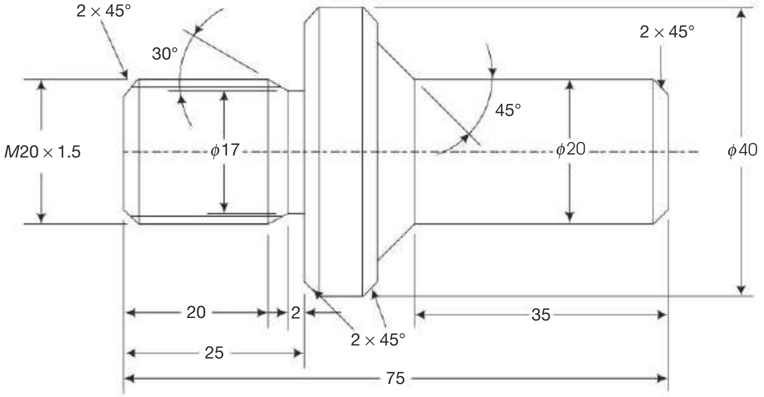


Fig. 14.30

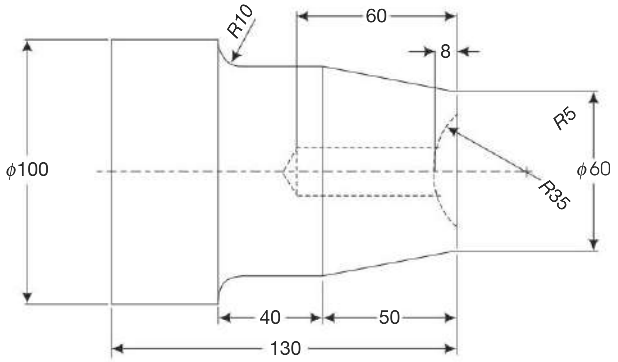


Fig. 14.31

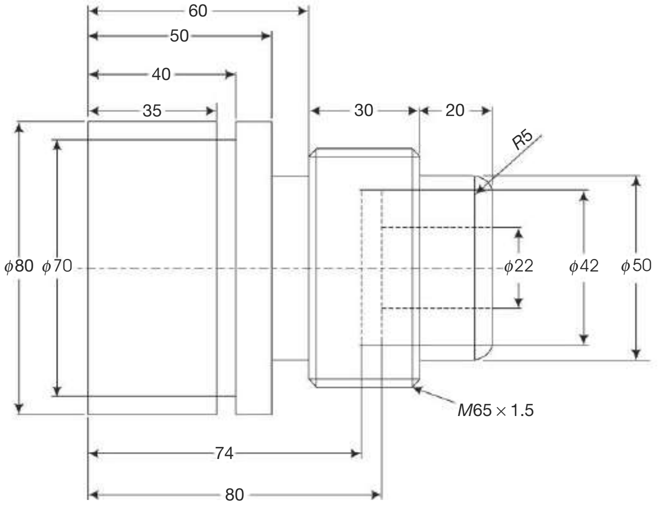


Fig. 14.32

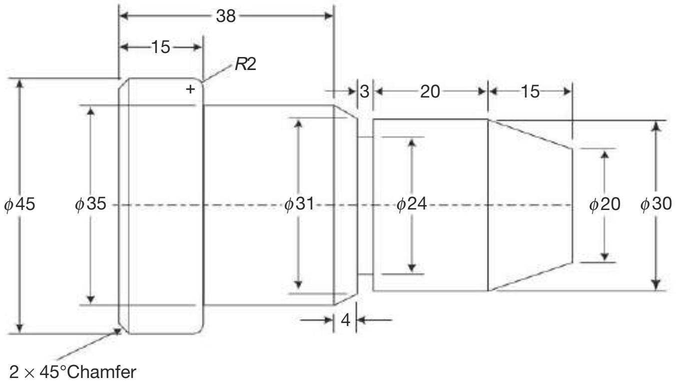
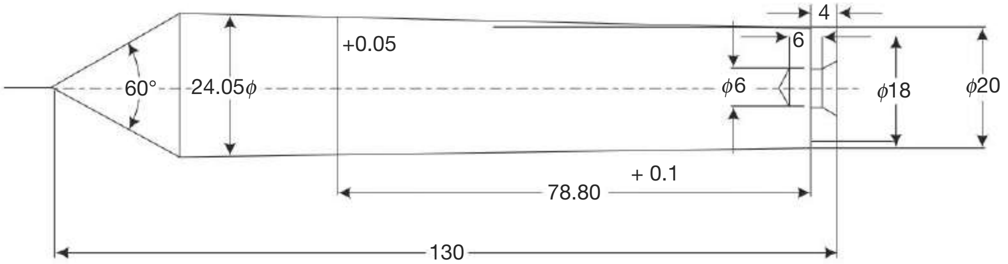


Fig. 14.33



Lathe dead centre
all dimensions in mm
T103 steel

Fig. 14.34

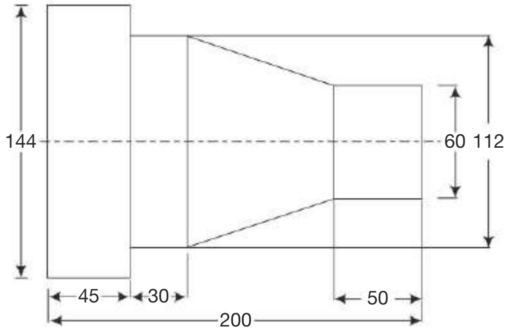


Fig. 14.35

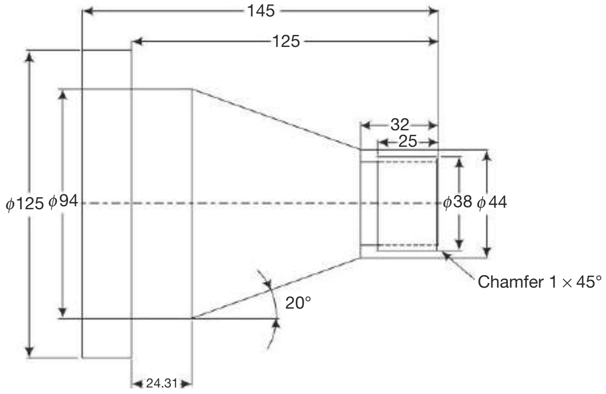


Fig. 14.36

15

ADVANCED PART-PROGRAMMING METHODS

Objectives

The manual part-programming methods seen in the previous chapters are suitable for simple shapes, but in case of parts with complex shapes or those having considerable symmetry there is need for special programming facilities to simplify programming. With the advent of more powerful microprocessors, the capability of the machine control unit is increasing. This chapter introduces some of these advanced programming facilities that are non-standard and hence will have to be discussed with reference to the particular control system in question. After completing the study of this chapter, the reader should be able to

- Program parts using polar coordinates
- Understand the use of parameters in part programs
- Develop programs using looping such as IF and DO
- Use subroutines to combine repeated instructions
- Use geometric transformation such as mirroring and scaling to exploit the symmetry in part geometry
- Appreciate the use of special canned cycles that can utilise the part geometry information directly to create complex part programs.

15.1 POLAR COORDINATES

There are times, when the coordinates of a given point in a drawing are available in the form of polar coordinates rather than the cartesian coordinates that are generally used for CNC programming. In such cases the programmer has to convert the polar coordinates into cartesian coordinates and then prepare the program. However, some controllers provide for programming directly in polar coordinates in which case no calculations need to be done.

MAHO

In the Philips 532 controller used in MAHO machine tools the polar coordinates are directly specified using the following word addresses.

```
N015      G00      B1 = - 161.565      L1 = 63.245
N020      G01      B2 = 116.565        L2 = 67.082    F250
```

B1 = Incremental angle
B2 = Absolute angle
L1 = Incremental radius
L2 = Absolute radius

The usage is like any other word address used in the part program.

15.2 PARAMETERS

A number of times the programmers are faced with a large amount of calculations needed to be done before they can complete the writing of the part program. Many of these calculations can be quite complex. The complex calculations once completed are difficult to trace after sometime, if any debugging of the program is to be done. Further, the accuracy of calculations is always a problem. Hence many of the controllers allow for defining variables in a program in such a way that the arithmetic calculations can be carried using these variables. Further, these variables can be used to specify the various values in association with the word addresses in a program. These will allow for a large amount of complex programming. It is therefore necessary to understand the concept of these variables usage. A comparative evaluation of these variables as used by various controller manufacturers is given Table 15.1.

Table 15.1 *Paramater specifications*

Operator	*General Electric*	*Fanuc 0, 6*	*Maho*	*Sinumeric 3*	*Anilam*	*Heidenhain*
Parameter word	P	#	E	R	V	Q
Maximum number	99	9900	255	99	99	99
Numbers not to be used	N	Y	N	N	N	N

Variables can be used to carry out the arithmetic operations as well as assign the values for the word addresses in a given block. In the following the use of parameters with reference to the specific controllers is discussed.

15.2.1 Parameters in MAHO

Parameters in MAHO 532 controller are identified by the word address E. A maximum of 255 variables can be used in a program and they are numbered as E1 to E255.

In addition to this all the arithmetic calculations can be carried out in the program using the following operators and functions.

`E1 = 120.5`	Assignment
`E1 = E1 + E2`	Addition
`E1 = E2 - E5`	Subtraction
`E1 = E3 * E5`	Multiplication

`E1 = E5 : E8`	Division
`E2 = E1 ^ 4`	Exponentiation

The priority of arithmetic operations is

function
exponentiation
multiplication, division
addition, subtraction

The arithmetic operations can be nested using parentheses as shown.

```
E1 = E2 + E3 * ((E4 - E6) : E12 ) + E23
```

Arithmetic functions available are

`E1 = SIN (E2)`	Sine with argument in degrees
`E1 = SIN (E2 + E3 * (E5 - E7))`	
`E2 = COS (E3)`	Cosine with argument in degrees
`E3 = TAN (E4)`	Tangent with argument in degrees
`E4 = ASIN (E6)`	Arc sine with result in degrees
`E5 = ACOS (E6)`	Arc cosine with result in degrees
`E6 = ATAN (E7)`	Arc tangent with result in degrees
`E7 = SQRT (E8)`	Square root
`E8 = ABS (E9)`	Absolute value
`E9 = INT (E12)`	Rounding based on the fraction, if more than 0.5 incremented to the next higher integer and if less disregards the fraction.
`E9 = INT (3.452)`	E9 = 3
`E9 = INT (3.512)`	E9 = 4

15.2.2 Parameters in Fanuc

Parameters in Fanuc controllers are identified by # and can vary from 0000 to 9999. There are some variables, which have specified meaning and are called system variables. They cannot be used by the programmers. They are as follows.

`#1000 to #1032`	
`#1100 to #1132`	
`#2000 to #2200`	
`#2500 to #2506`	
`#2600 to #2606`	
`#2700 to #2706`	
`#2800 to #2806`	
`#3000 to #3012`	
`#3901 to #3902`	
`#4001 to #4120`	
`#5001 to #5104`	
`#8000 to #8150`	for G codes and word address information

In addition to this, all the arithmetic calculations can be carried out in the program using the following operators and functions.

`#1 = 120.5`	Assignment
`#1 = #1 + #2`	Addition
`#1 = #2 - #5`	Subtraction
`#1 = #3 * #5`	Multiplication
`#1 = #5 / #8`	Division

The priority of arithmetic operations is

- function
- multiplication, division
- addition, subtraction

The arithmetic operations can be nested using parentheses up to 5 levels of nesting as shown

```
#1 = #2 + #3 * [[#4 - #6] / #12] + #23
```

Arithmetic functions available are the following.

`#1 = SIN [#2]`	Sine with argument in degrees
`#1 = SIN [#2 + #3 * [#5 - #7]]`	
`#2 = COS [#3]`	Cosine with argument in degrees
`#3 = TAN [#4]`	Tangent with argument in degrees
`#4 = ATAN [#6]`	Arc tangent with result in degrees
`#5 = SQRT [#8]`	Square root
`#6 = ABS [#9]`	Absolute value
`#7 = ROUND [#12]`	Rounding based on the fraction, if more than 0.5 incremented to the next higher integer and if less disregards the fraction.
`#7 = ROUND [3.452]`	#7 = 3
`#7 = ROUND [3.512]`	#7 = 4
`#8 = FIX [#11]`	Discard all the fractions
`#8 = FIX [4.98]`	#8 = 4
`#9 = FUP [#10]`	Any fraction rounded off to the next higher integer
`#9 = FUP [4.12]`	#9 = 5

15.2.3 Parameters in Sinumeric

Parameters in Sinumeric controls are identified by the R word address. They range from 00 to 99. In a block a maximum number of 10 R parameters are permissible. Their values can go from 10^{-8} to 10^{27}. A space is used for assignment and addition of variable values.

`R01 120.5`	Assignment	R01 = 120.5
`R01 R02`	Addition	R01 = R01 + R02
`R01 - R02`	Subtraction	R01 = R01 – R02
`R01 . R03`	Multiplication	R01 = R01 × R03
`R01 / R08`	Division	R01 = R01/R08

In a block	*Meaning*
`R01 12.54`	R01 = 12.54
`X R01`	X = R01 = 12.54
`X R01 10.52`	X = R01 + 10.52 = 12.54 + 10.52 = 23.06 (X23.06)
`R02 18.45`	R02 = 18.45
`Z 24.68-R02`	Z = 24.68 – 18.45 = 06.23
`R01 R02`	R01 = R01 + R02 = 12.54 + 18.45 = 30.99
`R01-R02`	R01 = R01 – R02 = 30.99 – 18.45 = 12.54
`R01.R02`	R01 = R01 * R02 = 12.54 * 18.45 = 231.363
`R03 12.54`	R03 = 12.54
`R01/R02`	R01 = R01 / R02 = 12.54 / 18.45 = 0.679675
`R12 -10`	R12 = – 10
`R14 81.`	R14 = 81.0
`Z 45.-R12 R03`	Z = 45.0 – R12 + R03 = 45.0 + 10 + 12.54 = 67.54
`R02 15 R03`	R02 = 15 + R03

Special functions are identified by the symbol @ followed by two numerals.

To get the square root	*@10*
`N10 R15 25;`	R15 = 25
`N15 @10 R15;`	R15 = SQRT[R15] = SQRT [25] = 5

The value contained in R15 should be > 0.00000001 and < 99999999.0

To get the sine	*@15*
`N25 R26 30;`	R26 = 30
`N35 @15 R26;`	R26 = SIN [R26] = SIN [30] = 0.5

The angle is to be specified in degrees.

The value contained in R26 should be greater than –359.99999 and less than +359.99999

To get the inverse tangent	*@18*
`N55 R28 1.;`	R28 = 1.0
`N65 @18 R28;`	R28 = ATAN [R28] = ATAN [1.0] = 45

Only one parameter with @ sign may be present per block. A comparison of the use of arithmetic and logical operators between the various controllers is given in Table 15.2.

The use of these parameters become more clear when they are used in the later parts of this chapter.

15.3 LOOPING AND JUMPING

Many times it becomes necessary to repeat a set of blocks in a program. For example, in hole-making operations, it may be necessary first to centre drill, pre-drill, counter sink and ream the same set of holes. In such cases it becomes necessary to have a facility where by the geometric part could be repeated while the operation could be specified as required. This helps in reducing program size. This repetition is done in many ways. Also it depends upon the particular controller in use. The facilities that are generally provided are

- Unconditional jump
- Conditional jump
- Do loop

Table 15.2 *Comparison of arithmetic operations in CNC controllers*

Operator	*Fanuc 0*	*Fanuc 6*	*Sinumeric 3*	*Heidenhain*	*MAHO*
Assignment	H01	=	Space	D00	=
Addition	H02	+	+	D01	+
Subtract	H03	–	–	D02	–
Multiply	H04	*	.	D03	*
Divide	H05	/	/	D04	:
Exponentiation					^
Square root	H21	SQRT	@10	D05	SQRT
Root of addition of two squares	H27			D08	
Sine	H31	SIN	@15	D06	SIN
Cosine	H32	COS		D07	COS
Tangent	H33	TAN			TAN
Arc tangent	H34	ATAN	@18		ATAN
Absolute value	H22	ABS			ABS
Rounding of fractions		ROUND			INT
Logical OR	H11	OR			
Logical AND	H12	AND			
Jump to	H80	GOTO	@00		
Logical IF EQ	H81	EQ	@01	D09	=
Logical IF GT	H83	GT	@02	D11	>
Logical IF GE	H85	GE	@03		>=

Unconditional Jump The execution of a part program is done sequentially. However, this execution can be changed by giving an unconditional jump such as GOTO, which unconditionally transfers the program execution to a different block identified in the jump block. Since the unconditional jump block transfers the execution to a block identified in the jump block, the blocks that are next to a jump block will not be executed, unless another jump block transfers the execution to that block. The jump block can transfer the execution either in the forward or backward direction. It will be more clear when a few examples are seen later with specific controllers.

Generally the understanding of a program with a number of unconditional jumps becomes difficult. Hence it is not advisable to use this when compared to the other two.

Conditional Jump In the case of conditional jump a variable or an arithmetic expression is evaluated for its numerical value. Depending upon the numerical value, the jump to a new block is executed.

DO Loop This is a more elegant way of looping, where a set of blocks will be repeated a number of times using a DO index. It may also be possible to have a structure wherein instead of an index a condition could be specified, based on which the set of statements could be repeated.

15.3.1 Looping and Jumping in MAHO

Conditional Jump Conditional jump is achieved by using the G29 code. Based on a certain condition being satisfied, the program will jump to a section of the program specified in the block. The usage of the instruction is as follows.

```
N120 G29 E5 N=145 (Jump to block N145 when E5 > 0)
N125…….
N130…….
N140…….
N145…….
```

When the value of E5 (or any other E parameter used in that position) is less than or equal to 0, no jump takes place.

```
N120 …….
N125…….
N130…….
N140…….
N145 G29 E16 N=120 (Jump to block N120 when E16 > 0)
```

The jump can be executed backwards as well as forward as shown here.

DO Loop Execution of a set of blocks a number of times in a fashion similar to a DO loop is done by using G14 code in Maho. The format of usage is

N... G14 N1= ... . N2 = ... J...

N1 = Starting block number of the DO loop

N2 = Ending block number of the DO loop

J word address is used for the repetition count (number of times the loop is to be repeated). If it is to be repeated only once, the J word can be omitted. A typical example is shown to demonetrate the usage.

```
N120 G14 N1=130 N2=160 J5
N130
N135
N140
N150
N160
N170
```

15.3.2 Looping and Jumping in FANUC

Unconditional Jump, GOTO In Fanuc controls, the unconditional jump is specified by the vocabulary word GOTO similar to the normal programming languages. Typical format is as shown.

```
N400 G01 X #4;
N410 GOTO 550;
...
...
N540 ....
N550 X.. Y..;
N300 G01 X #4;
...
N410 GOTO 300;
...
...
N550 X.. Y..;
```

The jumping can be done in both forward as well as backward directions. However, jumping in the reverse direction takes more time than in the forward direction.

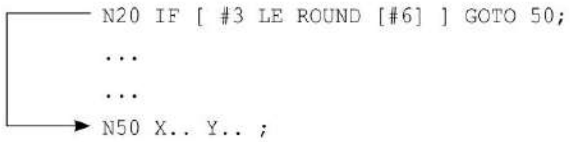
Conditional Jump, IF The construct used for the conditional jump is as follows.

IF [*expression*] GOTO number

where *expression* can be any valid logical expression that can be made using the logical operators as follows.

EQ	Equal
NE	Not equal
GT	Greater than
GE	Greater than or equal
LT	Less than
LE	Less than or equal

Number refers to the block number to which the execution of the program gets transferred when the *expression* is satisfied. A typical example is shown below to demonstrate the usage.



```
N20 IF [ #3 LE ROUND [#6] ] GOTO 50;
...
...
N50 X.. Y.. ;
```

DO Loop The statements are repeated while the condition is satisfied.

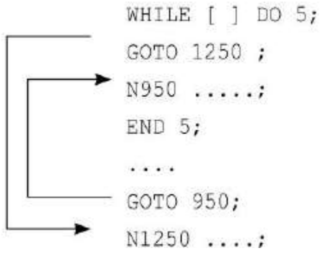
```
WHILE [ #3 = 20 ] DO 1;
.....;
.....;
END 1;
```

The DO loop index can be from 1 to 9.

They can be nested up to 3 levels.

A DO must be specified before END comes.

The WHILE condition can be any of the arithmetic condition that can be formed.



```
WHILE [ ] DO 5;
GOTO 1250 ;
N950 .....;
END 5;
....
GOTO 950;
N1250 ....;
```

This is not allowed as jumping into a DO loop.

However, jumping out of a DO loop is allowed.

15.3.3 Looping and Jumping in Sinumeric

Unconditional Branching, @00 In Sinumeric controls the unconditional branching is specified using the special word @00. A few examples are given below to demonstrate the use of this word.

```
N100 X... Z....;
..
..
..
N200 @00 -100;            (GOTO N100 backwards)
```

```
N100 X... Z....;
N105 X.. Y..;
..
N190 R12 5;
N200 @00 -100 R12;        GOTO N [100 + R12] backwards
```

```
N100 X... Z....;
..
..
N190 R12 5;
N200 @00 +260;            GOTO N [260] forward
..
..
N260 X.. Y.. ;
```

Conditional Branching In Sinumeric controls the uconditional branching is specified using the special word @01, @02 and @03 followed by the logical condition. A few examples are given below to demonstrate the use of this word.

@01 for expression when the variables are equal

```
N100 X... Z....;
..
..
..
N200 @01 -100 R01 R02;          GOTO N100 backwards IF [R01 = R02]
```

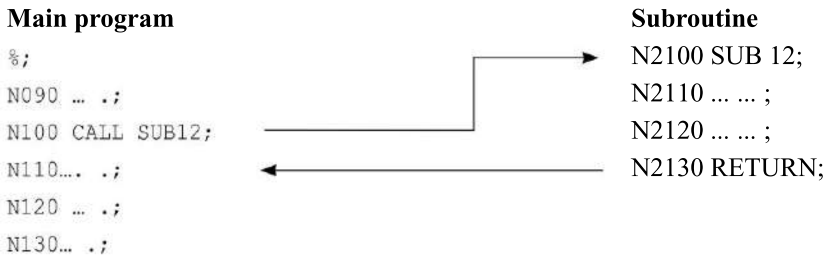
```
N110 X... Z....;
N120 X.. Y.. ;
..
..
N190 R12 10;
N200 @01 -110 R12 R01 R02;      GOTO N [110 + R12] backwards
                                IF [R01 = R02]
```

Similar conditional branching would be used with @02 for expression when the variables are with the condition greater than and @03 for expression when the variables are with the condition greater than or equal.

15.4 SUBROUTINES

One of the major improvements in the part-programming efficiency is the ability to reuse part of the program a number of times in a particular program as well as in a number of other programs. That is the concept being used in the subprograms or macros as they are called by the various manufacturers. These subprograms allow the programmer to code more efficiently the often used procedures into small pieces of separate code and store them in the controller memory permanently. Any user will simply have to call these subroutines for using them in his programs.

As explained earlier in looping the normal execution of a program is sequential unless altered by a special feature such as GOTO. Same thing happens in case of the use of subroutine as well. However there is major difference in the case of subroutine, i.e. though the program jumps to a subroutine like a jump statement but it jumps outside the main program. However, after executing the subroutine the program execution returns back to the block next to the subroutine calling block. This is shown schematically in the following way.



Main program	**Subroutine**
`%;`	N2100 SUB 12;
`N090 … .;`	N2110 ... ... ;
`N100 CALL SUB12;`	N2120 ... ... ;
`N110…. .;`	N2130 RETURN;
`N120 … .;`	
`N130… .;`	

It is also possible to call another subroutine from a subroutine, call the same subroutine a number of times within a program, etc. The individual controller implementations are shown in coming sections for more clarification.

15.4.1 Subroutines in MAHO

In the MAHO controller the program memory is divided into several parts, with each holding a separate type of program. The memory segment will be identified in the beginning of the program such that they are stored in the appropriate places.

PM –program memory, where the main programs are stored.

MM –subroutine memory or macro memory, where subroutines are stored.

The address is given in the following way.

Main Program	**Subroutine**
`%`	%
`PM`	MM
`N9352`	N9102
`N010… ;`	N010 ... ;
`….. .`	... ..
`N120 M30 ;`	N050... ;

To call a subprogram from the main program, G22 is used. The usage is as follows.

`N120 G22 N=9102` (9102 is the subprogram number)

It also allows to call another subprogram from a subprogram up to a nesting level of 8.

The following example allows for drilling any number of holes along a circle (Fig. 15.1) which are all equi-distant. The inputs are the following.

E1 = X coordinate of the centre of the circle
E2 = Y coordinate of the centre of the circle
E3 = Diameter of the circle
E6 = Total number of points

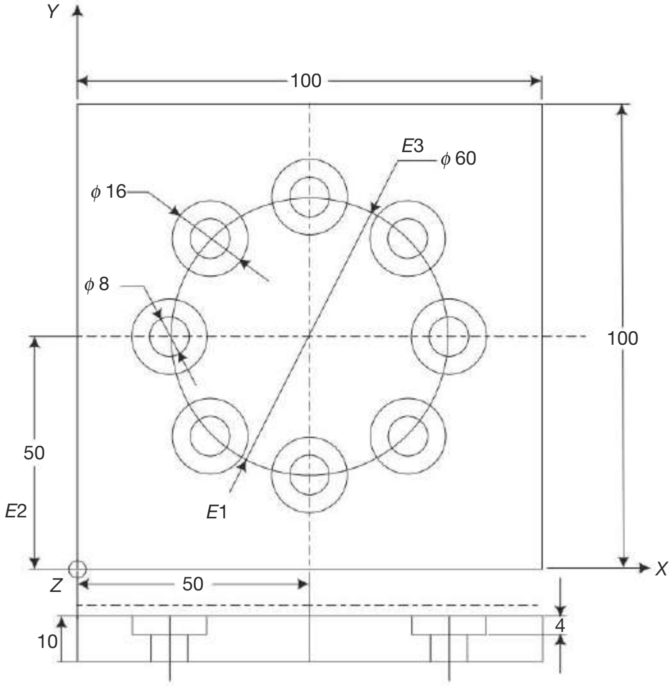


Fig. 15.1 *MAHO subroutine*

Subroutine

```
%MM
N9102;                                              (Subroutine number)
N05 E11=360.0/E6;                                   (angle between points)
N10 E12 = 0;                                        (Counter for drilled hole)
N20 E14 = E1;
N30 E15 = E2;
N40 E12 = E12 + 1;                                  (Increment counter)
N50 E14 = E14 + 0.5 * E3 * COS (E11 * E12);         (X value)
N60 E15 = E15 + 0.5 * E3 * SIN (E11 * E12);         (Y value)
N70 G79 X = E14 Y = E15 Z0;                         (Drill hole)
```

```
N80 E13 = E6 - E12;
N90 G29 E13 N=40;                        (Repeat loop for all holes)
N95 G80;
```

Main program

```
%PM
N8019;
…..
N30 M06 T01;
N40 E1 = 50 E2 = 50 E3 = 60 E6 = 8;      (Set variables)
N50 G81 Y2.0 Z-13.0 F120;                (Define canned cycle)
N60 G22 N=9102;                          (Call subroutine for drill)
N70 M06 T02;
N80 G81 X2.0 Y2.0 Z-4.0 F120;            (Define canned cycle for counter bore)
N90 G22 N=9102;                          (Call subroutine for counter boring)
….
```

15.4.2 Subroutines and Macros in Fanuc

In Fanuc controls two types of subprograms are used.

- Subroutines which are temporary in nature and would be specific to a particular main program.
- Macros that are permanent in nature and therefore will be permanently stored in the controller memory and can be used by any program.

Subroutine The format to be used for writing the subroutines is as follows which is very similar to that of the main programs, with the exception that it should end by M99 and not M02 or M30. The M99 should be in a block by itself.

```
O8234;             Subprogram identification
N010….. ;          Program blocks
N020 …. ;
N030 …. ;
N040 …. ;
N050 M99 ;         Return to calling program
```

Subprograms can be activated by giving a call block (M98) in the main program which must have the following format:

```
N090 M98 Prrrnnnn;
```

where M98 is used to call a subroutine.

P is used to specify the subprogram number as well as the number of times it is to be repeated.

"rrr" specifies the number of times the subprogram is to be repeated. It is possible to repeat a subroutine up to 999 times. If no value is entered, the subprogram is called once.

"nnnn" specifies number of the subroutine to be executed.

A few examples are given below to clarify the usage.

```
N090    M98    P0028023;    (Subroutine 8023 is to be repeated twice)
N140    M98    P8142;       (Subroutine 8142 is to be repeated once)
```

Permanent Macro Permanent macro in Fanuc are meant for storing permanently in the controller memory. These can be called by any program. For this purpose G65 code is used for calling a macro in the main program. The following is the procedure as implemented in Fanuc 0.

```
N055    G65    P2012    L01    A...    B...    C... ;
```

Arguments are assigned using the word addresses in the main program where as they have an equivalent variable number in the sub program as follows.

A	B	C	D	E	F	H	I	J	K	M
#1	#2	#3	#7	#8	#9	#11	#4	#5	#6	#13
Q	R	S	T	U	V	W	X	Y	Z	
#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	

In the main program

```
G65    P1021    A12.5    B23.4    C15.6    D10    X43.5
```

In the sub program

```
 #1 = 12.5
 #2 = 23.4
 #3 = 15.6
 #4 = 10
#24 = 43.5
```

Program number of a custom macro can vary from 0001 to 8999. A few typical examples are presented below.

Example 15.1

```
%
O1243;
....
....
N050 G65 P9082 R2.0 X2.0 Z-12.0;
... ;
... ;
N120 M02;
O9082;
N010 G00 Z #18;
N020 G01 Z #26;
N030 G04 X #24;
N040 G00 Z -[ ROUND[#18] + ROUND[#26] ];
N050 M99;
```

Example 15.2

```
%
O2431;
....
....
N120 G65 P9300 X25.0 Y42.0 Z-12.0;
... ;
... ;
N160 M02;
O9300;
N9100 #1=#5001;                    (Read end of X axis position)
N9110 #2=#5002;
N9120 #3=#5003;
N9130 G00 X#24 Y#25;
N9140 G04;
N9150 X#24 Y#25 Z#26;
N9160 X#1 Y#2;
N9170 Z#3;
N9180 M99;
```

Example 15.3

This example allows for drilling any number of holes along a circle (Fig. 15.2) which are all equi-distant. The inputs are the following.

#1 = *X* coordinate of the centre of the circle
#2 = *Y* coordinate of the centre of the circle
#3 = Diameter of the circle
#5 = Total number of points
#8 = Height of the rapid plane in *Z* coordinate
#4 = Total depth through which the hole making operation is to be done.
#6 = Actual canned cycle code to be used (81, 82, 84, 85, 86).
#9 = The feed rate to be used

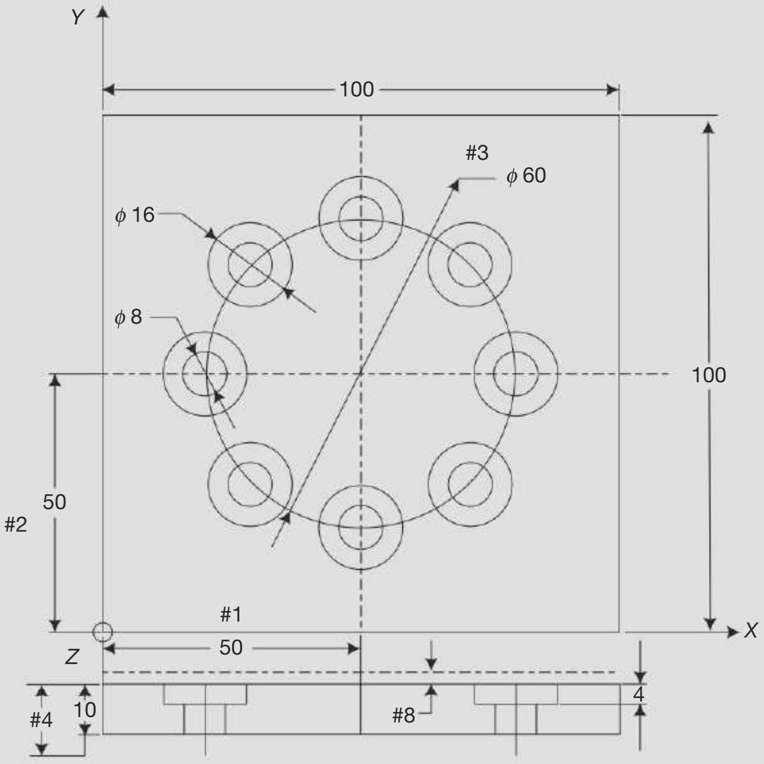


Fig. 15.2 *Fanuc subroutine*

Subroutine

```
%
O2431;                                                (Subroutine number)
N05 #34=360.0/#5;                                     (Angle between points)
N10 #31 = 0;                                          (Counter for drilled hole)
N20 #32 = #1;
N30 #33 = #2;
N40 #31 = #31 + 1;                                    (Increment counter)
N50 #32 = #32 + 0.5 * #3 * COS (#34 * #31); (X value)
N60 #33 = #33 + 0.5 * #3 * SIN (#34 * #31); (Y value)
N70 G#6 X#32 Y#33 Z-#4 R#8 F#9;                       (Drill hole)
N80 IF [ #31 LT #5 ] GOTO 50;                         (Repeat loop for all holes)
N90 G80;
N95 M99;
```

Main program

```
%PM
N8019;
…..
N30 M06 T01;
N50 G65 P2431 A50.0 B50.0 C80.0 E2.0 I13.0 J8 K81 F120;
                                                       (Call subroutine for drill)
N70 M06 T02;
N80 G65 P2431 A50.0 B50.0 C80.0 E2.0 I4.0 J8 K82 F100;
                                                (Call canned cycle for counter bore)
….
```

15.4.3 Subroutines in Sinumeric

The format to be used for writing the subroutines is as follows which is very similar to that of the main programs, with the exception that it should end by M17 and not M02 or M30. The subroutine number is identified by the L word address and can range from 001 to 999. The last two digits should be zero as shown in the example below.

```
L82300;             Subprogram identification, 823
N010….. ;           Program blocks
N020 …. ;
N040 …. ;
N050 M17;           Return to calling program
```

Subprograms can be activated by giving a call using the L word address in the main program with the following format.

N090 Lnnnrr;

where

“nnn” specifies the number of subroutines to be executed.

"rr" specifies the number of times the subprogram is to be repeated. It is possible to repeat a subroutine up to 99 times. If 00 is entered, the subprogram is called once. A few examples are given below to clarify the usage.

```
N090        L41200;       (Subroutine 412 is to be repeated once)
N140        L81402;       (Subroutine 814 is to be repeated twice)
```

Subroutines can be nested up to 3 levels in a program.

15.5 MIRROR IMAGING AND SCALING

It is very common to see the part geometries that are generally symmetric in nature. For such symmetric geometries, when the part program is written in common practice, the bulk of the program may be repeating with small changes. Hence, mirror-imaging facility is incorporated in almost all the controllers to exploit such symmetries. The part programmer has to identify that part of the component geometry which becomes the core, by taking careful consideration of the identification of the part datum. Then the part program may be simply repeated by using appropriate mirror imaging codes. Mirroring or reflection can be carried about *X* axis, *Y* axis or about *X* and *Y* axes. The mirroring can be done about the other axes as well and will be similar to what is being done in the *XY* plane.

Mirroring In MAHO, mirroring can be done using the G code 73 as follows.

```
N125    G73    X-1                (Mirroring the X axis)
N135    G73    Y-1                (Mirroring the Y axis)
N145    G73    Z-1                (Mirroring the Z axis)
N155    G73    X-1    Y-1         (Mirroring the X and Y axis)
N165    G73    X1                 (Cancel mirroring the X axis)
N175    G72                       (Cancel mirroring)
```

Mirroring allows for changing the sign of the dimensions of the particular axis whose mirroring is specified. For example if *X*-1 is specified, then all *X* coordinates after that will have their sign reversed. Figure 15.3 shows an example component suitable for use of mirror imaging facility. When using the mirror imaging, it is important to select the coordinate datum suitably. For example in Fig. 15.3, the part is symmetrical about the *X* axis and hence the datum is chosen as the left hand side at the centre of the part as shown. It therefore needs to write the part program for the machining of half the part only and the rest can be simply copied using the mirror imaging (*Y*-1) facility.

The operations to be performed are the following.

1. Milling of the two pockets	Slot drill, 6 mm dia
2. Drilling 20 holes	Twist drill, 3 mm dia
3. Drilling the 5 holes	Twist drill, 6 mm dia

```
N020    M06    T01;                          (Slot drill)
N025    G00    X0     Y0       Z2.0    M03;
N030    G00    X28.0  Y22.0;
N035    G01    Z-1.0  F120;
N040    X52.0;
N045    Y28.0;
N050    X28.0;
```

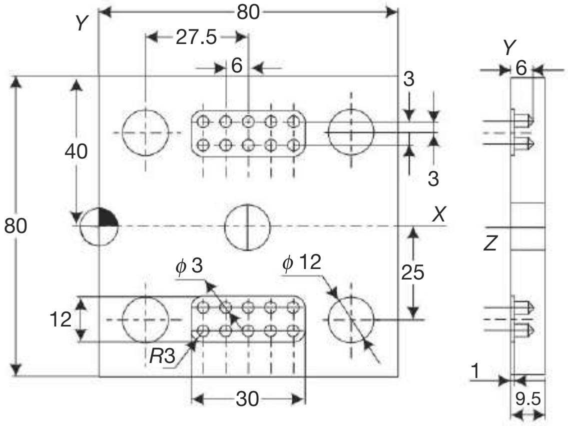


Fig. 15.3 *Example for mirror imaging*

```
N055    Y22.0;
N060    Z2.0;
N065    G00    X0      Y0;
N070    G73    Y-1;                               (Mirror Y axis)
N075    G14    N1=35   N2=65;
N080    G73    Y1 M05;                            (Cancel Y-mirroring)
N085    M06    T02;                               (Twist drill 3 dia)
N090    G81    Y2.0    Z-6.0   F100    M03;
N095    G79    X28.0   Y22.0   Z0;
N100    G79    X34.0;
N105    G79    X40.0;
N110    G79    X46.0;
N115    G79    X52.0;
N120    G79    Y28.0;
N125    G79    X46.0;
N130    G79    X40.0;
N135    G79    X34.0;
N140    G79    X28.0;
N145    G73    Y-1;                               (Mirror Y axis)
N150    G14    N1=95   N2=140;
N155    G72;                                      (Cancel mirroring)
N160    M06    T03;                               (Twist drill 6 dia)
N165    G81    Y2.0    Z-12.0  F100    M03;
N170    G79    X12.5   Y25.0   Z0;
N175    G79    X67.5;
```

```
N180    G79     Y-25.0;
N185    G79     X12.5;
N190    G79     X40.0   Y0      Z0;
N195    G00     X0      Y0      Z50.0   M05;
```

Scaling Scaling allows to resize the part program to any required scale. The size can be increased or decreased depending upon the scale factor supplied. Scaling in MAHO is done by using the G73 only but the scale factor is supplied by the use of the parameter A4. The following is a typical format used.

```
N125    G73     A4=2                  (Scaling by a factor 2)
N135    G73     A4=0.8                (Scaling by a factor 0.8)
N165    G73     A4=1                  (Cancel scaling)
N175    G72                           (Cancel scaling)
```

An example is shown in Fig. 15.54 where the letters MAHO are to be milled using a suitable slot drill to a depth of 1 or 2 mm in a plate. The path shown is for the centre of the tool used. Imagining that the letters are enclosed in a rectangular block as shown, a program is written below to show the use of scaling by a factor of 1.5. To do this, the part program for machining the letters is written as a subroutine in incremental format.

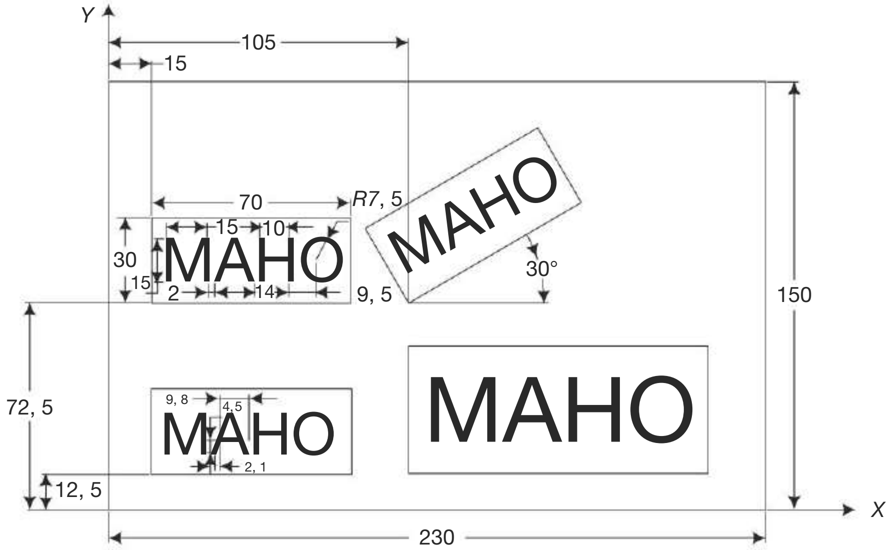


Fig. 15.4 *Scaling*

```
%MM
N9021                          (Subroutine to mill the letters)
N010 G91;
N015 G00 X5.0 Y7.5;
N020 G01 Z-4.0 F120;           (Letter M)
N025 Y15.0;
N030 X7.5 Y-15.0;
```

```
N035 X7.5 Y15.0;
N040 Y-15.0;
N045 G00 Z4.0;
N050 X2.0;
N055 G1 Z-4.0;                 (Letter A)
N060 X7.0 Y15.0;
N065 X7.0 Y-15.0;
N070 G00 Z4.0;
N075 X-11.9 Y4.5;
N080 G01 Z-4.0;
N085 X9.8;
N090 G00 Z4.0;
N095 X4.1 Y-4.5;
N100 G01 Z-4.0;                (Letter H)
N105 Y15.0;
N110 G00 Z4.0;
N115 X10.0;
N120 G01 Z-4.0;
N125 Y-15.0;
N130 G00 Z2.0;
N135 X-10.0 Y7.5;
N140 G01 Z-4.0;
N145 X10.0;
N150 G00 Z4.0;
N155 X2.0;
X160 G01 Z-4.0;                (Letter O)
X165 G02 X15.0 Y0 R7.5;
X170 G02 X-15.0 Y0 R7.5;
X175 G00 Z4.0;
X180 X-57.5 Y-15;
```

Following is the main program that calls the subroutine 9021 suitably to mill all the letters as shown in Fig. 15.4.

```
%PM
….
N045 G90;
N050 G00 X15.0 Y12.5 Z2.0;
N055 G22 N=9021;                       (Mill letters at one position)
N060 G90;
N065 G00 X105.0 Y12.5 Z2.0;
```

```
N070 G73 A4=1.5;                    (Scale factor 1.5)
N075 G22 N=9021;                    (Mill letters at second position)
N080 G73 A4=1;                      (Cancel scaling)
......
```

15.6 SPECIAL CANNED CYCLES

In addition to the normal canned cycles discussed in Chapters 13 and 14, there are a number of special canned cycles provided by the various control manufacturers to allow for specific programming requirements. A few such facilities are presented in the following sections.

15.6.1 Milling Macros (MAHO)

It is often noticed that pocket milling requires a lot of programming to clear all the material present inside the pocket. The pockets can be divided into a number of types such as the following.

- Rectangular
- Circular
- Contour with a number of elements
- Contour with a number of islands

Rectangular Pocket Milling To mill a rectangular pocket, G87 is used in MAHO. The format of G87 cycle definition is as follows (Fig. 15.5).

```
G87 X__Y__Z__ B__R__I__J__K__
```

where

X = width of pocket parallel to X axis
Y = length of pocket parallel to Y axis
Z = total depth of the pocket
B = rapid plane distance from the top of the workpiece
R = corner radius, when it is greater than the radius of the cutter
I = step over distance of cut as percentage of tool diameter
J = to specify type of milling, 1 = Down milling, –1 = Up milling
K = axial depth of cut for each cut

For example, the pocket in Fig. 15.5 is to be programmed as follows.

```
N120 G87 X70.0 Y40.0 Z10.0 B2.0 I70.0 J-1 K5.0 F120;
```

To call the cycle, the tool is to be positioned at the centre point of the pocket in the rapid plane. For example,

```
N130 G79 X55.0 Y40.0 Z0;
```

Circular Pocket Milling To mill a circular pocket, G89 is used in MAHO. The format of G89 cycle definition is as follows (Fig. 15.6).

```
G89 Z__ B__R__I__J__K__
```

where Z = total depth of the pocket
B = rapid plane distance from the top of the workpiece
R = radius of the pocket

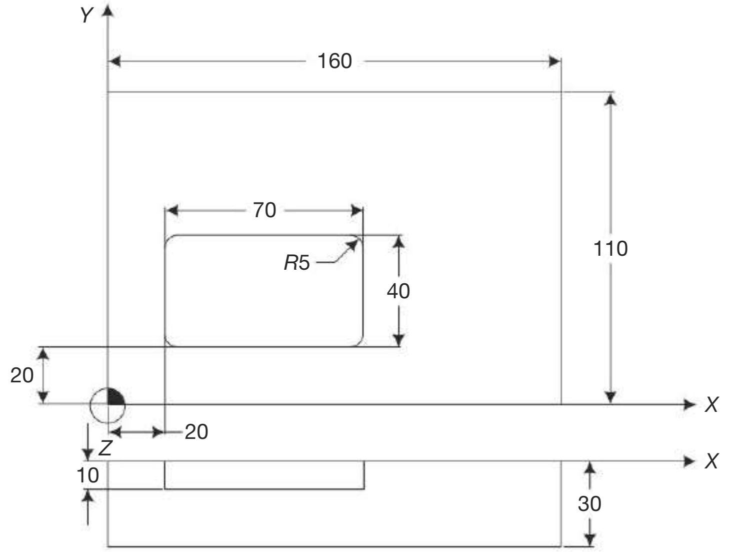


Fig. 15.5 Rectangular pocket milling

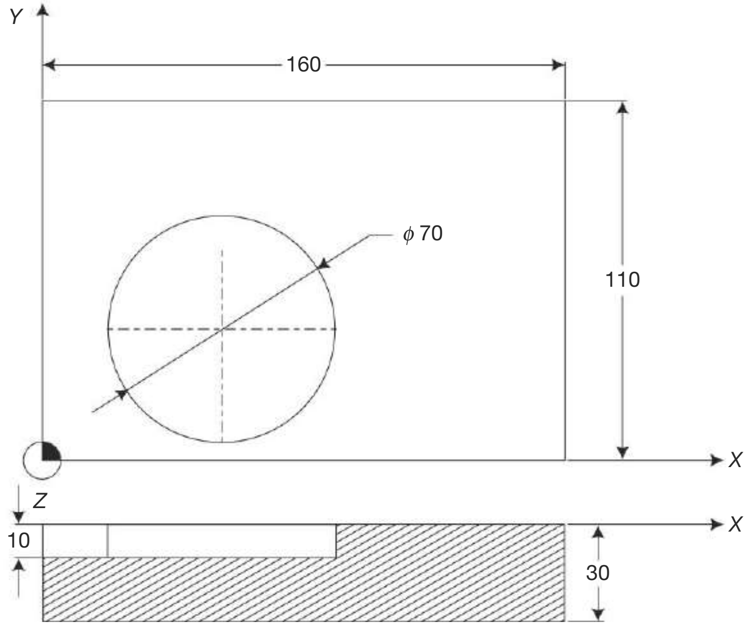


Fig. 15.6 Circular pocket milling

I = step over distance of cut as percentage of tool diameter

J = to specify type of milling, 1 = Down milling, –1 = Up milling

K = axial depth of cut for each cut

For example, the pocket in Fig. 15.6 is to be programmed as follows.

```
N120 G89 Z10.0 B2.0 R35.0 I70.0 J-1 K5.0 F120;
```

To call the cycle, the tool is to be positioned at the centre point of the pocket in the rapid plane. Example is shown here.

```
N130 G79 X55.0 Y40.0 Z0;
```

15.6.2 Turning Macros (Fanuc)

In Chapter 14, it was seen that the program becomes unusually long when a number of cuts are to be planned. Further, each of these cuts are also to be calculated in terms of their end points, which makes the job of a programmer very tedious. Fanuc provides a number of special canned cycles termed as stock removal cycles for such applications. One such cycle is described below for longitudinal turning and boring applications.

Typical stock-removal operation for a pocket in turning application is shown in Fig. 15.7. The tool is positioned at the starting point shown in Fig. 15.7. The finishing contour of the pocket is to be programmed like the normal part programming using the *G* codes. Then using the G71 cycle the pocket will be machined. All the roughing cuts required to turn the pocket will be generated by the controller automatically. The usage of the canned cycle is shown below.

```
N110 G71 U__ R__ ;
N120 G71 P__ Q__ U__ W__ ;
```

The first block establishes the parameters for the rough turning cycle.

where *U* = depth of cut of each pass during the roughing cycle.

R = distance by which the tool will be withdrawn from the part for the return pass.

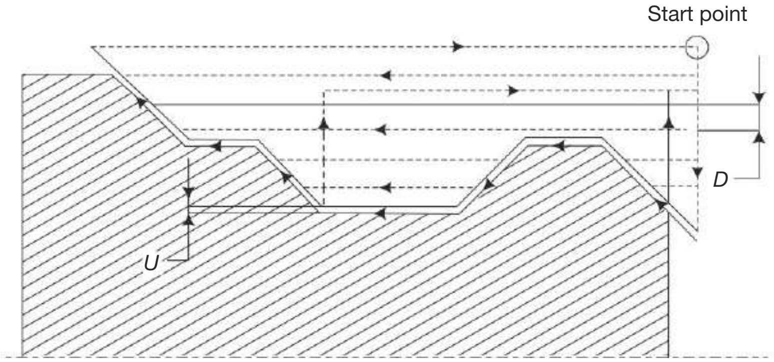


Fig. 15.7 *Fanuc stock-removal cycle for turning/boring*

The second block executes the rough turning cycle.

where

P = first block of the sequence number in the program that controls the workpiece area to be machined

Q = last block of the sequence number in the program that controls the workpiece area to be machined

U = finish allowance to be left for removal during the finish machining cycle along the *X* axis

W = finish allowance to be left for removal during the finish machining cycle along the *Z* axis

The roughing canned cycle G71 leaves the finish allowance on the contour which needs to be removed later using the G70 canned cycle. The format of the G70 canned cycle is

```
N140  G70  P__  Q__;
```

where

P = first block of the sequence number in the program that controls the workpiece area to be machined

Q = last block of the sequence number in the program that controls the workpiece area to be machined

There are a few points to be considered during the use of these stock-removal cycle as given below.

1. A block specified by a P word address should not have a *Z* move.
2. G00 or G01 should be programmed in the block specified by the P word.
3. The contouring path must be a steadily increasing or decreasing pattern in the *Z* axis.
4. A maximum of ten pockets can be programmed in the G71 turning cycle.
5. It is not necessary to program a return to the start point at the end of the program. The control automatically returns the slides to the start point after the block specified by Q is executed.

Given below is a sample program using the above stock-removal cycles to simplify the programming of a component shown in Fig. 15.8.

```
N010 G21
N015 G92 X125.0 Z100.0;
N020 G50 S3500;
N025 M06 T0101;
N030 G97 S300;
N035 G98 F0.2 M13;
N040 G00 X26.0 Z0;
N045 G01 X-1.0;
N050 G00 Z2.0;
N055 X26.0;
N057 G71 U1.0 R0.6;
N060 G71 P065 Q100 U0.5 W0.5 F0.15;
N065 G01 Z0;
N070 X3.0;
N075 G03 X6.0 Z-3.0 R3.0;
N080 G01 Z-22.0;
N085 G02 X9.0 Z-25.0 R3.0;
N090 G03 X12.0 Z-28.0 R3.0;
N095 G01 Z-50.0;
N100 U2.0;
N105 G00 Z2.0;
N110 G70 P065 Q100 F0.3;
N120 G00 X125.0 Z100.0 M05
N130 M30;
```

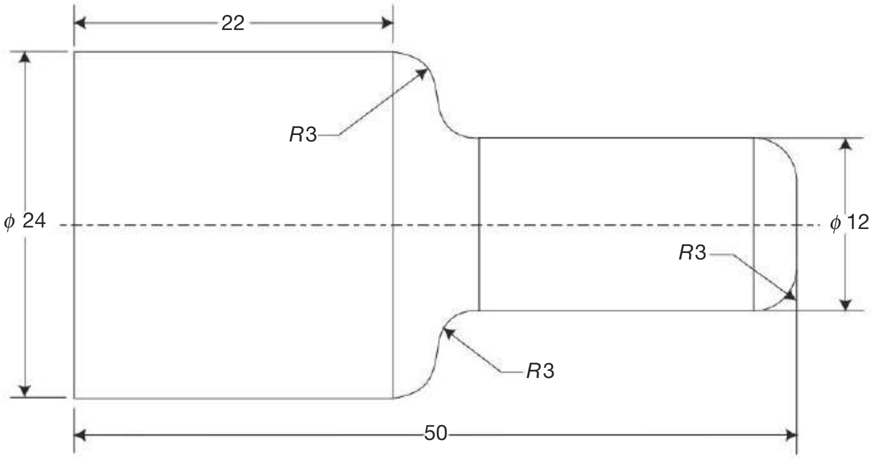


Fig. 15.8 *Exercise for Fanuc stock-removal cycle*

Summary

- Advanced part-programming methods help in reducing the length of program as well as simplify the programming methods. These methods require to examine the logic in developing program where large chunks of programs can be reused by making them as subroutines or special canned cycles. All this is possible because of the high computing capacity of the modern control systems.
- All these facilities are available in all the commonly used controllers such Fanuc and Sinumeric.
- Coordinates of the tool tip can be entered using polar coordinates (radius and angle) in any plane.
- Use of parameters is increasingly encouraged. It is possible to carryout complex arithmetic operations including trigonometric evaluations within a part program.
- It is possible to add unconditional (GOTO) and conditional (IF and DO) loops in a part program, thereby adding a lot of intelligence in the program. Couple this with para-meters it is possible to write a single part program for a group of similar parts.
- Subroutines and macros are powerful ways in which the code segments that required to be used a number of times in a program can be stored.
- Symmetric part geometries can be easily programmed by the use of geometric transformations such as scaling and mirroring.
- Special canned cycles are used to automatically generate part programs for special geometries that are often found.

Questions

1. Give the details of any one advanced function that facilitates the CNC part programming.
2. What type of looping facilities are available in CNC machine tools of the modern generation?
3. What are the normal functions (applications) served by a macro or a sub programme in a NC part program with parameters? Explain with the help of an example for machining an elliptical solid (given the major axis along X axis and minor axis along Y axis). Program only a finish cut along the contour. What modifications need to be done if the major axis is to be oriented along the Z axis?
4. Develop a macro for machining multiple grooves (groove width equals the width of the tool) in a CNC turning centre.
5. Develop a macro for machining any hole-making operation along a straight line in a CNC machining centre.
6. Develop a macro for machining any hole-making operation along a parallelogram grid formed by two straight lines in a CNC machining centre.
7. Develop a macro for milling an elliptical cavity in a solid block in a CNC machining centre.
8. Develop a macro for machining wide grooves (groove width greater than the width of the tool) in a CNC turning centre.

16

COMPUTER-AIDED PART PROGRAMMING

Objectives

Part-programming methods discussed so far are suitable for relatively simple parts involving a small number of features. However, for very complex shapes involving three-dimensional motion, preparing a part program purely by manual means may mean a lot of calculations and the associated time spent in developing the program. All this may lead to inaccuracies and complications within the program. Computer-aided part-programming methods are therefore used for complex shapes that are difficult or impossible to be programmed using manual methods. After completing the study of this chapter, the reader should be able to

- Understand the concept of computer-aided part programming as a universal programming system used for a large range of machine tools
- Appreciate the structure of APT, the first computer-aided part-programming system developed
- Learn the various geometric facilities that are available in APT language and methods to use them
- Use the concept of surface and prepare part programs using the various motion commands available in APT
- Write a complete APT program using the various commands such as postprocessor, compilation control and looping
- Understand Mastercam as the modern computer-aided part-programming system that is more user-friendly and graphics-oriented, unlike APT
- Learn the methods utilised for entering geometric information into Mastercam
- Understand the various tool-path generation modules within Mastercam through which the CNC part programs can be generated
- Write programs for machining centres as well as turning centres using Mastercam

16.1 CONCEPT OF CAP

Preparing part programs for CNC machine tools manually is a viable system for any kind of job. But the assistance of the computer is desirable for part programming because of a variety of reasons. The first and foremost in this respect is the complexity of the workpiece, which makes manual part programming almost impossible. Close tolerance contouring to some mathematically defined, or through a set of points other than a circular arc, would be an example requiring too many coordinate calculations making manual part programming too tedious to be practicable. Thus, simple repetitive and complex manual calculations involved in part programming are taken care of by the computer, leaving the part programmer to attend to other functions to make a better part program.

The reliability of the part program is enhanced as the computer makes all the calculations and thus the part programmer is less likely to make any errors. Besides, the computer has facilities for some error-detection to assist the part programmer in producing a better part program. The input language to the computer is a universal language akin to English and identical for all types of machine-tool controllers. The part programmer is not burdened with having to learn the idiosyncrasies and specific coding requirements of each of the machine tools, enabling him to handle the diverse array of machines and controls with ease. The part-programming time is considerably reduced by as much as 75 per cent, depending on the complexity of the job; in particular for components having repeated geometry in various locations. The computer, in addition to generating the valid NC codes, is also able to provide such additional useful information such as a plot of the cutter path, total machining time for a program, reducing the tape proving costs.

The APT (Automatically Programmed Tools) language system originated at the Servomechanism laboratory of the Massachusetts Institute of Technology, as did the first NC machine tool in 1952. This was the pilot study sponsored by the US Air Marshal Command, which resulted in the prototype system being released for the whirlwind computer in 1955. Though this version was an important step towards the computer preparation of tapes, the user still had to calculate the endpoints of each straight line cut to be performed by the machine tool. MIT, under the sponsorship of the Aerospace Industries Association, then released APT II for IBM 7040 wherein the complete job of part-program preparation from the part drawing was undertaken by the computer. This version was continually developed until 1961 when the APT Long Range Program (ALRP) was created by the AIA and the job of keeping APT up-to-date was given to the Illinois Institute of Technology Research Institute, Chicago. In recognition of the role played by the computer in manufacture, over and above the simple guiding of the cutter tool along the workpiece, the original sponsors have changed the ALRP in 1969 to Computer Aided Manufacture International or CAM-I. Now the work of CAM-I is done by IITRI as well as a large number of contractors all over the globe.

The APT NC reference language consists of a specially structured set of vocabulary, symbols, rules and conventions which are easily understood by the part programmer and help him in faster preparation of control tapes. The vocabulary, which forms the mainstay of the reference language, is a carefully selected set of mnemonics chosen for their similarity in form and meaning with English. The computer translation of

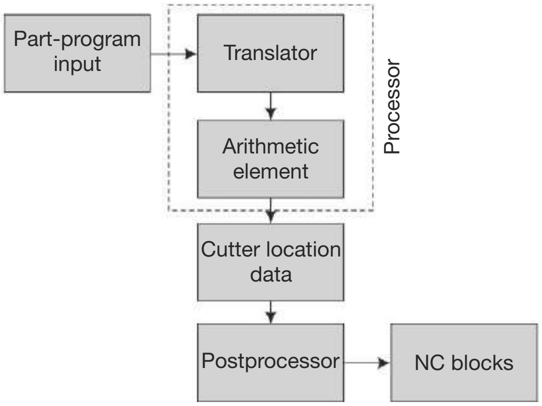


Fig. 16.1 *Computer-aided part-programming system (APT) configuration*

these English-like statements to the valid NC codes for any particular machine-tool controller is generally carried out in two stages as shown in Fig. 16.1.

The first phase involves the conversion of input information into a generalised set of Cutter Location (CL) data and the relevant machine motions. At this stage, the output generated is the universally applicable cutter-centre coordinates (called CLDATA, CLFILE, CLTAPE) which are independent of the machine tool on which the part is to be finally made. This set of programs is called *processor*, and only one such processor is sufficient for any number of NC machine tools in the shop. The output of the processor contains information regarding the feed rates, spindle speeds, directions, coolant status, tool selection and other pieces of information which are machine tool/control unit-oriented, in addition to the cutter-centre coordinates with respect to the workpiece.

The second set of programs are called *postprocessor*, which convert the generalised cutter location data into the specific control codes of the machine tool. As a result, the postprocessor is no more general like the processor but one each is needed for every machine tool/control system combination. The postprocessor converts the cutter-location data along with the machine-tool-oriented information into the appropriate NC codes employed or the particular machine tool/control system combination in question.

The two-pass preparation detailed above is most commonly used. The prime need for this is for making the part-programming system more flexible. By taking out all the machine-tool control-unit-oriented information and making it a separate module, which would be far smaller compared to the main tape-preparation system, one is able to achieve the desired generality. Since the machine-tool-oriented information is embedded in the postprocessor, which happens to be a much smaller segment in the overall tape-preparation system, it is far more economical to duplicate for the various other machine tools which one may be willing to operate.

The various functions that can be attributed to the postprocessor are

1. Converting the CLDATA to the machine-tool coordinate system
2. Converting the CLDATA to the control unit understandable NC blocks taking care of the following machine tool functions:
    - Maximum table or spindle traverses,
    - Available feeds and speeds,
    - Available preparatory, miscellaneous and other functions,
    - Straight line and circular interpolations,
    - Acceleration and decelerations of slides taking care of the overshoot of corners, and
    - Other machine-tool control-unit system requirements such as tape reader time, servo setting time, etc.
3. Provide output
    - Required control tape,
    - Diagnostic listing on line printer, and
    - Other operator/programmer instructions.

The modification of the main processor, which is very complex, will generally not be possible except by the people who had originally written it. Thus, it would be expedient for the users to develop expertise only in the writing or maintenance of the postprocessor, which is far simpler and requires much less manpower. Postprocessors are easier to write or modify. Further, if the same part is to be made on two or more different machines, then the first pass, i.e., the geometrical processing is to be done only once for creating the CLFILE which would be used to postprocess for all the machines concerned, thus saving the geometrical processing time.

A careful examination of its functions reveals the fact that though the postprocessor has to be written for a particular machine-tool control unit, there are many things which are common to all postprocessors. For example, the input and output sections are almost the same for all postprocessors because of the greater standardisation that has been achieved in these areas. Therefore, it seems that by modularising such common sections, it is possible to reduce a large amount of postprocessor writing time. Further standardisation is done by grouping classes of machine tools such as lathes, 2-axis machining centres, drills, punching machines, etc., which have a large amount of commonality between them. Thus, it is desirable to write an imaginary postprocessor for a class of machine tools, which has all features possible in these machines, termed as the universal postprocessor.

16.2 APT LANGUAGE STRUCTURE

The APT language consists of many different types of statements made up of the following valid letters, numerals and punctuation marks.

Letters	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Numerals	0123456789

Punctuation Marks

/	A slash divides a statement into two sections. To the left of the slash are the MAJOR words, and to the right are the words, symbols and/or scalars that modify the word on the left of the slash so as to give it a complete and precise meaning or definition, e.g., GO/PAST, LN, TO, CS.
,	A comma is used as a separator between the elements in a statement generally to the right of the slash.
=	An equals is used for assigning an entity to a symbolic name, e.g., CI = CIRCLE/25, 50, 30.
)	A closing parenthesis is used as a statement label separator.
()	The parentheses are used for enclosing the nested statements.
$	A single dollar sign when placed at the end of a line in the part program indicates that the statement continues in the following line.
$$	The double-dollar sign permits the programmer to make notations and maintain brief records at convenient locations in the part program without interfering in the processing of the part program. The APT processor does not act upon any strings contained to the right of a double-dollar sign. As such, it is preferable to sprinkle as many comments as possible in the part program so as to make it more readable.

A blank character has no meaning in the part program except in strings. In all other cases, the processor ignores a blank. Thus, a blank could be employed profitably to improve the readability of a part program.

Words The words to be used in the statements are built up from one to six letters or numerals with the first one being a letter. No special character is allowed in the words.

Keywords There are certain reserved names called keywords in the language, which have a fixed meaning. These words cannot be used for any other purpose. A keyword may be replaced by another name using a SYN statement. All keywords consist of between two and six letters, without any numerals. The keywords are divided into two classes, the MAJOR keywords, which define the type of the statement, and the MINOR keywords, which give the required parameters and modifiers,

e.g., CIRCLE
MATRIX

LEFT
TANTO

Symbols Symbols are the words used as substitutes for geometrical definitions and numerical values, where the first character must be a letter. A symbol must be defined before it is referenced in a subsequent part-program statement,

e.g., L4
LIN32
CIRC23

Labels Label names are used to reference a statement so that control can be transferred to that statement changing the usual linear-execution sequence. Labels are identical to the words with the difference that all the characters in a label can be numerals. A label must be terminated by a right parenthesis.

Numbers Numbers have their usual meaning as in algebra and are often referred to as scalars. If a number is unsigned, the positive sign is assumed. No distinction is made between integer and real numbers. The APT processor treats all scalars as real numbers. The following are valid numbers:

–37624
259.0
0.145

The maximum digits in a number should be 15 excluding the sign.

The structure of statements used in an APT part program is shown in Fig. 16.2.

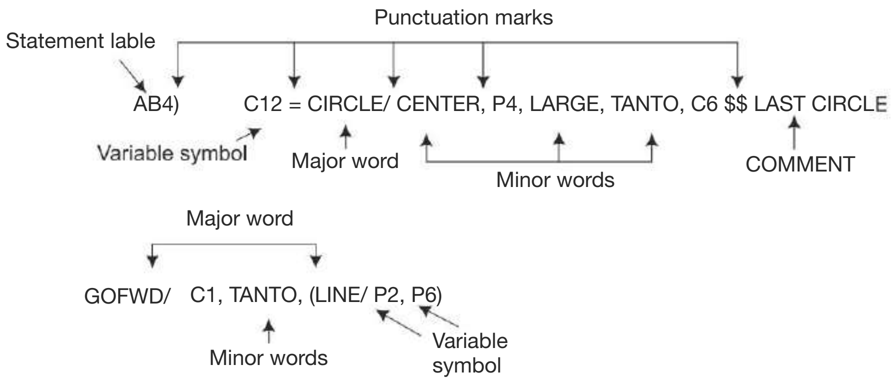


Fig. 16.2 *Structure of statements in APT*

Angles Angles are quoted as degrees and decimal fractions of a degree, e.g., 13°42’ would be 13.70. Angles have a positive value when measured in an anticlockwise direction from the positive *X*-axis.

Arithmetic APT provides the facility for arithmetic computations in the part program. The operators allowed are

Addition +
Subtraction –
Multiplication *

Division / and

Exponentiation **

The character '=' is used in arithmetic statements but may not have the same meaning as in algebra. In part-program arithmetic it should be read as 'is replaced by' or 'is given the value of'. As a result, we can write $B = B + 4$, which means the variable B is assigned the value of $B + 4$. The implied multiplication feature in algebra is not allowed,

e.g., $A = 5(4 + 2)$ should be written as $A = 5 * (4 + 2)$

$B = 2^3$ should be written as $B = 2 ** 3$

$C = \frac{1 + 2 + 5}{4}$ should be written as $C = (1 + 2 + 5)/4$

Normal algebraic priorities would apply.

Priority	
1	( )
2	**
3	* , /
4.	+, –

To facilitate computations, the language provides for some additional library functions in additions to the operators just described. The most commonly used library functions are

ABS	Absolute value
SQRT	Square root
SIN	Sine of the angle in degrees
COS	Cosine of the angle in degrees
TAN	Tangent of the angle in degrees
ASIN	Angle of the sine in degrees
ACOS	Angle of the cosine in degrees
ATAN	Angle of the tangent in degrees
EXP	Exponential, the value of the *e* to the power
LOG	Natural logarithm

The usage is similar to all the functions with the argument enclosed in parentheses,

e.g., A = SQRT (14.2 + 36 * 12)

B = EXP (A * 34 – 32 * ABS (C – D))

Any valid arithmetic expression is permissible as an argument in parentheses for the function subject to some specific restrictions related to the functions. For example, in the case of SQRT, the argument should be non-negative. Similarly, the value of an argument in the case of ASIN and ACOS should be less than ± 1.

The complete APT part program consists of the following four types of statements:

- Geometry
- Motion
- Postprocessor
- Compilation control

16.3 GEOMETRY COMMANDS

There are many ways in which the part geometry in APT could be specified. The part geometry is normally broken into a number of surface elements that could be defined from the data given in a part print. These are POINT, LINE, CIRCLE, PLANE, VECTOR, PATERN, SPHERE, GCONIC, TABCYL, etc. For each of the types of surface that can be defined, a number of alternative ways are possible for definition to simplify the definition procedure. For each of the definitions, the general form is presented in which the following rules are observed:

(i) a symbol in lower-case letters underlined represents the surface type specified,
(ii) a symbol in lower-case letters without underline represents a scalar,
(iii) the words shown in the upper-case letters are the MINOR keywords, and
(iv) all possible modifiers are presented one below the other enclosed by an opening brace from among which only one appropriate for the definition is to be selected.

A few examples are shown below:

Nesting The geometrical definitions or arithmetic computations can be nested, wherever necessary. A nested definition is a definition enclosed in parentheses and then inserted in another statement, e.g.,

```
PL2 = PLANE/ P1, P2, P3
ZSURF/ PL2
```

The above two statements could be combined into a single statement as

```
ZSURF/ (PL2 = PLANE/ P1, P2, P3)
```

Similarly, arithmetic computations if necessary during the geometric definitions or in any other statement can also be nested, e.g.,

```
P2 = POINT/ (25 + 12 * COS(35)), (12 – 6 * SIN (35))
```

The above statement without any nesting could have been written as

```
X1 = 25 + 12 * COS(35)
Y1 = 12 – 6 * SIN(35)
P2 = POINT/ X1, Y1
```

Too much of nesting in a single statement would make it unnecessarily long and is likely to be error-prone, and as such it is not advisable to indulge in unnecessary nesting. At the same time, it is desirable to leave all the arithmetic manipulations to be done by the APT processor in the interest of accuracy.

ZSURF ZSURF specifies a plane which is to be used to provide a Z value for every subsequent point definition in which no Z value has been specified. ZSURF may be redefined within a part program. Where ZSURF is not specified, all point definitions, which have no Z value specified, are given a Z value equal to zero. The general usage of the definition is shown above along with the examples for nesting.

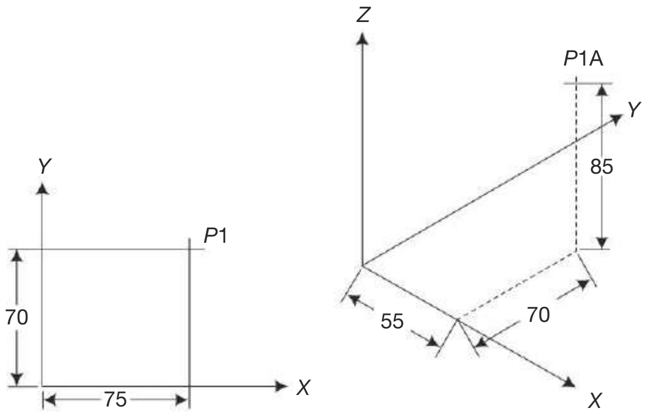
16.3.1 Point

The point has three coordinates along X, Y and Z-axes. The Z coordinate when not specified is taken as either zero or the prevailing Z surface definition.

By Rectangular Coordinates (Fig. 16.3)

General syntax is <symbol> = POINT/ x, y, z

```
P1 = POINT/75.0, 70.0
P1A = POINT/55.0, 70.0, 85.0
```



(a) (b)

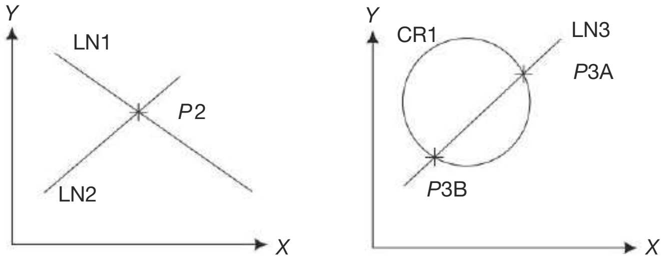
Fig. 16.3 *Point definitions*

By the Intersection of Two Lines (Fig. 16.4a)

General syntax is <symbol> = POINT/ INTOF, line1, line2

Keyword INTOF refers to the intersection

P2 = POINT/ INTOF, LN1, LN2



(a) (b)

Fig. 16.4 *Point definitions*

By the Intersection of Line and Circle (Fig. 16.4b)

General syntax is

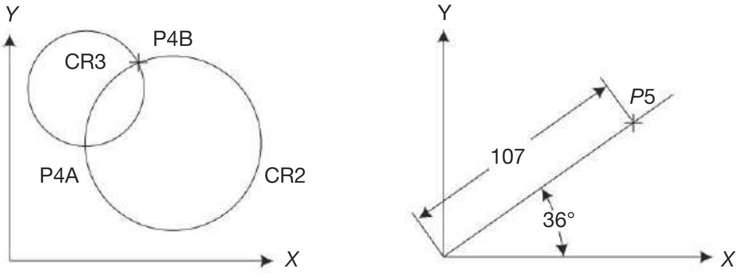
$$\text{<symbol>} = \text{POINT/}\left\{\begin{matrix}\text{XSMALL}\\ \text{XLARGE}\\ \text{YSMALL}\\ \text{YLARGE}\end{matrix}\right\}\text{, INTOF, line1, circle1}$$

In the definition, the line symbol is specified first followed by the circle symbol.

The modifiers XSMALL, XLARGE, etc., one of which is to be used, signify the point, which has an algebraically small or large coordinate when projected onto that axis.

P3B = POINT/ XSMALL, INTOF, LN3, CR1

P3A = POINT/ XLARGE, INTOF, LN3, CR1



(a) (b)

Fig. 16.5 *Point definitions*

By the Intersection of Two Circles (Fig. 16.5a)

General syntax is <symbol> = POINT/ INTOF, circle1, circle2

P4A = POINT/ XSMALL, INTOF, CR2, CR3

P4B = POINT/ XLARGE, INTOF, CR2, CR3

Polar Coordinates in a Coordinate Plane (Fig. 16.5b)

General syntax

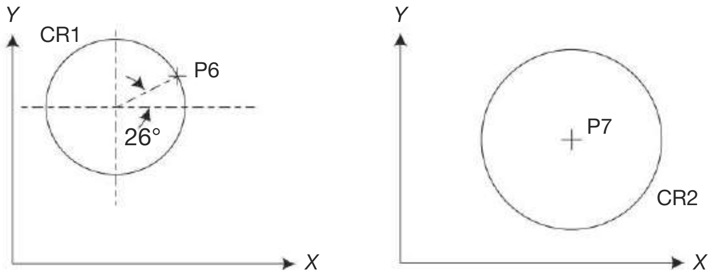
$$\text{<symbol> = POINT/ RTHETA, } \begin{Bmatrix} XYPLAN \\ YZPLAN \\ ZXPLAN \end{Bmatrix} \text{, radius, angle}$$

$$\text{<symbol> = POINT/ THETAR, } \begin{Bmatrix} XYPLAN \\ YZPLAN \\ ZXPLAN \end{Bmatrix} \text{, angle, radius}$$

The radius must not be a negative value.

The modifiers XYPLAN, etc., specify the plane in which the point is lying.

P5 = POINT/ RTHETA, XYPLAN, 107, 36



(a) (b)

Fig. 16.6 *Point definitions*

On a Circle at an Angle with the X-axis (Fig. 16.6a)

General syntax is <symbol> = POINT/ circle1, ATANGL, angle1

P7 = POINT/CR1, ATANGL, 26

By the Centre of a Circle (Fig. 16.6b)

General syntax is <symbol> = POINT/ CENTER, circle1

P6 = POINT/CENTER, CR2

16.3.2 Line

Lines are considered to be of infinite length and do not have a direction. Lines must not be perpendicular to the *XY* plane. Lines are considered as planes perpendicular to the *XY* plane.

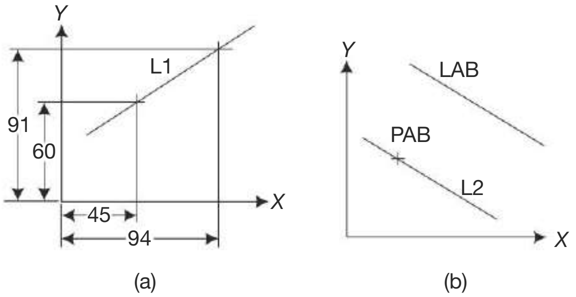


Fig. 16.7 *Line definitions*

By Two Point Symbols or Rectangular Coordinates of Points (Fig. 16.7a)

General syntax is <symbol> = LINE/ x1, y1, z1, x2, y2, z2

<symbol> = LINE/ point1, point2

The two points should not be the same or they should not have the same coordinates.

L1 = LINE/ 45, 60, 94, 91

By a Point and a Parallel Line (Fig. 16.7b)

General syntax is <symbol> = LINE/ point1, PARLEL, line2

L2 = LINE/ PAB, PARLEL, LAB

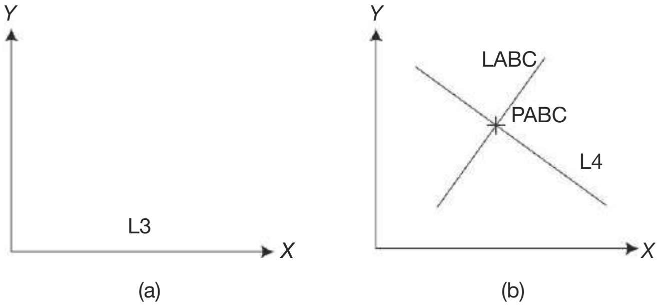


Fig. 16.8 *Line definitions*

As One of the Coordinate Axis (Fig. 16.8a)

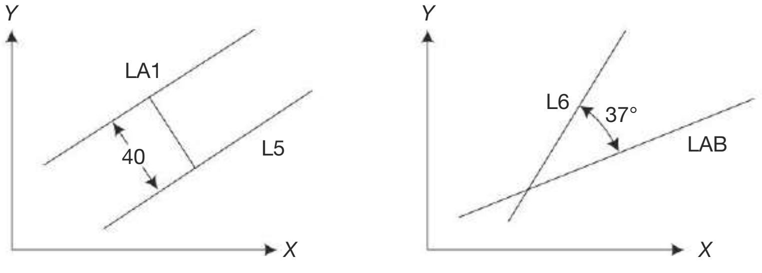
General syntax is $\quad$ <symbol> = LINE/ $\left\{\begin{matrix} XAXIS \\ YAXIS \end{matrix}\right\}$

L3 = LINE/ XAXIS

By a Point and a Perpendicular Line (Fig. 16.8b)

General syntax is $\quad$ <symbol> = LINE/ point1, PERPTO, line2

L4 = LINE/ PABC, PERPTO, LABC



(a) $\qquad$ (b)

Fig. 16.9 *Line definitions*

By a Parallel Line at a Distance (Fig. 16.9a)

General syntax is $\quad$ <symbol> = LINE/ PARLEL, line2, $\left(\begin{matrix} XSMALL \\ XLARGE \\ YSMALL \\ YLARGE \end{matrix}\right)$, distance

L5 = LINE/ PARLEL, LA1, YSMALL, 40

By a Point and Angle made with a Line (Fig. 16.9b)

General syntax is $\quad$ <symbol> = LINE/ point, ATANGL, angle1, $\left\{\begin{matrix} \text{line 1} \\ XAXIS \\ YAXIS \end{matrix}\right\}$

L6 = LINE/ PNT1, ATANGL, 37, LAB

When the line is not specified, XAXIS is assumed.

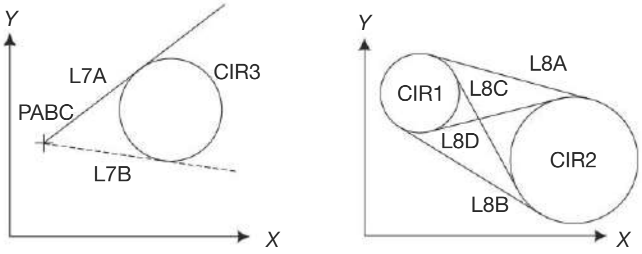
By a Point and a Tangential Circle (Fig. 16.10a)

General syntax is $\quad$ <symbol> = LINE/ point, $\left\{\begin{matrix} \text{LEFT} \\ \text{RIGHT} \end{matrix}\right\}$, TANTO, circle1

RIGHT and LEFT is established looking from the point towards the centre of the circle. The point must not be inside the circle.

L7A = LINE/ PABC, LEFT, TANTO, CIR3

L7B = LINE/ PABC, RIGHT, TANTO, CIR3



(a) (b)

Fig. 16.10 *Line definitions*

Tangential to Two Circles (Fig. 16.10b)

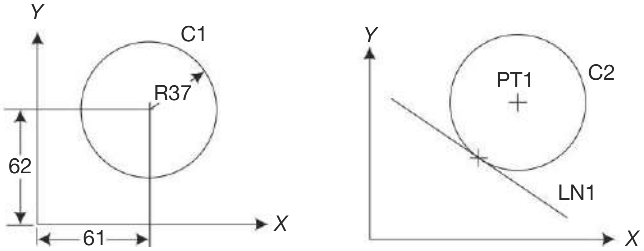
General syntax is $\text{<symbol>} = \text{LINE/}\begin{Bmatrix} LEFT \\ RIGHT \end{Bmatrix}, \text{TANTO, circle1, } \begin{Bmatrix} LEFT \\ RIGHT \end{Bmatrix}, \text{TANTO, circle 2}$

RIGHT and LEFT is established looking from the centre of the first circle specified in the definition towards the centre of the other circle. One circle must not be completely inside the other circle.

L8A = LINE/ LEFT, TANTO, CIR1, LEFT, TANTO, CIR2
L8B = LINE/ RIGHT, TANTO, CIR1, RIGHT, TANTO, CIR2
L8C = LINE/ LEFT, TANTO, CIR1, RIGHT, TANTO, CIR2
L8D = LINE/ RIGHT, TANTO, CIR1, LEFT, TANTO, CIR2

16.3.3 Circle

A circle is always considered as a circular cylinder perpendicular to the *XY* plane of infinite length. The radius value when specified must not be negative.



(a) (b)

Fig. 16.11 *Circle definitions*

By the Centre and Radius (Fig. 16.11a)

General syntax is <symbol> = CIRCLE/ x1, y1, radius1
<symbol> = CIRCLE/ x1, y1, z1, radius1
<symbol> = CIRCLE/ CENTER, point1, RADIUS, radius1

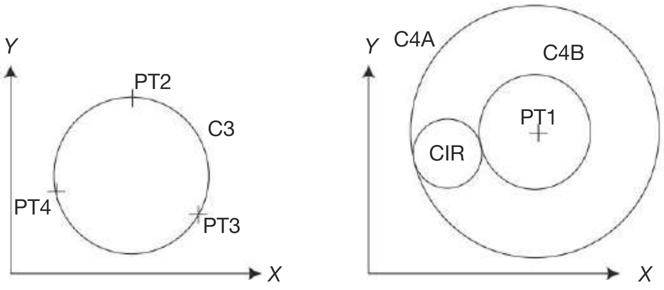
The centre can be specified by a symbol or by its coordinate values.

C1 = CIRCLE/ 61, 62, 37

By the Centre and Tangential Line (Fig. 16.11b)

General syntax is <symbol> = CIRCLE/ CENTER, point1, TANTO, line1

C2 = CIRCLE/ CENTER, PT1, TANTO, LN1



(a) (b)

Fig. 16.12 *Circle definitions*

By Three Points on the Circumferences (Fig. 16.12a)

General syntax is <symbol> = CIRCLE/ point1, point2, point3

No two points should coincide. The three points must not be collinear.

C3 = CIRCLE/ PT4, PT2, PT3

By the Centre and a Tangential Circle (Fig. 16.12b)

General syntax is

$$\text{<symbol> = CIRCLE/ CENTER, point1, } \left\{\begin{matrix} LARGE \\ SMALL \end{matrix}\right\}\text{, TANTO, circle1}$$

The point should not coincide with the centre of the given circle.

C4A = CIRCLE/ CENTER, PT1, LARGE, TANTO, CIR

C4B = CIRCLE/ CENTER, PT1, SMALL, TANTO, CIR

By Two Tangential Lines and Radius (Fig. 16.13a)

General syntax is

$$\text{<symbol> = CIRCLE/ } \left\{\begin{matrix} XSMALL \\ XLARGE \\ YSMALL \\ YLARGE \end{matrix}\right\}\text{, line1, } \left\{\begin{matrix} XSMALL \\ XLARGE \\ YSMALL \\ YLARGE \end{matrix}\right\}\text{, line2, RADIUS, radius1}$$

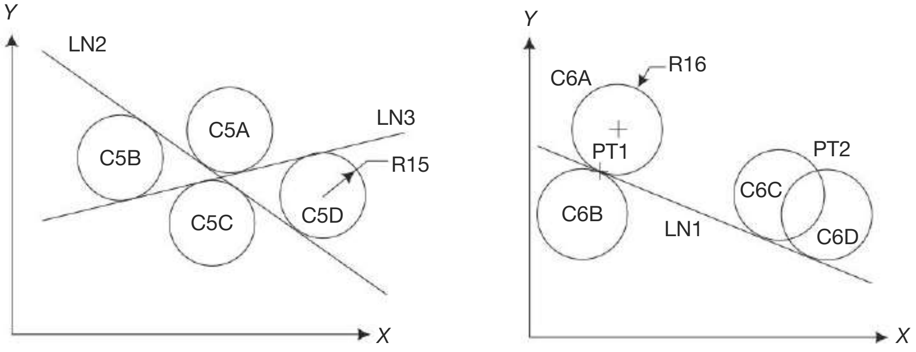
These modifiers specify the choice of the circle whose centre has algebraically larger or smaller X or Y coordinate with reference to the line. The two lines must not be parallel.

C5A = CIRCLE/ YLARGE, LN2, YLARGE, LN3, RADIUS, 15

C5B = CIRCLE/ XSMALL, LN2, YLARGE, LN3, RADIUS, 15

C5C = CIRCLE/ YSMALL, LN2, YSMALL, LN3, RADIUS, 15

C5D = CIRCLE/ YSMALL, LN3, XLARGE, LN2, RADIUS, 15



(a) (b)

Fig. 16.13 *Circle definitions*

By a Tangential Line, a Point on the Circumference and Radius (Fig. 16.13b)

General syntax is

$$\text{<symbol>} = \text{CIRCLE/TANTO, line1, } \left\{\begin{matrix} XSMALL \\ XLARGE \\ YSMALL \\ YLARGE \end{matrix}\right\}, \text{ point1, RADIUS, radius1}$$

These modifiers specify the choice of the circle whose centre has algebraically larger or smaller X or Y coordinate with reference to the point.

C6A = CIRCLE/ TANTO, LN1, YLARGE, PT1, RADIUS, 16
C6B = CIRCLE/ TANTO, LN1, YSMALL, PT1, RADIUS, 16
C6C = CIRCLE/ TANTO, LN1, XSMALL, PT2, RADIUS, 16
C6D = CIRCLE/ TANTO, LN1, XLARGE, PT2, RADIUS, 16

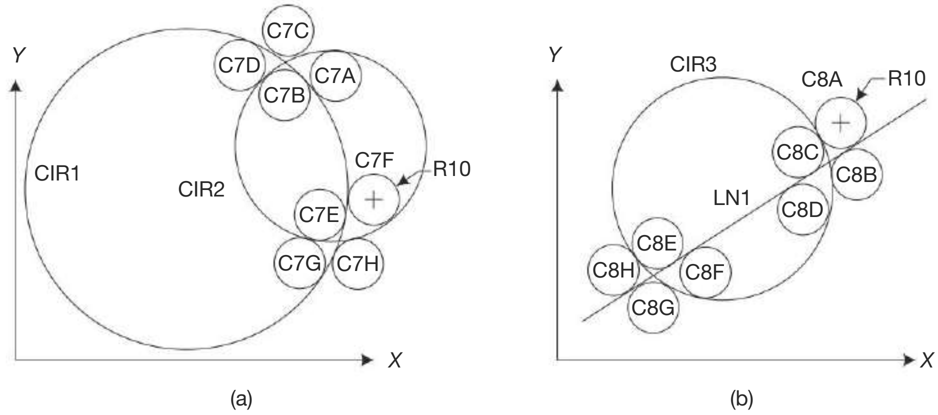
By Two Tangential Circles and Radius (Fig. 16.14a)

General syntax is

$$\text{<symbol>} = \text{CIRCLE/} \left\{\begin{matrix} XSMALL \\ XLARGE \\ YSMALL \\ YLARGE \end{matrix}\right\}, \left\{\begin{matrix} \text{IN} \\ \text{OUT} \end{matrix}\right\}, \text{ circ1, } \left\{\begin{matrix} \text{IN} \\ \text{OUT} \end{matrix}\right\}, \text{ circ2, RADIUS, rad1}$$

IN and OUT allows the selection of the considered circle by indicating the mode of tangency between the two circles.

C7A = CIRCLE/ YLARGE, OUT, CIR1, IN, CIR2, RADIUS, 10
C7B = CIRCLE/ YLARGE, IN, CIR1, IN, CIR2, RADIUS, 10
C7C = CIRCLE/ YLARGE, OUT, CIR1, OUT, CIR2, RADIUS, 10
C7D = CIRCLE/ YLARGE, IN, CIR1, OUT, CIR2, RADIUS, 10
C7E = CIRCLE/ YSMALL, IN, CIR1, IN, CIR2, RADIUS, 10
C7F = CIRCLE/ YSMALL, OUT, CIR1, IN, CIR2, RADIUS, 10
C7G = CIRCLE/ YSMALL, IN, CIR1, OUT, CIR2, RADIUS, 10
C7H = CIRCLE/ YSMALL, OUT, CIR1, OUT, CIR2, RADIUS, 10



(a) (b)

Fig. 16.14 *Circle definitions*

By a Tangential Line, a Tangential Circle and Radius (Fig. 16.14b)

General syntax is

$$\text{<symbol>} = \text{CIRCLE/}\left\{\begin{matrix} XSMALL \\ XLARGE \\ YSMALL \\ YLARGE \end{matrix}\right\}, \text{line1}, \left\{\begin{matrix} XSMALL \\ XLARGE \\ YSMALL \\ YLARGE \end{matrix}\right\}, \left\{\begin{matrix} IN \\ OUT \end{matrix}\right\}, \text{circ2, RADIUS, radius1}$$

C8A = CIRCLE/ YLARGE, LN1, XLARGE, OUT, CIR3, RADIUS, 10
C8B = CIRCLE/ YSMALL, LN1, XLARGE, OUT, CIR3, RADIUS, 10
C8C = CIRCLE/ YLARGE, LN1, XLARGE, IN, CIR3, RADIUS, 10
C8D = CIRCLE/ YSMALL, LN1, XLARGE, IN, CIR3, RADIUS, 10
C8E = CIRCLE/ YLARGE, LN1, XSMALL, IN, CIR3, RADIUS, 10
C8F = CIRCLE/ YSMALL, LN1, XSMALL, IN, CIR3, RADIUS, 10
C8H = CIRCLE/ YLARGE, LN1, XSMALL, OUT, CIR3, RADIUS, 10
C8G = CIRCLE/ YSMALL, LN1, XSMALL, OUT, CIR3, RADIUS, 10

16.3.4 Vector

Vectors have both magnitude and direction. They are often used to specify the direction in geometry application as well as the motion of the cutter.

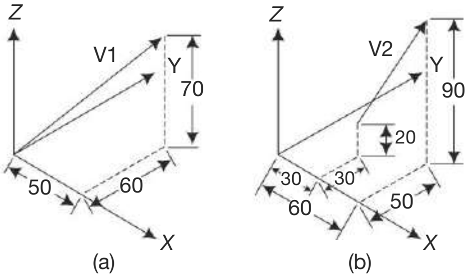
Syntax is <symbol> = VECTOR/ <parameters>

By x, y, z Components (Fig. 16.15a)

Syntax is <symbol> = VECTOR/ x1, y1, z1

V1 = VECTOR/ 50, 60, 70

This definition generates a vector from the origin of the coordinate system.



(a) (b)

Fig. 16.15 *Vector definitions*

By Two Points (Fig. 16.15b)

Syntax is <symbol> = VECTOR/ x1, y1, z1, x2, y2, z2

<symbol> = VECTOR/ point1, point2

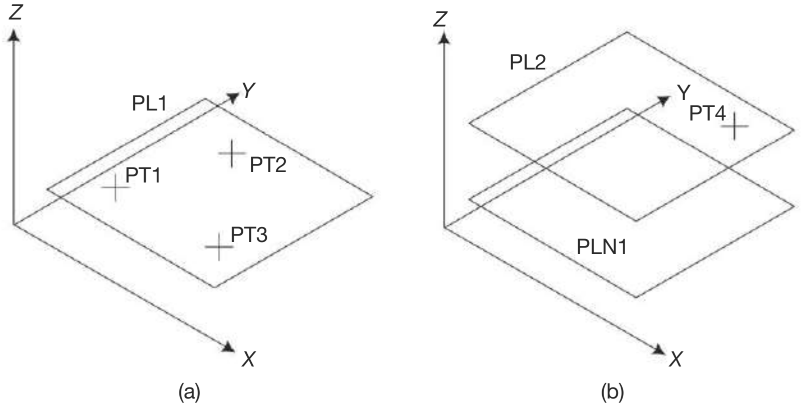
V1 = VECTOR/ 30, 30, 0, 60, 50, 90

When a vector is defined by the coordinate values, all the components need to be mentioned without omission.

16.3.5 Plane

Planes are surfaces with infinite areas. These are often used to specify the machining surfaces and tool-end surface location.

Syntax is <symbol> = PLANE/ <parameters>



(a) (b)

Fig. 16.16 *Plane definitions*

By Three Points (Fig. 16.16a)

Syntax is <symbol> = PLANE/ point1, point2, point3

PL1 = PLANE/ PT1, PT2, PT3

By a Point and a Parallel Plane (Fig. 16.16b)

Syntax is <symbol> = PLANE/ point1, PARLEL, plane1

PL2 = PLANE/ PT4, PARLEL, PLN1

By the Coefficient of a Plane Equation aX + bY + cZ = d (Fig. 16.17)

Syntax is <symbol> = PLANE/ a, b, c, d

PL3 = PLANE/ 0, 0, 0, 27

PL3A = PLANE/ 0, 0, 0, 49

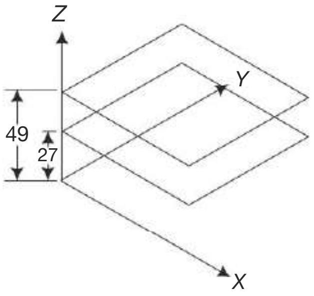


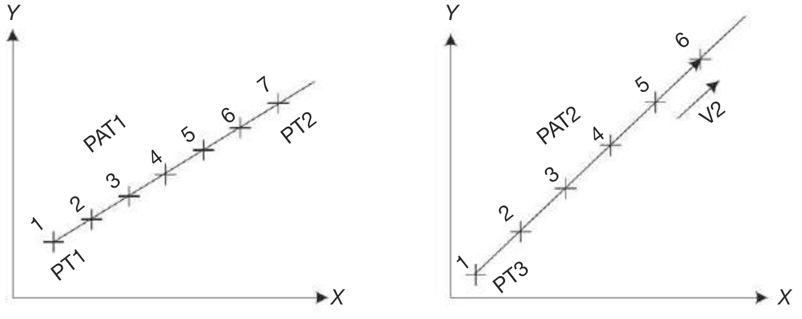
Fig. 16.17 *Plane definitions*

16.3.6 Pattern

Patterns are used for the purpose of grouping of holes that need identical processing. These definitions rely on some form of symmetry, which is present in the group.

Syntax is <symbol> = PATERN/ <parameters>

The *Z* value of any point used in the pattern definition is ignored. The *Z* levels associated with a pattern are determined by a CYCLE command during the motion statement by the postprocessor.



(a) (b)

Fig. 16.18 *Pattern definitions*

Linear Pattern by the First, Last and Number of Points (Fig. 16.18a)

Syntax is <symbol> = PATERN/ LINEAR, point1, point2, number

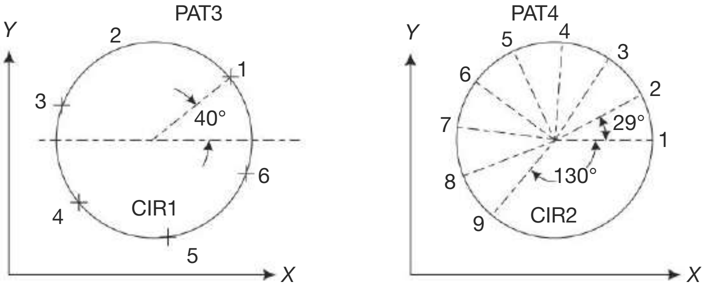
PAT1 = PATERN/ LINEAR, PT1, PT2, 7

Numbering of the holes in the patterns starts from the start point as shown in the example.

Linear Pattern by the First Point, Vector and Number of Points (Fig. 16.18b)

Syntax is <symbol> = PATERN/ LINEAR, point1, vector1, number

PAT1 = PATERN/ LINEAR, PT3, V2, 6



(a) (b)

Fig. 16.19 *Pattern definitions*

Circular Pattern by a Circle, Angular Positions of First Hole, Last Hole and Number of Points (Fig. 16.19a)

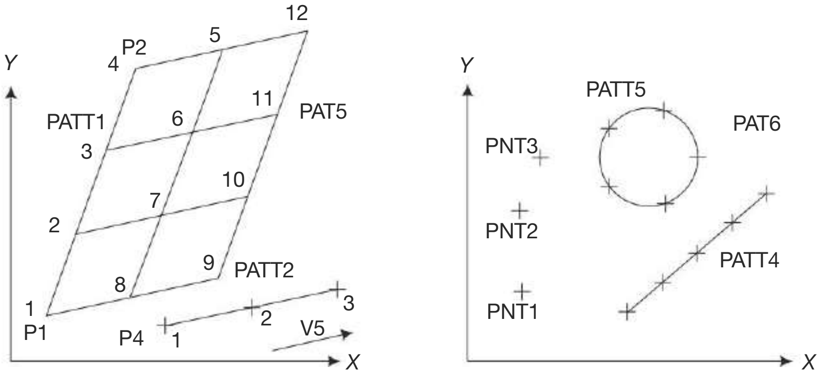
Syntax is <symbol> = PATERN/ ARC, circle1, ang1, ang2, $\left\{\begin{matrix}\text{CLW}\\\text{CCLW}\end{matrix}\right\}$, number

If the position of the last hole, ang2 is not specified then it would be treated as the first hole position, ang1. Angles are measured from the radius parallel to the positive *X*-axis direction.

CLW The pattern is produced moving in a clockwise direction from the starting angle given.

CCLW Same as above in the counter-clockwise direction.

PAT3 = PATERN/ ARC, CIR1, 40, CCLW, 6
PAT4 = PATERN/ ARC, CIR2, 29, -130, CCLW, 9



(a) (b)

Fig. 16.20 *Pattern definitions*

Parallelogram Constructed From Two Patterns (Fig. 16.20a)

Syntax is <symbol> = PATERN/ PARLEL, patern1, patern2

PATT1 = PATERN/ LINEAR, P1, P2, 4
PATT2 = PATERN/ LINEAR, P4, V5, 3
PAT5 = PATERN/ PARLEL, PATT1, PATT2

The sides of the parallel pattern are formed by moving the second pattern quoted, so that its origin coincides with the origin of the first pattern given in the definition. The sequence of the point numbering is as shown in the example.

Random Strings of Points and Patterns (Fig. 16.20b)

Syntax is <symbol> = PATERN/ RANDOM, point1, patern1, point2, patern2

PAT6 = PATERN/ RANDOM, PATT4, PATT5, PNT1, PNT2, PNT3

16.3.7 Matrix

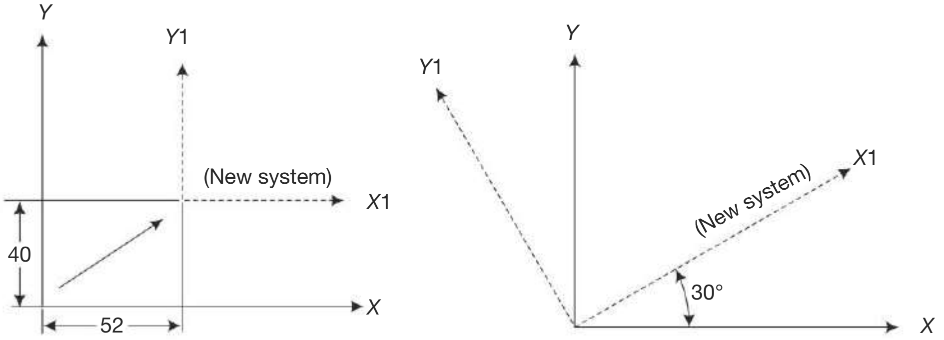
Matrix definitions are useful for the transformations of the cutter coordinates from one system to another. The advantage is that the geometry could be defined from a convenient coordinate system and the same could be transformed into any of the required coordinate frames.

A matrix is a set of twelve parameters. These parameters are the coefficients of three linear equations defining the mathematical relationship between the two coordinate systems. Nine of the coefficients are the directional cosines and the other three are the origin of the new coordinate system.

Syntax is <symbol> = MATRIX/ <parameters>

Translation (Fig. 16.21a)

Syntax is <symbol> = MATRIX/ TRANSL, dist1, dist2, dist3



(a) (b)

Fig. 16.21 *Matrix definitions*

dist1, dist2 and dist3 are the coordinates of the origin of the new system with its axes parallel to the old system.

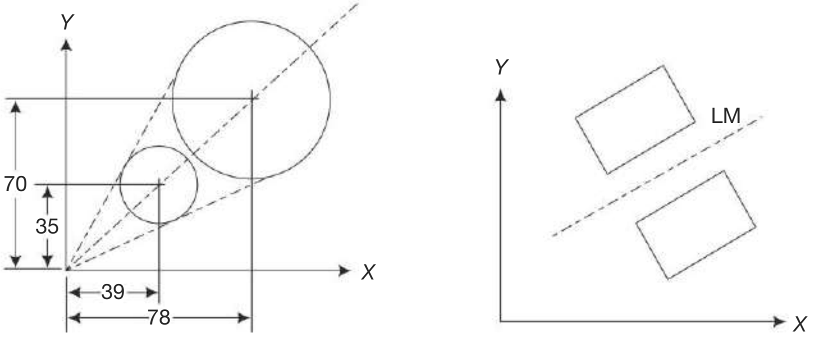
MAT1 = MATRIX/ TRANSL, 40, 52

Rotation (Fig. 16.21b)

Syntax is $$\text{<symbol>} = \text{MATRIX/}\begin{Bmatrix} XYROT \\ YZROT \\ ZXROT \end{Bmatrix}, \text{angle1}$$

The positive sense of the rotation is defined from the first referenced axis to the new one in the counter-clockwise direction.

MAT2 = MATRIX/ XYROT, 30



(a) (b)

Fig. 16.22 *Matrix definitions*

Translation (Fig. 16.22a)

Syntax is <symbol> = MATRIX/ SCALE, factor1

The factor represents the ratio between a unit vector in the old system to the corresponding one in the new system.

MAT3 = MATRIX/ SCALE, 2

Mirroring (Fig. 16.22b)

Syntax is
$$\text{<symbol>} = \text{MATRIX/ MIRROR,} \begin{Bmatrix} XZPLAN \\ XYPLAN \\ YZPLAN \\ \text{line 1} \\ \text{plane 1} \end{Bmatrix}$$

This definition transforms the geometry as if viewed in a mirror against a surface, which could be a line, plane or any of the principal coordinate plane.

MAT4 = MATRIX/ MIRROR, LM

Rotation and Translation (Fig. 16.23)

Syntax is

$$\text{<symbol>} = \text{MATRIX/}\begin{Bmatrix} XYROT \\ YZROT \\ ZXROT \end{Bmatrix}\text{, angle1, TRANSL, x1, y1, z1}$$

This definition first rotates the geometry about the origin of the part coordinate system and then translates the rotated geometry.

MAT2 = MATRIX/ XYROT, 30, TRANSL, 52, 40, 0

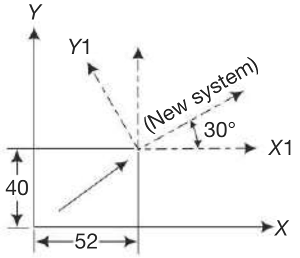


Fig. 16.23 *Matrix definitions*

16.3.8 Geometry Examples

In the following a few examples of APT geometries of components (views shown in XY plane only) have been defined:

Example 16.1 (Fig. 16.24)

```
PARTNO/ EXAMPLE 16.1 FIG. 16.24
P2 = POINT/ 0, 0
L1 = LINE/ 20, 20, 20, (20 + 80)
L2 = LINE/ (POINT/ 20, (20 + 80)), ATANGL, 45
P1 = POINT/ (20 + 30 + 40 + 20), 20
C2 = CIRCLE/ CENTER, P1, RADIUS, 20
L4 = LINE/ P1, PERPTO, (LINE/ XAXIS)
C1 = CIRCLE/ (20 + 30 + 40), (20 + 80 + 30 – 20), 20
L3 = LINE/(POINT/(20 + 30),(20 + 80 + 30)),PARLEL,(LINE/
XAXIS)
```

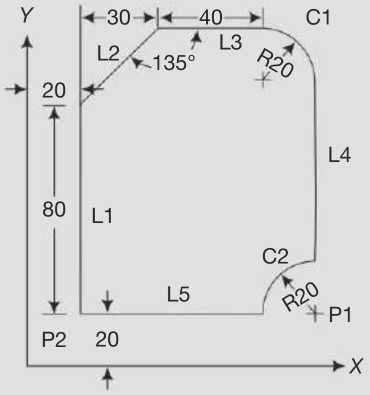


Fig. 16.24 *Example 1 for APT geometry definition*

Example 16.2 (Fig. 16.25)

```
PARTNO/ EXAMPLE 16.2 FIG. 16.25
L4 = LINE / XAXIS
C1 = CIRCLE/ 24, 20, 12.5
L1 = LINE/ (POINT/ 0,0), ATANGL, 15
```

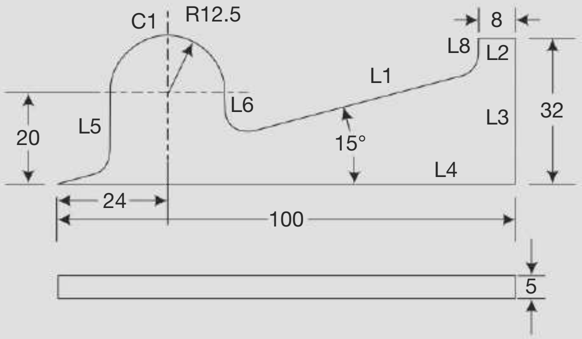


Fig. 16.25 *Example 2 for APT geometry definition*

```
L2 = LINE/ PARLEL, L4, YLARGE, 32
L7 = LINE/ YAXIS
L5 = LINE/ PARLEL, L7, XLARGE, (24 – 12.5)
L6 = LINE/ PARLEL, L5, XLARGE, (12.5 + 12.5)
L3 = LINE/ PARLEL, L7, XLARGE, 100
```

Example 16.3 (Fig. 16.26)

```
PARTNO/ EXAMPLE 16.3 FIG. 16.26
P1 = POINT/ 40, (160 + 40 - 10)
P2 = POINT/ 40, 40
L1 = LINE/ P1, P2
L5 = LINE/ XAXIS
L4 = LINE/ PARLEL, L5, YLARGE,40
L2 = LINE/ PARLEL, L4, YLARGE, 160
L3 = LINE/ (POINT/ (40 + 120), (40 + 160)),
ATANGL, 135
C2 = CIRCLE/ YLARGE, L4, XSMALL, L3,
RADIUS, 20
A = SQRT (40 * 40 – 30 * 30)
C1 = CIRCLE/ (40 + A), 160, 40
```

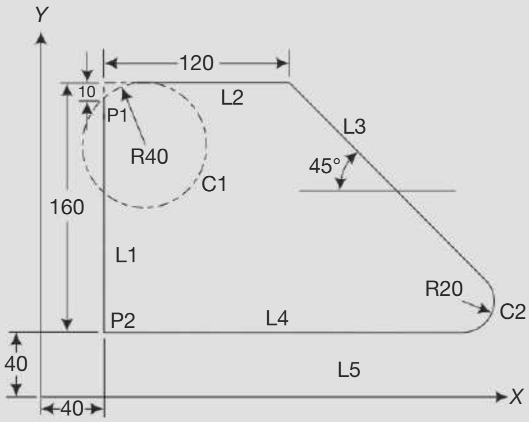


Fig. 16.26 *Example 3 for APT geometry definition*

Example 16.4 (Fig. 16.27)

```
PARTNO/ EXAMPLE 16.4 FIG. 16.27
P1 = POINT/ 65, 65
START = POINT/ 0, 0
C2 = CIRCLE/ 65, (230 – 115), (50/2)
P2 = POINT/ (65 + 65), (230 – 40)
```

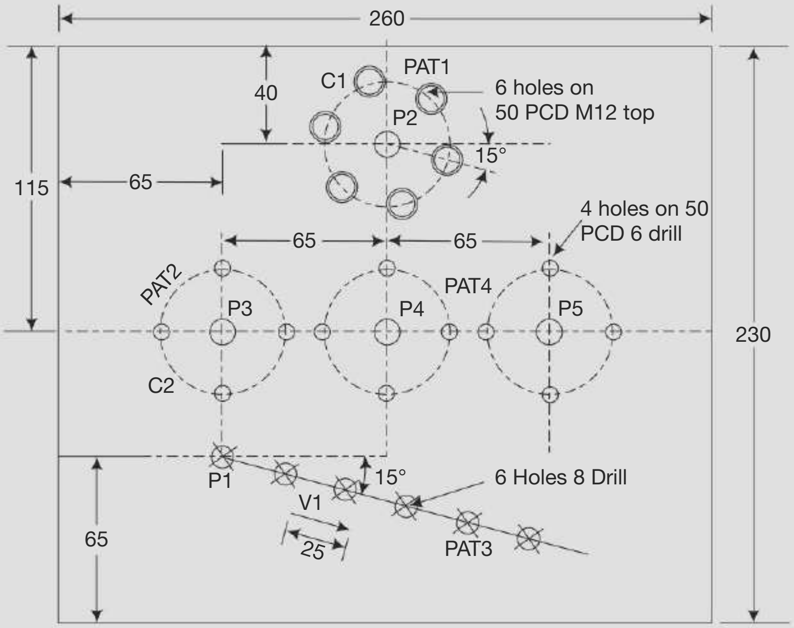


Fig. 16.27 *Example 4 for APT geometry definition*

```
C1 = CIRCLE/ CENTER, P2, RADIUS, (50/2)
Y = 25 * SIN (15)
X = 25 * COS (15)
V1 = VECTOR/ 65, 65, 0, (65 + X), (65 – Y), 0
PAT3 = PATERN/ LINEAR, P1, V1, 6
PAT2 = PATERN/ ARC, C2, 0, CLW, 4
PAT1 = PATERN/ ARC, C1, –15, CCLW, 6
MAT1 = MATRIX/ TRANSL, 65, 0, 0
MAT2 = MATRIX/ TRANSL, (65 + 65), 0, 0
P3 = POINT/ 65, (230 – 115)
```

Example 16.5 (Fig. 16.28)

```
PARTNO/ EXAMPLE 16.5 FIG. 16.28
L4 = LINE/ XAXIS
L5 = LINE/ YAXIS
L3 = LINE/ PARLEL, L5, XLARGE, (20 – 5)
LM = LINE/ PARLEL, L5, XLARGE, (140/2)
C1 = CIRCLE/ (20 + 20), 22.5, 5
C2 = CIRCLE/ 20, (15 + 15), 5
```

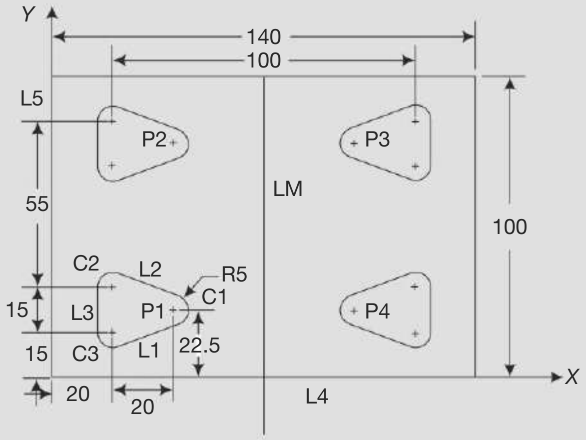


Fig. 16.28 *Example 16.5 for APT geometry definition*

```
C3 = CIRCLE/ 20, 15, 5
L1 = LINE/ LEFT, TANTO, C1, LEFT, TANTO, C3
L2 = LINE/ RIGHT, TANTO, C1, RIGHT, TANTO, C2
L3 = LINE/ PARLEL, L5, XLARGE, (20 – 5)
P1 = POINT/ CIRCLE, C1
P2 = POINT/ (20 + 20), (22.5 + 55)
P3 = POINT/ (20 + 100 – 20), (22.5 + 55)
P4 = POINT/ (20 + 100 – 20), 22.5
PAT= PATERN/ RANDOM, P1, P2, P3, P4
```

16.4 MOTION COMMANDS

This section of commands is more complex compared to the rest of the part program. The main function of these commands is to describe the actual machining sequence making use of the geometry elements defined earlier.

The processor assumes that the tool moves around the workpiece for the purpose of machining. If this were not true for any machine-tool control unit, the postprocessor would take care of the necessary conversions.

The motion commands can be broadly divided into three groups:

- set-up commands,
- point-to-point motion commands, and
- continuous path motion commands.

16.4.1 Set-up Commands

In this category, the types of statements required for initiating the machining operation are now described.

Start Point In general, the position of the tool at the end of a motion is treated as the starting point for the subsequent motion. But to establish the starting point of the cutter when it starts the first motion, the FROM command is to be used. The format is

FROM/ point
FROM/ x1, y1
FROM/ x1, y1, z1

When no *Z* value is specified, it is derived from the current ZSURF in force.

This point is the set point where the tool will be set with reference to the holding fixture or component. The first FROM command does not involve any movement of the tool but simply defines where the tool already is. In a part program, normally only one FROM may appear and that should precede the first motion command. All subsequent FROM commands encountered would be treated as GOTO commands.

Cutter The cutting-tool description is necessary in every part program involving continuous path commands. General format is

CUTTER/ dia
CUTTER/ dia, radius
CUTTER/ dia, radius, e, f, ... , h

Cutter is modal and would be cancelled by another CUTTER statement only. In a part program, any number of CUTTER statements could be present. In the absence of full description of the cutter, it would be treated as cylindrical in calculating the machining path. The points calculated are represented by the *X*, *Y* and *Z* coordinates of the tool endpoints as shown in Fig. 16.29.

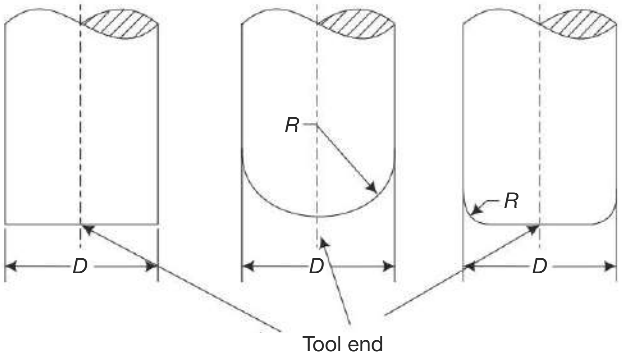


Fig. 16.29 *Typical cutter shapes used for programming contour statements in APT*

Surfaces As mentioned earlier, in APT the tool moves round the workpiece to do the machining. The desired path of the 3-dimensional cutting tool is described by means of three intersecting surfaces, which are designated as drive surface (dsurf), part surface (psurf) and check surface (csurf). For any motion to be described, all three surfaces are to be designated either explicitly or otherwise. The cutting tool is expected to move along the intersection of the drive and part surfaces till it is stopped by means of the check surface as shown in Fig. 16.30.

Part surface is the one which is in continual contact with the tool tip and helps in the control of depth of cut. Drive surface is the other surface with which the cutting tool is in continual contact during a given motion. The tool periphery or tool axis follows the drive surface. This is the surface, which should always be explicitly mentioned in any continuous path motion command. Check surface is the one which limits the given motion statement.

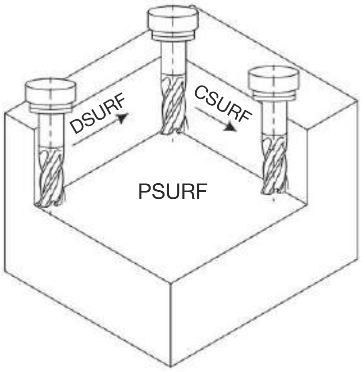


Fig. 16.30 *Surfaces required during a continuous path statement in APT*

Tolerances Though the surfaces could be accurately defined, most CNC machine tools produce curved surfaces by a series of straight-line motions. Hence, it is expedient on the part of the part programmer to specify a tolerance range so that the straight-line motions, called *cut vectors,* generated are close to the required surface. The format of the tolerance specification is

INTOL/ dsval, psval, cs1val, cs2val
OUTTOL/ dsval, psval, cs1val, cs2val
TOLER/ dsval, psval, cs1val, cs2val

In the above definitions, the values refer to the drive surface, part surface, first check surface and second check surface respectively. The last three values are optional. Any value can be omitted from the right side in which case it is assumed as the one given for the previous one. For example, if only dsval is specified, the same tolerance value is used for all the surfaces.

INTOL specifies the amount by which the cutter can cut into the workpiece surface, and OUTTOL for the excess material to be left out as shown in Fig. 16.31.

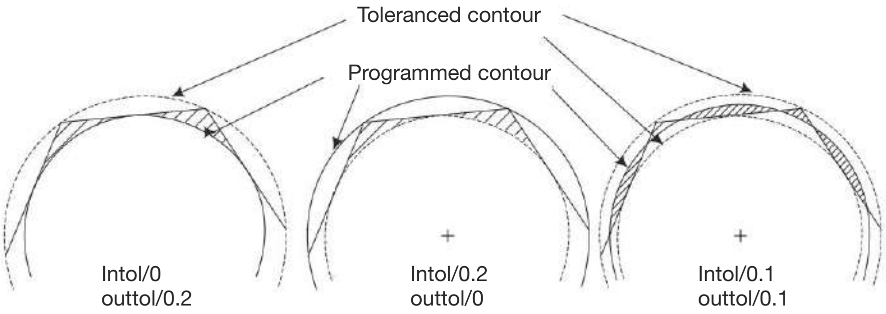


Fig. 16.31 *Tolerance specification for surfaces in APT*

The tolerance specification should be either or both of INTOL and OUTTOL or TOLER. If TOLER is specified, OUTTOL is set to zero. If no tolerance is specified, some default value, which is low, will be assumed by the processor. Tolerance values are modal and can be changed in a part program at any time.

16.4.2 Point-to-Point Motion Commands

These are used to specify the positioning commands used for point-to-point applications such as drilling operation.

GODLTA This statement specifies the relative movement along the axes specified.

GODLTA/ dx, dy, dz
GODLTA/ dz
GODLTA/ vector

This results in moving *dx* along the *X*-axis, *dy* along the *Y*-axis and *dz* along the *Z*-axis from the current position. If only one value is specified, it refers to the *Z*-axis.

GOTO This is an absolute movement statement. It is used to move the spindle from the current position to the point specified in the statement. The general format is

GOTO/ x, y
GOTO/ x, y, z
GOTO/ point

When no *Z*-value is specified, it is taken from ZSURF in force.

The GOTO statement can also be utilised for positioning of a series of points that are already grouped into a pattern in the specified sequence.

GOTO/ pattern

The type of processing to be done at the points visited by the spindle is specified by means of the postprocessor commands.

The part programmer has the ability to change the order of point positions or tool movements by the use of a number of pattern modifiers such as INVERS, OMIT, RETAIN, AVOID, THRU and CONST. The use of these modifiers does not change the order of points in the pattern defined, but only changes the way the tool reaches these points as defined by the GOTO statement.

INVERS	reverses the order in which the tool visits the locations
OMIT	tool will not visit the specified points
RETAIN	tool will visit only the specified points

Normally, when positioning is done in a pattern, the tool moves at rapid rate in a clearance plane, which is close to the workpiece top surface. If any work-holding clamp or other obstruction is present then the tool has to clear them by moving to a higher plane than the clearance plane. This is achieved by using the AVOID modifier. The statement

GOTO/patern, AVOID, 15, 2, 4

causes the tool to be raised 15 units at point 2 and 4 and lowered at points 3 and 5 respectively as shown in Fig. 16.32.

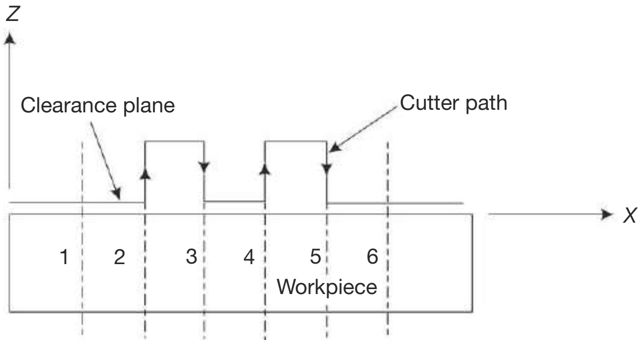


Fig. 16.32 *AVOID example in GOTO/ PATERN statement*

THRU Establishes the sequence of locations rather than specifying all points for modifiers OMIT, RETAIN and AVOID.

Output order of the points is the order in which the GOTO statement specifies the points. This is the same as that of the pattern definition if the INVERS modifier is not used. However, if INVERS is used then the numbering order also reverses. To maintain the numbering order the same as that in the definition the modifier CONST may be used. For example,

GOTO/ patern, INVERS, CONST, OMIT, 9, THRU, 4

provides a sequence as shown in Fig. 16.33.

16.4.3 Continuous-Path Motion Commands

These are used to specify the continuous path motion involving milling and turning operations to generate a variety of surfaces.

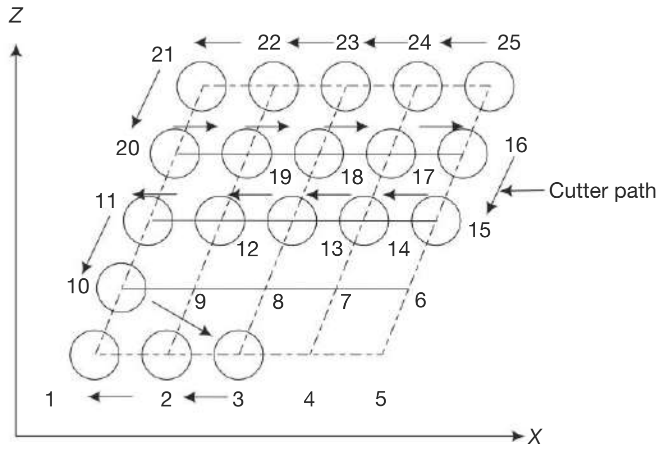


Fig. 16.33 *GOTO PATERN with modifiers*

Part Surface As mentioned earlier, part surface is the one which is in continual contact with the tool tip and helps in control of the depth of cut. It can be explicitly specified by the following commands.

PSIS/ ps	Part surface is
AUTOPS	Automatic part surface (current Z level)
NOPS	No part surface

Start-up Command Before proceeding with the machining of the part, it is necessary to establish the contact of the cutting tool with the part and drive surface within the specified tolerance limits. The general format used for the purpose is

$$\text{GO/}\begin{Bmatrix} TO \\ ON \\ PAST \end{Bmatrix}, \text{ds}, \begin{Bmatrix} TO \\ ON \\ PAST \end{Bmatrix}, \text{ps}, \begin{Bmatrix} TO \\ ON \\ PAST \\ TANTO \end{Bmatrix}, \text{cs}$$

The modifier TO affects the relationship of the tool with reference to the drive, part or check surface, ON, PAST or TANTO used with that particular surface. The relationship is shown in Fig. 16.34. If no modifier is specified, TO is assumed.

When a three-surface start-up is used as given above, the tool will be correctly positioned at the drive surface as shown in Fig. 16.35.

GO/TO, DSURF, TO, PSURF, TO,CSURF

Continuous Motion Commands Having started the continuous path motion, it is necessary to continue the motion using the action verbs to specify the direction in which the tool needs to move. The action verbs available are

GOLFT/ ds, TO, cs	Contour motion command to go left
GORGT/ ds, TO, cs	Contour motion command to go right
GOFWD/ ds, TO, cs	Contour motion command to go forward
GOBACK/ ds, TO, cs	Contour motion command to go back
GOUP/ ds, TO, cs	Contour motion command to go up
GODOWN/ ds, TO, cs	Contour motion command to go down

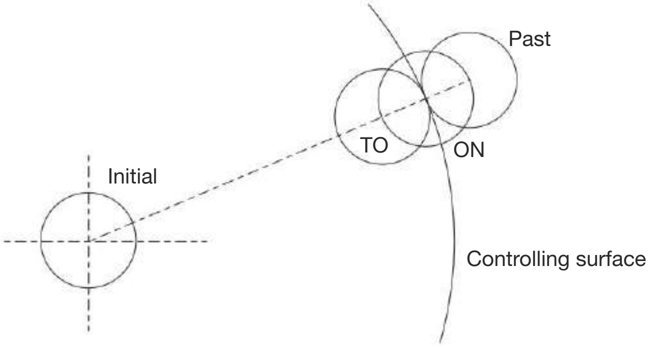


Fig. 16.34 *Modifiers used for motion commands*

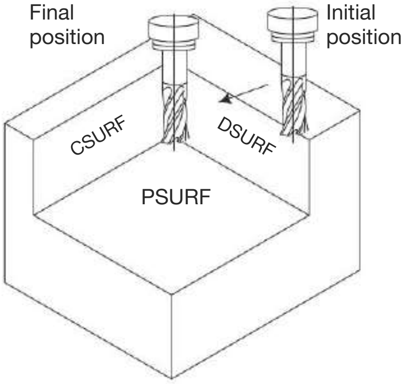


Fig. 16.35 *Startup command for continuous-path motion*

Each of these specifies the direction in which the tool moves with reference to the move as shown in Fig. 16.36. In order to arrive at the directions, which are relative to the previous motion, the part programmer may imagine sitting on the tool and moving along with it.

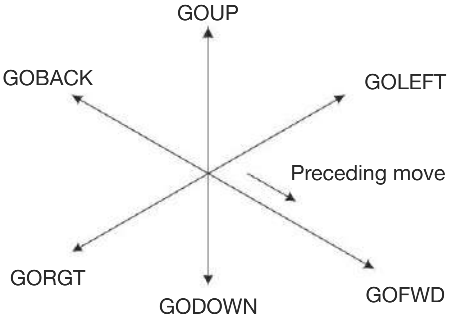


Fig. 16.36 *Action verbs in continuous-path motion*

Though the direction is shown as at right angles in Fig. 13.36, APT allows the action verbs to be valid over a large range (to be precise up to 88° on either side of validity) as shown in Fig. 13.37. In Fig. 13.37, GORGT is valid at any angle at 176° from the forward direction. Similarly, other action verbs also have similar validity. As a result, there is an overlap of 86° where either GOFWD or GORGT is valid. Similarly, there is a large overlap, which makes the life of the part programmer relatively easy to give the action verb.

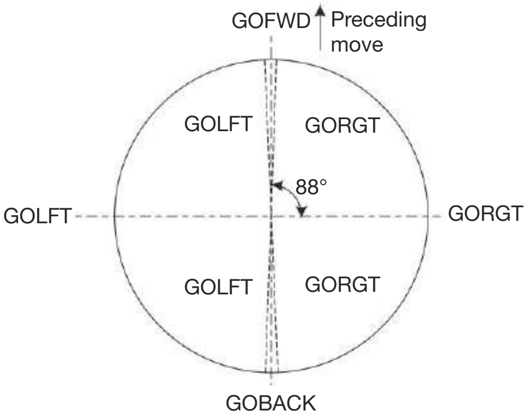


Fig. 16.37 *Validity of the action verbs in motion commands*

Typical applications of the action verbs are given in Fig. 16.38, which are self-explanatory. Typical formats that would be used with action verbs are

$$\text{GOLFT/ ds,} \left\{ \begin{matrix} \mathit{TO} \\ \mathit{ON} \\ \mathit{PAST} \\ \mathit{TANTO} \end{matrix} \right\} \text{, cs}$$

$$\text{GOLFT/ ds,} \left\{ \begin{matrix} \mathit{TO} \\ \mathit{ON} \\ \mathit{PAST} \\ \mathit{TANTO} \end{matrix} \right\} \text{, n, INTOF, cs}$$

Normally, the action verb requires a drive surface and check surface. However, when a continuous motion command is given, the check surface for the present motion becomes the drive surface for the next motion as

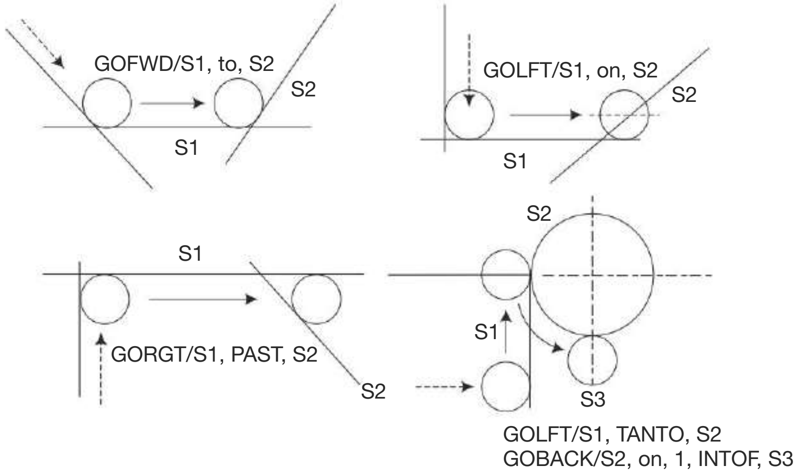


Fig. 16.38 *Typical application of action verbs in motion*

shown in Fig. 16.39. As a result, it is not necessary to specify the check surface explicitly, except the last motion command.

```
FROM/ SETPT
GO/ L1, PS, L4
GOLFT/ L1, PAST, L2
GORGT/ L2, PAST, L3
GORGT/ L3, PAST, L4
GORGT/ L4, PAST, L1
GOTO/SETPT
```

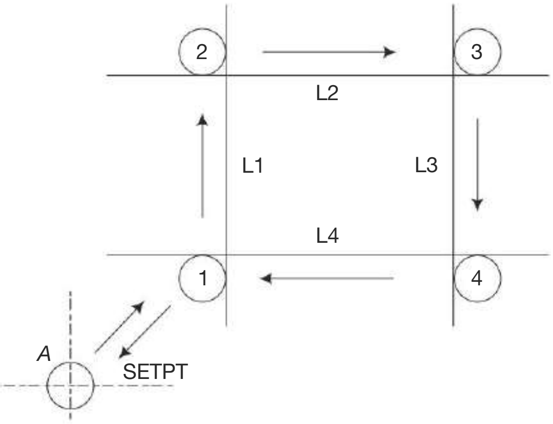


Fig. 16.39 *Implied check surface*

16.4.4 Motion Examples

In the following, few examples, complete motion statements are presented.

Example 16.6 (Fig. 16.24)

```
PARTNO/ EXAMPLE 16.1 FIG. 16.24
FROM/ 0, 0, 50
CUTTER/ 20
TOLER/ 0.01
GO/L1, (PL1 = PLANE/ 0, 0, 1, 3), L5
AUTOPS
GODLTA/ -8
TLLFT, GOLFT/ L1, PAST, L2
GORGT/ L2, PAST, L3
```

```
GORGT/ L3, TANTO, C1
GOFWD/ C1, TANTO, L4
GOFWD/ L4, PAST, 1, INTOF, C2
GORGT/ C2, PAST, L5
GORGT/ L5, PAST, L1
GODLTA/ 8
GOTO/ 0, 0, 50
```

Example 16.7 (Fig. 16.25)

```
PARTNO/ EXAMPLE 16.2 FIG. 16.25
FROM/ -50, -50, 50
CUTTER/ 25
INTOL/ 0.01
GO/L4, (PL1 = PLANE/ 0, 0, 1, 2), L1
AUTOPS
GODLTA/ -7.5
TLRGT, GORGT/ L4, PAST, L3
GOLFT/ L3
GOLFT/ L2
GOLFT/ L8
GORGT/ L1
GORGT/ L6, TANTO, C1
GOFWD/ C1, TANTO, L5
GOFWD/ L5
GORGT/ L1, PAST, L7
GODLTA/ 7.5
GOTO/ -50, -50, 50
```

Example 16.8 (Fig. 16.26)

```
PARTNO/ EXAMPLE 16.3 FIG. 16.26
FROM/ 0, 0, 50
CUTTER/ 25
INTOL/ 0.02
GO/L1, (PL1 = PLANE/ 0, 0, 1, 2), L4
AUTOPS
GODLTA/ -6
TLRGT, GOLFT/ L1, PAST, 2, INTOF, C1
GOLFT/ C1, TANTO, L2
```

```
GOFWD/ L2
GORGT/ L3
GOFWD/ C2
GOFWD/ L4, PAST, L1
GODLTA/ 6
GOTO/ 0, 0, 50
```

Example 16.9 (Fig. 16.40)

```
PARTNO/ EXAMPLE 16.6 FIG. 16.40
FROM/ SETPT
CUTTER/ 25
INTOL/ 0.02
GO/L4, PS, TO, L5
AUTOPS
GODLTA/ -6
GORGT/ L4, TANTO, C4
GOFWD/ C4, TANTO, L3
GOFWD/ L3
GOFWD/ C3
GOFWD/ L2
GOFWD/ C2, TANTO, L1
GOFWD/ L1, PAST, 2, INTOF, C1
GOFWD/ C1
GOFWD/ L5, PAST, L4
GODLTA/ 6
GOTO/ SETPT
```

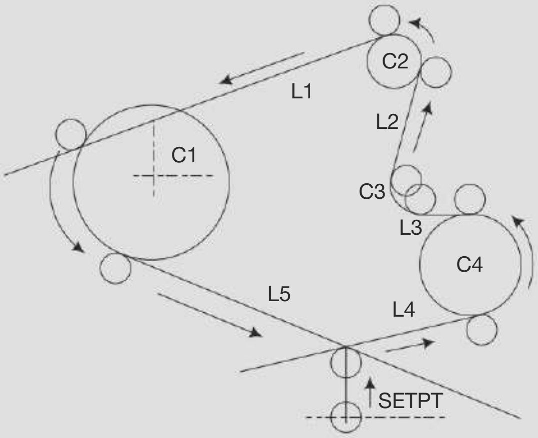


Fig. 16.40 *Example 16.6*

16.5 POSTPROCESSOR COMMANDS

These commands are used to specify the machine-tool functions and are supposed to be acted upon by the postprocessor identified earlier in the part program.

$$\text{COOLNT/}\left\{\begin{matrix} ON \\ OFF \\ MIST \\ FLOOD \\ TAPKUL \end{matrix}\right\}$$

This specifies the type of cutting fluid application to be used and when to start or stop.

$$\text{CYCLE/}\left\{\begin{matrix}\mathit{DRILL}\\ \mathit{FACE}\\ \mathit{TAP}\\ \mathit{BORE}\\ \mathit{REAM}\end{matrix}\right\},\ \text{Z},\ \left\{\begin{matrix}\mathit{IPM}\\ \mathit{IPR}\\ \mathit{MMPM}\\ \mathit{MMPR}\end{matrix}\right\},\ \text{feed, rapid\_pl}$$

IPM	Inch per minute
IPR	Inch per revolution
MMPM	mm per minute
MMPR	mm per revolution

This command specifies the generation of a fixed sequence of motions similar to the canned cycles—*z* is the depth through the work material, which the tool has to travel with *f*, the feed rate given in the units specified. The rapid_pl specifies the position of the rapid plane above the point where the cycle will be effective.

$$\text{FEDRAT/}\left\{\begin{matrix}\mathit{IPM}\\ \mathit{IPR}\\ \mathit{MMPM}\\ \mathit{MMPR}\end{matrix}\right\},\ \text{feed}$$

This specifies the rate at which the slides move in appropriate units as designated.

LOADTL/ toolno, magpos

It indicates the loading of the tool number, which is loaded in the tool magazine position indicated.

REWIND

This command is inserted to rewind the control tape to a known tape mark, which is generally the start of the tape (part program).

$$\text{SPINDL/}\left\{\begin{matrix}\mathit{SFM}\\ \mathit{RPM}\\ \mathit{SMM}\end{matrix}\right\},\ \text{n},\ \left\{\begin{matrix}\mathit{CLW}\\ \mathit{CCLW}\end{matrix}\right\}$$

SFM	Surface feet per minute
RPM	Revolutions per minute
SMM	Surface metres per minute

This command signifies the selection of spindle speed in appropriate units specified in the statement.

16.6 COMPILATION CONTROL COMMANDS

The segment covers statements that normally prepare the computer for accepting the part program, improve the readability of the part program and control the output of the computer.

PARTNO/ <literal string>

The PARTNO is used as identification for the part program and as such, should be the very first statement in a part program. It should also be noted that in a given line, PARTNO should appear in the first six character positions without any blanks. The length of <Literal string> will depend on the particular implementation, e.g., PARTNO/LEVER 138C

The FINI statement is the physical end of a part program and should always be present at the end of any part program.

MACHIN/name, <parameters>

The MACHIN statement identifies the postprocessor to be used for outputting the necessary NC blocks. **name** signifies the name of the particular post processor identification name in literal strings, e.g., EXPOST, followed by < parameters > which initialises the various flags in the postprocessor program. The number of such parameters depends upon a particular implementation, e.g.,

```
MACHIN/EXPOST, 2,2,1,OPTION, 0
```

When no postprocessing of CLDATA is required then the following statement may be specified which suppresses the postprocessing.

```
NOPOST
```

To add comments in a part program, the command REMARK may be used as shown:

```
REMARK/GEOMETRICAL DEFINITIONS
```

This statement is ignored by the processor and as such, is identical to a $$ in usage, the only difference being that REMARK should be present only in the first six character positions of a line.

16.7 REPETITIVE PROGRAMMING

Just as DO loop, subroutines and macros are used in manual part programming, similarly facilities are available for repetitive programming in computer-aided programming. These are described here.

16.7.1 Looping

Normally, a part program is executed sequentially starting from a PARTNO statement to the FINI statement. But it is possible to change this sequential execution by means of the transfer statements available in APT.

JUMPTO	unconditional transfer
IF	conditional transfer

The usage is

```
JUMPTO/ lbl1
```

A better option for looping is the arithmetic IF statement which allows a conditional transfer to a segment of the program depending on the value of an arithmetical expression. The general usage is

```
IF (< expression>)          lb11, lb12, lb13
```

When the numerical value of the < expression > is negative, zero or positive, then control is transferred to the statement referenced by lb11, lb12, or lb13 respectively. It is always necessary to label the statement which immediately follows the IF statement in a part program. The <expression > could be a variable or an arithmetic expression.

```
        X= 0
LB0)    YVAL = 20
LB1)    GOTO/X, YVAL, 0
        GODLTA/-10
        GODLTA/10
        YVAL = YVAL+30
        IF (500 -YVAL) LB2, LB1, LB1
LB2)    X=X + 50
        IF (500 -X) LB3, LB, LB0
LB3)    -----------
```

16.7. 2 MACRO

The sequence of similar or identical statements which need to be referred more often in a part program are best referred by a MACRO statement such that the part-program bulk is reduced. This statement is very similar to a FORTRAN SUBROUTINE statement. The syntax is

```
<name >= MACRO/<parameters>
-----------
-----------
TERMAC
```

All the statements that are enclosed between a MACRO statement and a TERMAC statement are to be executed whenever this macro is called by

```
CALL/name, <parameters>
```

The MACRO definition should always end with TERMAC. The definition of a given MACRO should always precede the first calling of that MACRO. A MACRO definition should not contain another MACRO definition but can call another MACRO which is already defined. The maximum number of parameters one can specify in a given MACRO definition depends on the type of implementation. The parameters of a MACRO can be assigned values at the time of calling or can be initialised (i.e., the default values) in the definition itself. When the variables are initialised at the time of definition, then there is no need to reassign the values at the time of execution, if they are not different from the normal values. FINI is not allowed in a MACRO definition. It is also not allowed to label the first statement in the MACRO definition which defines the name of the MACRO.

16.7.3 TRACUT

The result of TRACUT usage in motion statements is to TRAnspose the CUTter locations only without actually altering the original geometrical definitions. This is useful particularly for repetitive geometries. The usage is

```
TRACUT/matrix
--------------
--------------
TRACUT/NOMORE
```

TRACUT is not accumulative. A TRACUT /NOMORE cancels the present TRACUT which is in force. But a second TRACUT statement without NOMORE would also cancel a previous transformation matrix in force. It is preferable to have an explicit NOMORE to make things clear. Care should be exercised in locating the cutter while moving through a number of TRACUT statements, the reason being that the tool will move in a straight line from the last point of one TRACUT to the first point of the next TRACUT hitting any obstruction if present in the way.

A few examples of complete APT part programs (geometry and motion statements only) are given here to illustrate all the points made above.

Example 16.10 Write a complete part program for the following component shown in Fig. 16.41 using an end-mill cutter of 20 mm diameter. Clearly show the axes system chosen with a sketch and the direction of the cutter for the motion statements.

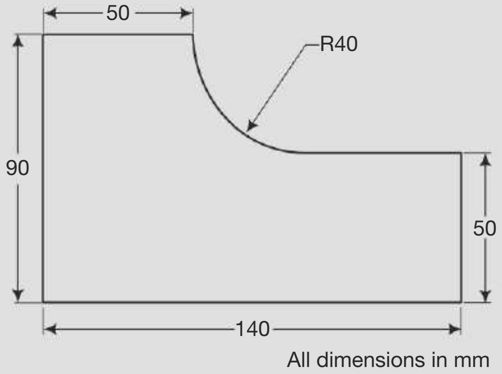


Fig. 16.41 *Part drawing for Example 16.10*

Solution The first step in the solution process is to identify the axes system and give names to all the important geometric elements to be designated, as in Fig. 16.42. Always define the minimum number of geometry elements required. Notice here that there is no mention of how the part will be clamped on a machining centre table. Normally, for that purpose a separate set of holes are present in the part with which it is possible to clamp it to the table using the T-bolts.

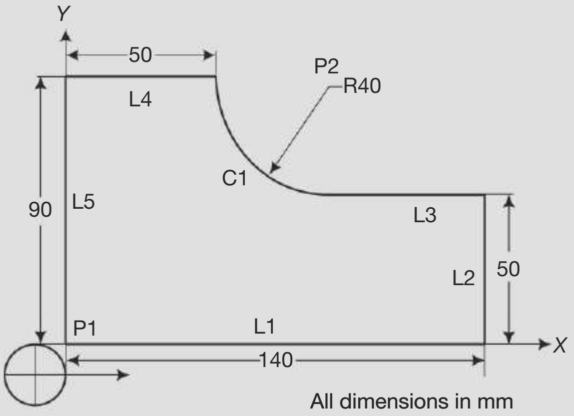
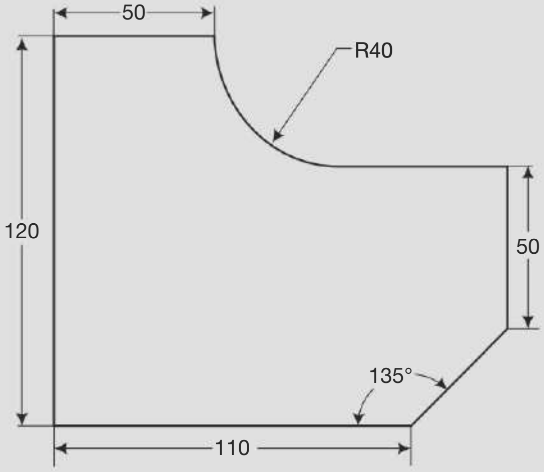


Fig. 16.42 *Part drawing for Example 16.10 with assigned names for the geometric elements*

```
                                  Comment
$$ Geometry statements
L1 = LINE/0, 0, 140, 0            $$ Using the coordinates of the endpoints
L2 = LINE/140, 0, 140, 50
L5 = LINE/0, 0, 0, 90
L4 = LINE/0, 90, 50, 90
C1 = CIRCLE/(50+40),90,40         $$ Using centre coordinates and radius
L3 = LINE/140, 50, (50+40), 50
```

```
PL1 = PLANE/ 0, 0, 1, 2              $$ Clearance plane parallel to XY above 2 mm
$$ Motion statements
FROM/ -50, -50, 50                   $$ Tool away from the work
CUTTER/ 20
TOLER/ 0.01
GO/L1, PL1, L5                       $$ Start-up command
AUTOPS
GODLTA/ -6                           $$ Tool goes to a depth of 4 mm
TLRGT, GORGT/ L1, PAST, L2
GOLFT/ L2, PAST, L3
GOLFT/ L3, TANTO, C1
GOFWD/ C1, INTOF, L4
GOLFT/ L4, PAST, L5                  $$ Last part of contour
GOLFT/ L5, PAST, L1                  $$ Lift the tool to clearance plane
GODLTA/ 6                            $$ Park the tool away from workpiece
GOTO/ -50, -50, 50
```

Example 16.11 Write a complete part program for the following component shown in Fig. 16.43 using an end-mill cutter of 20 mm diameter. Clearly show the axes system chosen with a sketch and the direction of the cutter for the motion statements.



All dimensions in mm

Fig. 16.43 *Part drawing for Example 16.11*

Solution The first step in the solution process is to identify the axes system and give names to all the important geometric elements to be designated, as in Fig. 16.44. Always define the minimum number of geometry elements required.

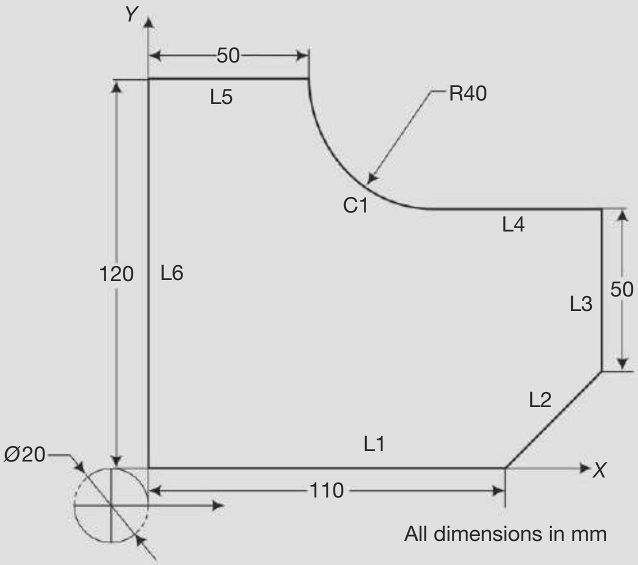


Fig. 16.44 *Part drawing for Example 16.11 with assigned names for the geometric elements*

```
                                         Comment
$$ Geometry statements
L1 = LINE/0, 0, 110, 0                   $$ Using the coordinates of the endpoints
L2 = LINE/110, 0, 140, (120-40-50)
L3 = LINE/140, 30, 140, (120 - 40)
L4 = LINE/140, 80, (50+40), 80
C1 = CIRCLE/(50+40),120,40               $$ Using centre coordinates and radius
L5 = LINE/0, 120, 50, 120
L6 = LINE/0, 0, 0, 120
PL1 = PLANE/ 0, 0, 1, 2                  $$ Clearance plane parallel to XY above 2 mm
$$ Motion statements
FROM/ -50, -50, 50                       $$ Tool away from the work
CUTTER/ 20
TOLER/ 0.01
GO/L1, PL1, L6                           $$ Start-up command
AUTOPS
GODLTA/ -6                               $$ Tool goes to a depth of 4 mm
TLRGT, GORGT/ L1, L2
GOLFT/ L2, L3
GOLFT/ L3, L4
GOLFT/ L4, TANTO, C1
```

```
GOFWD/ C1, INTOF, L5
GOLFT/ L5, L6
GOLFT/ L6, PAST, L1                     $$ Last part of contour
GODLTA/ 6                               $$ Lift the tool to clearance plane
GOTO/ -50, -50, 50                      $$ Park the tool away from workpiece
```

Example 16.12 Write a complete part program for the following component shown in Fig. 16.45 using an end-mill cutter of 20 mm diameter. Clearly show the axes system chosen with a sketch and the direction of the cutter for the motion statements.

Solution The first step in the solution process is to identify the axes system and give names to all the important geometric elements to be designated, as in Fig. 16.44. Always define minimum number of geometry elements required.

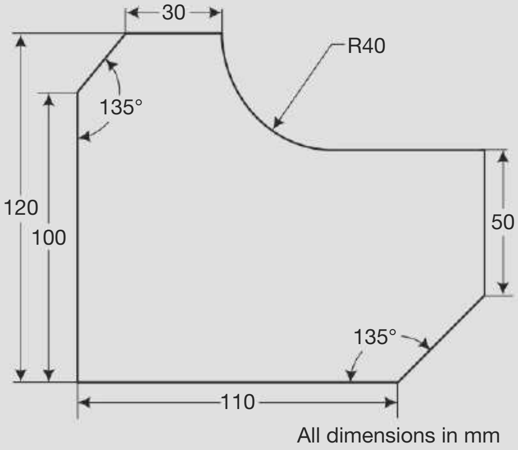


Fig. 16.45 *Part drawing for Example 16.12*

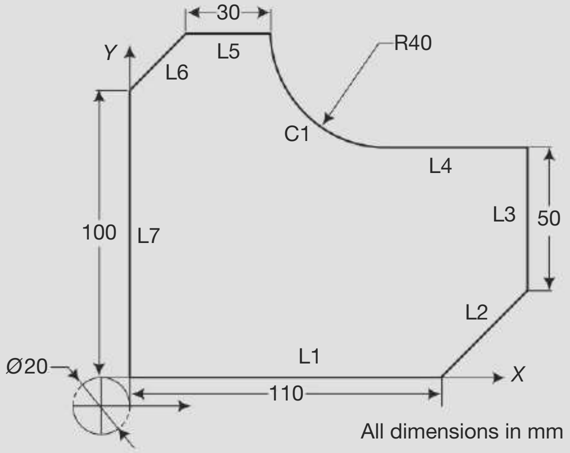


Fig. 16.46 *Part drawing for Example 16.12 with assigned names for the geometric elements*

```
                                        Comment
$$ Geometry statements
L1 = LINE/0, 0, 110, 0                  $$ Using the coordinates of the endpoints
L2 = LINE/110, 0, 140, 30
L3 = LINE/140, 30, 140, (120 – 40)
L4 = LINE/140, 80, (50+40), 80
C1 = CIRCLE/(50+40),120,40              $$ Using centre coordinates and radius
L5 = LINE/0, 120, 50, 120
L6 = LINE/0, 100, 20, 120
L7 = LINE/0, 0, 0, 120
PL1 = PLANE/ 0, 0, 1, 2                 $$ Clearance plane parallel to XY above 2 mm
```

```
$$ Motion statements
FROM/ -50, -50, 50                 $$ Tool away from the work
CUTTER/ 20
TOLER/ 0.01
GO/L1, PL1, L6                     $$ Start-up command
AUTOPS
GODLTA/ -6                         $$ Tool goes to a depth of 4 mm in the workpiece
TLRGT, GORGT/ L1, L2
GOLFT/ L2, L3
GOLFT/ L3, L4
GOLFT/ L4, TANTO, C1
GOFWD/ C1, INTOF, L5
GOLFT/ L5, L6
GOLFT/ L6, L7
GOLFT/ L7, PAST, L1                $$ Last part of contour
GODLTA/ 6                          $$ Lift the tool to clearance plane
GOTO/ -50, -50, 50                 $$ Park the tool away from workpiece
```

Example 16.13 Write a complete part program for the following component shown in Fig. 16.47 using an end-mill cutter of 20 mm diameter. Show the statements only for the profile shown in the top view up to a depth of 5 mm in one cut. Clearly show the axes system chosen with a sketch and the direction of the cutter for the motion statements.

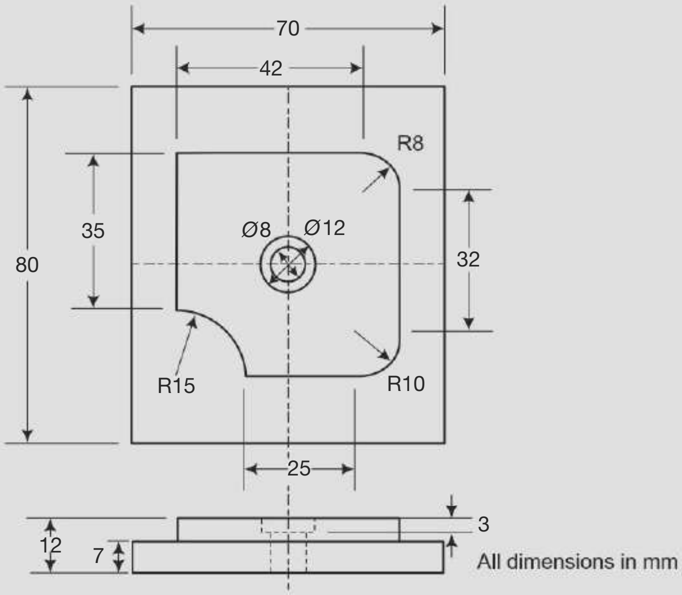


Fig. 16.47 *Part drawing for Example 16.13*

Solution The first step in the solution process is to identify the axes system and give names to all the important geometric elements to be designated, as in Fig. 16.48. Always define the minimum number of geometry elements required. The tool is starting from the top left-hand corner so that the start-up command starts with two lines.

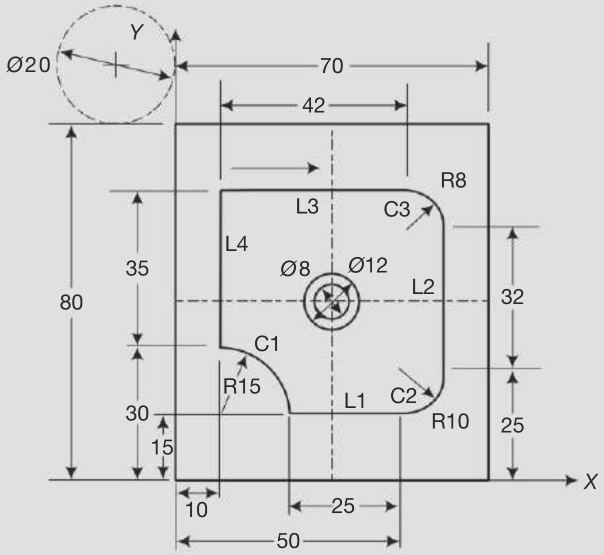


Fig. 16.48 *Part drawing for Example 16.13 with assigned names for the geometric elements*

```
                                                 Comment
$$ Geometry statements
L1 = LINE/0, 15, 70, 15                          $$ Using the coordinates of the endpoints
L2 = LINE/(50+10), 0, 60, 80
L3 = LINE/0, (30+35), 70, 65
L4 = LINE/0, 10, 0, 80
C1 = CIRCLE/15,15, 15                            $$ Using centre coordinates and radius
C2 = CIRCLE/50,25, 10
C3 = CIRCLE/52,(30+35–8), 8
PL1 = PLANE/ 0, 0, 1, 2                          $$ Clearance plane parallel to XY above 2 mm
$$ Motion statements
FROM/ -50, 130, 50                               $$ Tool away from the workpiece
CUTTER/ 20
TOLER/ 0.01
GO/L3, PL1, L4                                   $$ Start-up command
AUTOPS
GODLTA/ -7                                       $$ Tool goes to a depth of 5 mm into the workpiece
```

```
TLLFT, GOLFT/ L3, TANTO, C3
GOFWD/ C3, TANTO, L2
GOFWD/ L2, TANTO, C2
GOFWD/ C2, TANTO, L1
GOFWD/ L1, INTOF, C1
GORGT/ C1, INTOF, L4
GORGT/ L4, PAST, L3                    $$ Last part of contour
GODLTA/ 6                              $$ Lift the tool to clearance plane
GOTO/ -50, -50, 50                     $$ Park the tool away from the workpiece
```

16.8 COMPLETE PART PROGRAM IN APT

In the following is shown the example of a complete part program for the part shown in Fig. 16.28. The machining of the four pockets involves the following steps:

1. Centre drill four holes P1, P2, P3 and P4
2. Drill of these holes through
3. Rough mill of each pocket to a depth of 3 mm leaving 0.5 mm for finishing
4. Rough mill to entire depth
5. Finish mill of complete pocket
6. Repeat steps (iii) to (v) for the other three pockets

The complete part program is presented below with the necessary remarks to facilitate understanding.

```
PARTNO MILLING EXERCISE ON MOOG HYDRAPOINT
REMARK PROGRAMMER P.N.RAO
MACHIN/63, 3, 2, 1, 0 OPTION, 0
MACHIN/40, OPTION, 3, 1, 1, 2, 2, 2, 2, 2, 3          $$ PLOTTER
PPLOT/ALL,LOWLFT,-60,0,0,XYPLAN, UPRGT, 120, 160, 0, SCALE, 0.04
PRINT/ON                                              $$ TO GET CANONICAL INFORMATION
CLPRNT/ON                                             $$ TO GET CENTRE LINE DATA
TOLER/0.01
REMARK GEOMETRY STATEMENTS
L4 = LINE/ XAXIS
L5 = LINE/ YAXIS
L3 = LINE/ PARLEL, L5, XLARGE, (20 - 5)
LM = LINE/ PARLEL, L5, XLARGE, (140/2)
C1 = CIRCLE/ (20 + 20), 22.5, 5
C2 = CIRCLE/ 20, (15 + 15), 5
C3 = CIRCLE/ 20, 15, 5
L1 = LINE/ LEFT, TANTO, C1, LEFT, TANTO, C3
L2 = LINE/ RIGHT, TANTO, C1, RIGHT, TANTO, C2
```

```
L3 = LINE/ PARLEL, L5, XLARGE, (20 – 5)
P1 = POINT/ CIRCLE, C1
P2 = POINT/ (20 + 20), (22.5 + 55)
P3 = POINT/ (20 + 100 - 20), (22.5 + 55)
P4 = POINT/ (20 + 100 - 20), 22.5
PAT= PATERN/ RANDOM, P1, P2, P3, P4
M1=MATRIX/ TRANSL, 0, 55, 0                 $$ FOR POCKET 2
M2=MATRIX/ MIRROR, LM                       $$ FOR POCKET 4
M3=MATRIX/XYROT,180,TRANSL,(20+120),15+15+15 + 55)   $$FOR POCKET 3
PLN = PLANE/ 0, 0, 1, 2                     $$ CLEARANCE PLANE
TPP = POINT/ -50, 150                       $$ START POINT
REMARK MOTION STATEMENTS START HERE
CLRSRF/ (76 + 19.5)                         $$ TOOL ABOVE WORK
FROM/ TPP
REMARK CENTRE DRILLING
TOOLNO/ 1, LENGTH, 19.5
SPINDL/ RPM, 2000, CLW
COOLNT/ MIST
CYCLE/ DRILL, 8, MMPM, 160, 3
GOTO/PAT
COOLNT/OFF
CYCLE/OFF
GOTO/TPP
TOOLNO/ 3, LENGTH, 85
SPINDL/ RPM, 1500, CLW
COOLNT/ ON
CYCLE/ DRILL, 10.5, MMPM, 200, 3
GOTO/ PAT
COOLNT/ OFF
CYCLE/ OFF
GOTO/ TPP
REMARK MILLING OF POCKETS
TOOLNO/ 4, LENGTH, 20.5                     $$ MILL TOOL
SPINDL/ RPM, 1000, CLW
COOLNT/ ON
MAC1 = MACRO/                               $$ MACRO FOR GOING ROUND
FEDRAT/ MMPM, 60                            $$ THE POCKET ONCE
TLRGT, GORGT/ L1, TO, L3
GORGT/ L3, TO, L2
```

```
GORGT/ L2, TO, L1
TERMAC
MAC2 = MACRO/                 $$ MACRO FOR COMPLETE MACHINING
RAPID                         $$ OF A POCKET
CUTTER/ 11                    $$ LEAVE 0.5 FOR FINISHING
GO/ L1, PLN, PAST, L2
CUT
CYCLE/ MILL, 3, MMPM, 60      $$ MILL TO 3 DEPTH
CALL/ MAC1
CYCLE/ MILL, 8                $$ MILL FULL DEPTH
CALL/ MAC1
CUTTER/ 10                    $$ FINISH THE POCKET
GO/ L1, PLN, L2
CALL/ MAC1
CYCLE/ OFF
DNTCUT
RAPID
GOTO/ TPP                     $$ THEORETICAL MOTION ONLY
TERMAC
CALL/ MAC2                    $$ MACHINE POCKET 1
TRACUT/ M1
CALL/ MAC2                    $$ MACHINE POCKET 2
TRACUT/ NOMORE
TRACUT/ M3
CALL/ MAC2                    $$ MACHINE POCKET 3
TRACUT/ NOMORE
TRACUT/ M2
CALL/ MAC2                    $$ MACHINE POCKET 4
TRACUT/ NOMORE
CUT                           $$ LAST MOTION STATEMENT
REWIND
FINI
```

After entering the program, a printout of the APT processor can be obtained showing the canonical information of all the geometry defined and the cutter-location data. A plot of the CLDATA is also obtained to prove the validity of the program.

16.9 CAM SYSTEMS

The APT programming language that is described previously utilises a programming-language approach that is not very convenient compared to the various graphical-user-interface-based systems that are available for developing CNC part programs. In these systems, a dedicated CAM system helps in developing the CNC part programs. The CAM systems may optionally be linked to any major CAD system such as AutoCAD, Solidworks, and CADKEY. Typical examples are Mastercam, Virtual Gibbs, Smartcam, Surfcam, Edgecam, and Alphacam.

Out of these, Mastercam is the most popular system with a large number of actual installations in industry as well as educational institutions. It also links directly to most of the popular CAD systems, thereby making it one of the best options to take care of the CAD investments made by the industry. It incorporates a comprehensive cutting-tool database for cutting-process-parameter selection utilising the actual tools from the various cutting-tool manufacturers. This helps in customising the cutting-tool database of individual companies to suit their requirements. Also, a large amount of intelligence is built into the various tool-path modules available for tool-path generation. This means that the user has to give very little information and most of the required information can be directly taken from the part drawing. Technical decisions can be left to Mastercam most of the times. We will study a few details of the Mastercam programming system with some examples as to its use in typical situations.

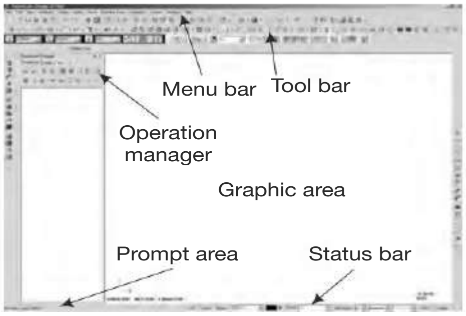


Fig. 16.49 *Opening screen of Mastercam version X3*

The opening screen of Mastercam is shown in Fig. 16.49. The screen is divided into a number of areas with each serving a specific function. They are the following:

- ***Menu Bar*** This is the main part of the menu with which the user interacts most of the time. This contains a number of sections that are used to create geometry and tool paths, besides other utilities. It is similar to the Windows menu system.
- ***Tool Bar*** It contains complete commands in the form of keys. These can be used instead of going through the main menu, which are in the form of a tree with a number of embedded layers.
- ***Graphics Area*** This is the main workspace where the user views, creates, and modifies geometry, drafting entities, and tool paths in Mastercam.
- ***Prompt Area*** System displays data and the necessary prompts for the user input data.
- ***Status Bar*** It indicates the current selection of some important elements related to the geometric environment used.
- ***Operations Manager*** It houses a number of functions related to the tool-path handling.

16.9.1 Main Menu Commands

These are the main set of command options that allow the user to navigate for developing the CNC part program for a given application. This has a number of submenus that are shown below:

Analyze Displays information about geometry

Create This option is used to add geometry to the system's database while drawing it on the screen. The various facilities available under this category will be discussed later in design creation.

Edit This option is used to edit geometry, such as the Join entities, Modify Spline, Convert NURBS, and Simplify functions, and the Trim/Break submenu functions in addition to the normal editing functions such as cut, copy, paste, delete, or select all entities in the graphics window.

File This command allows to manipulate the disk files. The various options available are shown in Fig. 16.50. This has options similar to other Windows applications. 'Open' is used to open existing files in any of the acceptable formats. It allows for direct translation from standard formats such as AutoCAD drawing format or Auodesk Inventor format, while others are in the neutral data formats discussed in Chapter 5, such as STEP and IGES. The various options available are shown in Fig. 16.51.

Fig. 16.50 *Options available for file options in Mastercam version X3*

Fig. 16.51 *Various data translators built in the file options in Mastercam version X3*

Xform This option allows for the transformation of screen geometry. More details of this will be discussed later.

Toolpaths Use this command to generate the NC tool paths. This will be discussed in detail later.

Mastercam supports the work coordinate system indirectly. It provides a set of eight pre-defined system views for working in 3D space: Top (X, Y) Front (X, Z), Back ($-X$, Z), Left Side ($-Y$, Z), Right Side (Y, Z), Bottom ($-X$, Y), Isometric and Axonometric. In addition, the user can define other views relative to the top view, which can then be used to define geometry. The work plane will be defined using the construction plane (Cplane) command discussed later. The secondary menu commands are used for controlling some of the parameters related to the geometric input.

Z This defines the z depth of geometry, which is considered as any axis perpendicular to the specified view. For example, in the case of the top view, it will be the Z-axis, whereas for the front view, it will be the Y-axis.

Color This allows for the assignment of system colour. All the objects that will be drawn are in this colour. It is also possible to change the colour of the objects later.

Level This is similar to layer in other CAD programs as discussed in Chapter 6. The user can set any one of 255 levels to be the main level such that the geometry defined can go to that level. The main level is set using this button. Mastercam also allows copying or moving of geometry from level to level, hiding levels from view, provides level-naming utilities, and allows to organise several levels into sets.

Style/Width This sets the current line style and width such that all the new objects drawn in the graphic window have these properties.

Group Groups allow one to assemble entities into a single unit for selection. Whenever the user needs to perform a function such as mirroring, etc., he needs to select the geometric entities. If there are a large number of entities then the selection process to pick individual entities is long and boring. In such cases, grouping helps.

Mask Affects object selection process by avoiding the type of mask specified.

Tplane The Tool Plane (Tplane) is the plane in which the tool approaches and cuts the metal from the part. The tool plane represents the CNC machine's coordinate system (*XY* axis and origin).

Cplane The construction plane is a flat, two-dimensional plane that can be defined anywhere in the 3-dimensional space. When creating geometry, the user will always be working on a Cplane. Usually, the default plane (top *XY*) is adequate for 2D work such as plain milling. When working in 3-dimensional environment, the user needs to change the construction plane similar to the work coordinate system.

Gview The graphics view signifies the plane from which the user views the work in the graphics window. All the standard views and isometric view are available. Optionally, the user can decide a specific orientation to best view the features in the case of 3-dimensional geometry.

Default Key Assignment Special keyboard assignments provide quick access to many common functions. The following are the default commands provided by Mastercam.

Alt + 0	Set Z depth for Cplane
Alt + 1	Set main colour
Alt + 4	Choose tool plane (Tplane)
Alt + 5	Choose construction plane (Cplane)
Alt + 6	Choose graphics view (Gview)
Alt + A	AutoSave
Alt + B	Toolbar on/off
Alt + D	Drafting global parameters
Alt + G	Selection grid parameters
Alt + H	On-line help
Alt + J	Job set-up
Alt + L	Set entity attributes
Alt + O	Operations Manager
Alt + Q	Undo last operation
Alt + R	Edit last operation
Alt + S	Full-time shading on/off
Alt + T	In Toolpath menu, turn tool path display on/off
Alt + U	Undo last action
Alt + X	Set main colour, level, line style and width from selected entity
Alt + Z	Set visible levels
Alt + F1	Fit geometry to screen

Alt + F2	Unzoom by 0.8
Alt + F3	Cursor tracking on/off
Alt + F4	Exit Mastercam
Alt + F5	Delete using window selection
Alt + F7	Blank geometry
Alt + F8	System configuration
Alt + F9	Display all axes
Alt + F10	Maximize and minimize screen
F1	Zoom
F2	Unzoom
F3	Repaint
F4	Show Analyze menu
F5	Show Delete menu
F6	Show File menu
F7	Show Modify menu
F8	Show Create menu
F9	Part information on/off
F10	List all functions and execute selected
Esc	System interrupt or menu backup
Page up	Zoom in by 0.8
Page down	Zoom out by 0.8
Arrow keys	Pan

16.9.2 Geometry

The geometry input function in Mastercam allows for creation of the geometry as well as its modification. The edit and transform functions allow the geometry to be modified to add further details as required. The Create command in the main menu gives the options for creating geometric and drafting entities. The functions on the Create menu can be divided into three categories:

Basic-Geometry Creation Functions These functions let you create simple and basic Mastercam entity types. They are the following:

- **Creating points** This option allows for creating the points that could be used for hole-making operations, or as points for creating other geometric elements.
- **Creating lines** This option allows for a variety of definitions to be used for defining lines.
- **Creating arcs** This option lets the user define a range of arcs and circles.
- **Creating splines** This option is used to create splines and other types of curves based a on a series of points.
- **Creating surfaces** This option is meant for defining surfaces that are analytical and sculptured as discussed in Chapter 4.
- **Creating drafting entities** This is used for defining the dimension and other annotations required for drafting purpose.

Complex-Geometry Creation Functions These functions create complex entities based on certain analytical principles. Some of these are the following:

- **Filleting curves** Fillet function creates arcs that are tangent to curves while trimming the curves beyond the filet.
- **Creating rectangles** Creates rectangular shaped objects such as simple rectangles or rectangles with corner rounding.
- **Chamfering lines** Creating chamfers between any two objects and trimming the overflowing curves beyond the chamfer.
- Creating an ellipse
- Creating a polygon
- Creating a spiral and a helix

Creating Points The Point menu in the Create options allows the user to create point entities in a number of ways. Some of these are

- Creating a point at any position in the graphics window
- Creating points at fixed intervals along a curve
- Creating points at the node points of a parametric spline
- Creating points at the control points of a NURBS spline
- Creating a point at any position on a curve or surface
- Creating a point at any position on a solid face
- Creating a point at a defined distance along a curve
- Creating points at the intersections of a slice plane with curves
- Projecting points onto surfaces and solids
- Creating a point at a perpendicular distance from a curve
- Creating points in a grid pattern
- Creating points in a bolt circle pattern
- Creating points at the centre of selected arcs

Entering of point data in Mastercam can be done by following the prompts from the system. The points are actually entered in 3D space. The system automatically projects from the 2D to the current *Z* position in the construction plane. It also allows you to pick from the existing objects in a number of ways.

- Entering a point using *XYZ* coordinates
- Entering a point at the system origin
- Entering a point at the centre of an arc
- Entering a point at the endpoint of an entity
- Entering a point at the intersection of two curves
- Entering a point at the midpoint of a curve
- Entering a point at the position of an existing point
- Entering a point at the last entered position
- Entering a point relative to the position of another point
- Entering a point at a quadrant position on an arc
- Sketching a point at any position
- Setting up a grid for point entry

Creating Lines The Line menu gives you various options for creating line entities. Some of these are

- Creating a horizontal line
- Creating a vertical line
- Creating a line between any two points
- Creating multiple lines connected at their endpoints
- Creating polar lines using the radius and angle

- Creating tangent lines
- Creating perpendicular lines
- Creating parallel lines
- Creating a bisecting line or a midline
- Creating a line at the closest position between two curves or a curve and a point

Creating Arcs The Arc option is used to create arc entities that also include circles. Similar to other CAD systems, Mastercam generates arcs in a counter-clockwise direction. The various options available are

- Creating polar arcs
- Creating an arc with a radius and endpoints
- Creating an arc with three points
- Creating tangent arcs
- Creating a circle using two points
- Creating a circle using three points
- Creating a circle with a centre point and radius
- Creating a circle with a centre point and diameter
- Creating a circle with a centre and point on the circumference

Filleting Curves The Fillet menu gives you various options for generating fillets between curves. It creates an arc of a given radius tangent to the selected curves. It can either trim the curves beyond the fillet or retain them depending on the choice by the user.

Creating Rectangular-Shaped Geometry The Rectangle option gives a number of choices for creating rectangles and other geometric shapes as shown in Fig. 16.52. All these options are created by a combination of lines and/or arcs and are defined within a rectangular boundary. It is also possible to create fillets at sharp corners on the rectangular shape, and/or a surface within the boundary of the rectangular shape.

Chamfering Lines This option lets the user create a chamfer of given specification between two lines. The various details required for the chamfer are given through the screen prompts.

Creating an Ellipse This option allows the creation of an ellipse with two axes and a rotation of the major axis. The options are set with the help of the pop-up window as shown in Fig. 16.53. It is possible to create a complete ellipse or partial, by defining the start and end angles.

Creating a Polygon This option generates a polygon of any number of sides. The options are set with the help of the pop-up window as shown in Fig. 16.54.

Fig. 16.52 *Options available for rectangle generation in Mastercam version X3*

Fig. 16.53 *Options that need to be specified for ellipse generation in Mastercam version X3*

Fig. 16.54 *Options that need to be specified for creating a polygon in Mastercam version X3*

The geometry created can be modified using the two options Edit and Transform. The following are some of the options available:

- Trimming curves
- Trimming surfaces
- Breaking entities
- Joining segments of broken curves
- Extending entities
- Dragging entities to new positions in the graphics window
- Mirroring entities
- Rotating entities
- Scaling entities
- Translating entities (move and copy)
- Offsetting entities

A few details of the facilities are described below:

Trimming Curves The Trim menu option allows for trimming curves to one another. Mastercam trims curves by cutting them back or extending them at their intersections. When the intersection between two curves offers many solutions, the user should select each curve on the part of the curve that is to be retained. The Trim options are the following:

- **Trimming one curve** This trims one curve to its intersection with a second curve without modifying the second entity.
- **Trimming two curves** This trims two curves to their intersection.
- **Trimming three curves** This trims three curves. The first two curves that you select are trimmed to the third selected curve, which acts as a trimming curve. The third curve is then trimmed to the first two curves.
- **Trimming a curve to a point** This function trims a curve to a point defined in the graphics window if the point that you enter does not lie on the selected entity.
- **Trimming multiple curves to a single trimming curve** This function trims multiple curves to a selected trimming curve without modifying the trimming curve.
- Closing an arc to form a full circle.
- **Dividing a line or arc between two trimming curves** This function trims a line or arc into two separate segments by removing the segment that lies between two dividing curves.

Breaking Entities This option lets you break curves into multiple segments. The different options available that are related to common geometry definitions are

- Breaking a curve into two segments at a defined point
- Breaking a curve into two segments at a defined length
- Breaking a line into multiple segments
- Breaking an arc into multiple arc or line segments
- Breaking curves at their intersections with other curves

Joining Segments of Broken Curves This option joins collinear lines, and arcs that have the same centre and radius.

Extending Entities This option extends the entities selected by a specified distance entered at the prompt. A negative value trims the object by the distance.

Mirroring Entities Mirroring is the process of creating mirror images of entities by reflecting them symmetrically with respect to a defined axis. Mirroring can be done about the X or Y-axis, and also about a defined line or any two points. It allows the option to move or copy the original entities within the drawing.

Rotating Entities Rotating is used to move or copy selected entities around a point by a defined angle. The angle is calculated relative to the horizontal axis of the current construction plane. A positive angle results in a counter-clockwise rotation, and a negative angle results in a clockwise rotation.

Scaling Entities Scaling is used to increase or decrease the size of entities by a factor relative to a defined point. When a single scale factor is used, the increase or decrease is proportional along all axes (XYZ). The entities change size while maintaining their original shape. It is also possible to apply a different scale factor to each of the axes (XYZ).

Translating Entities (move and copy) Translating is used for moving or copying selected entities to new positions without altering the orientation, size, or shape of the entities. You can move the entities within a plane or from one plane to another plane.

Offsetting Entities Offsetting is used to displace entities by a distance in a perpendicular direction. It can be used for offsetting a single curve, one or more chains of curves, or one or more surfaces. For curves, the offset direction is relative to the current construction plane.

- **Geometry Creation Data Entry Shortcuts**

  A – existing angle
  D – existing diameter
  L – existing length
  R – existing radius
  X – existing X-coordinate
  Y – existing Y-coordinate
  Z – existing Z-coordinate

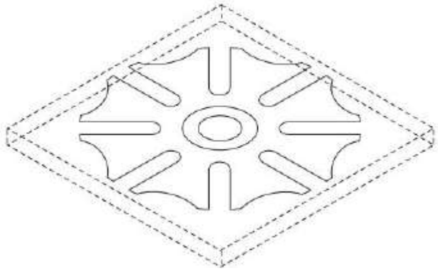


Fig. 16.55 *Typical example of a part geometry created. Only the geometry in top view needs to be created for 2D machining*

16.9.3 Tool-Path Generation for Planar Milling

Tool path specifies the path through which the tool traverses to remove the material from the stock. There are a large variety of tool paths that can be used in Mastercam. Before discussing the details of the tool paths, it is necessary to understand the basic steps that need to be followed in developing the tool path. Steps needed are

- Job set-up
- Identify the operation and geometry to be performed
- Select the tool required for the operation
- Specify the operating parameters
- Simulate the tool path generated

The next selection to be made is the postprocessor to be used for generating the NC program. The selection can be done by clicking the Machine Type from the main menu. A large number of commonly used postprocessors are part of the Mastercam library. It is also possible to define a new postprocessor by specifying all the needed parameters and customising the generalised postprocessor provided by Mastercam.

Job Set-up In the job set-up, all the data needed for the setting of the part for machining is given. The pop-up window is shown in Fig. 16.56. The details entered are

- Stock size which can be specified as the dimensions along the *X*, *Y* and *Z*-axes
- Stock origin which can be used to set up the job on the machine tool table
- Material of the workpiece is also specified. Mastercam maintains a large number of commonly used materials. It is also possible to add other materials as required. The selection can be done by clicking the button in the job set-up window and the resulting window is shown in Fig. 16.57. Mastercam will automatically take tool and stock material into consideration when calculating speed and feed needed for an operation. If necessary, the user can modify these values.

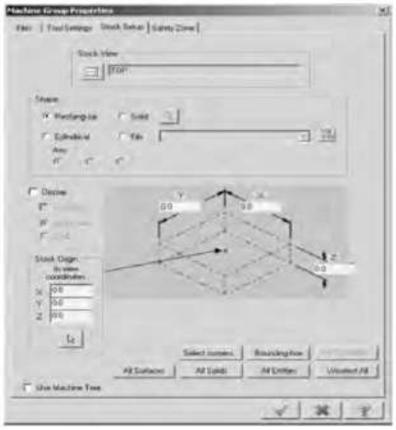
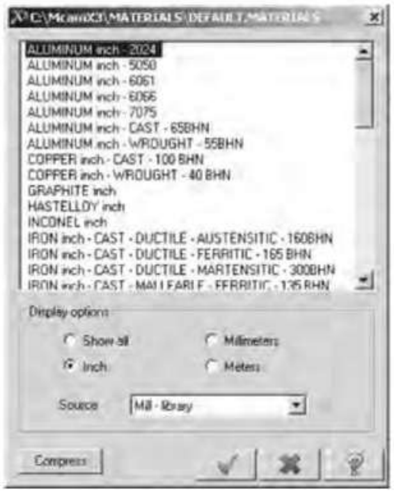


Fig. 16.56 *The pop-up window for job set-up in Mastercam version X3*

Identify the Operation and Geometry to be Performed As discussed earlier, there are a number of types of tool paths that can be generated for 2D drilling and milling such as drilling, contour milling, pocket milling and face milling.

Drilling For the drilling operation, the location geometry can be specified as points, or the centre point of circles that are present in the geometry. Mastercam can automatically identify the points, or the user can manually pick the points to be used for the drilling operation. Once the points are identified, the software opens the drilling-process-parameter selection window as shown in Fig. 16.58. This window helps in identifying all the necessary parameters for the drilling operation. The first selection is the cutting tool required for the operation, which can be done by selecting any one of the tools available in the library as shown in Fig. 16.59. The tools in the tool library can be updated by adding, removing or modifying the tools present in the library through the tool manager.


C:\McamX3\MATERIALS\DEFAULT.MATERIALS

ALUMINUM inch - 2024
ALUMINUM inch - 5050
ALUMINUM inch - 6061
ALUMINUM inch - 6066
ALUMINUM inch - 7075
ALUMINUM inch - CAST - 65BHN
ALUMINUM inch - WROUGHT - 55BHN
COPPER inch - CAST - 100 BHN
COPPER inch - WROUGHT - 40 BHN
GRAPHITE inch
HASTELLOY inch
INCONEL inch
IRON inch - CAST - DUCTILE - AUSTENSITIC - 160BHN
IRON inch - CAST - DUCTILE - FERRITIC - 165 BHN
IRON inch - CAST - DUCTILE - MARTENSITIC - 300BHN
IRON inch - CAST - MALLEABLE - FERRITIC - 135 BHN

Display options

○ Show all
● Inch
○ Millimeters
○ Meters

Source: Mill - library

Fig. 16.57 *The material selection pop-up window in Mastercam version X3*

Fig. 16.58 *Tool selection window in Mastercam version X3*

Fig. 16.59 *Tool library available in Mastercam version X3*

Fig. 16.60 *Drilling-parameter specification window in Mastercam version X3*

After the required cutting tool is selected, the drilling parameters need to be specified as shown in Fig. 16.60. Mastercam provides 7 pre-defined (canned) drilling cycles and 13 custom cycles. Selecting a drill cycle determines what parameters you can enter for the drill-tool path. You can select a drill cycle from the Cycle drop-down list on the Drill parameters dialog box. For the selected drilling operation, the parameters to be specified are initial height, retract height, feed plane, top of stock, and depth of hole. These are shown graphically in the window (Fig. 16.60) for ease in understanding. When entering clearance, retract, feed plane, top of stock, or depth parameters for a tool path, it is possible to use either absolute or incremental values. The Absolute option uses an absolute value that you enter for the tool-path parameter. The Incremental option uses a value relative to either the current top of the stock, the selected geometry, or the Z depth of each cut. Once all the relevant parameters are specified, Mastercam generates the necessary cldata for the machining of the holes.

Contour Milling Contour tool paths are used to remove material along a path defined by a chain of curves. Any number of chains can be selected for each tool path. Similar to the drilling operation, the parameters to be specified are initial height, retract height, feed plane, top of stock, and maximum depth of the contour. In addition to that, a few other details need to be specified. They are the following:

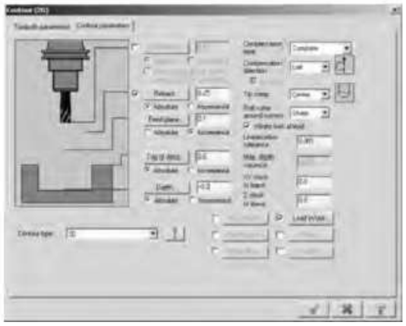


Fig. 16.61 *Contour-milling parameter specification window in Mastercam version X3*

(a) Compensation type The user can specify the type of cutter compensation to be used as follows:

- Computer This means the system will adjust the tool path for the tool size that is selected. This option does not give the machine tool operator the opportunity to adjust for tool wear at the control, and is not generally recommended.
- Control This option calculates the tool path to the geometry with no offset, but simulates compensation in the tool-path display. Mastercam outputs a G41 (left compensation in control) or G42 (right compensation in control) code in the NC program. Compensation in control allows for adjustment at the control using a wear value.
- Wear This option calculates the compensation into the tool path and also outputs a G41 code when the direction parameter is set to left, or G42, when the direction is set to right. Wear allows for a wear offset to be applied at the control instead of a diameter offset. When the Wear option is selected, compensation in computer and control are both enabled in the same direction.

(b) Multiple passes It is possible to select multiple passes which will generate multiple passes offset by the amount specified, as rough machining passes.

(c) Depth of cut User specifies the maximum rough step and Mastercam divides the total depth into equal steps. Alternatively, the user can specify the exact number of finish steps and the size of each finish step.

(d) Lead in/out This refers to the combination of lines and arcs at the beginning and/or end of a 2D contour-tool path finish passes. This option is used to generate a smooth contour with a tangent entry and leaving of the cutter from the workpiece surface. The system places entry and exit lines relative to the entry and exit arcs. If both an entry line and an entry arc are defined, the line gets machined first. If both an exit line and an exit arc are defined, the arc gets machined first.

Once all the options are set appropriately, the system generates the tool path and stores it as an intermediate file similar to a CLDATA file. The total operations that were performed on the part can be seen in the operations manager as shown in Fig. 16.62. From the operations manager, a number of functions can be performed on the tool paths. They are given below.

- **Back plot** Back plotting is a convenient process for displaying the path the tools take to cut the part. This helps in spotting errors in the program before it is actually machined. The tool path is displayed in the current graphics view and the approximate machining time displayed in the prompt area at the bottom of the screen.
- **Verification** This option allows using solid models to actually simulate the material removal process. The stock shape is updated as the tool moves along the tool path and produces the final part. The resulting model can be inspected, ensuring that programming errors are eliminated before they reach the shop floor. An example of solid verification display is shown in Fig. 16.59.
- **Postprocessing** This function converts the CLDATA into an actual NC program to be run in the control. Mastercam first saves the tool-path operations in an intermediate format, a Mastercam NCI file. The postprocessor then reads the NCI file, operation by operation, and creates an NC program.

Pocket Milling Pocket tool paths are used to rough and/or finish closed geometry in parts as shown in Fig. 16.64. All geometry used to define a pocket and any islands must be in the same construction plane.

Face Milling Facing tool paths quickly clean the top of the stock in preparation for further machining. The geometry can be selected as a chain for the tool path or use the stock boundaries from the Job Set-up dialog box.

Fig. 16.62 *The operations manager window in Mastercam version X3*

Fig. 16.63 *Simulation of material removal in Mastercam version X3*

Fig. 16.64 *Simulation of material removal in a pocket-milling operation*

16.9.4 Tool-Path Generation for Turning Centres

Developing the tool path for CNC turning centres in Mastercam is similar to the machining centres, while taking special aspects of turning centres into account.

Job Set-up This option is used to define the stock geometry and chuck geometry (Fig. 16.65) so that the tool does not violate that space and thus avoid tool collisions. The stock boundary defined determines where the stock is in relation to the part. Stock boundary can be defined by chaining existing geometry or by defining the stock from bar stock parameters. When defined properly, Mastercam ensures that the tool never makes a rapid move into the stock. Also, defining stock boundaries is useful to see the graphical simulation of any tool-path operations.

Similarly, it is possible to define the chuck's tool-collision boundaries using the Chuck options on the Boundaries tab in the Job Setup dialog box. The chuck-jaw boundary defined determines where the chuck jaw is in relation to the part. The chuck-jaw boundary can be defined by selecting an existing Mastercam geometry file that represents the chuck, chaining existing geometry or by defining chuck jaws from parameters (Fig. 16.66). Mastercam ensures that the tool will not make a move into the chuck.

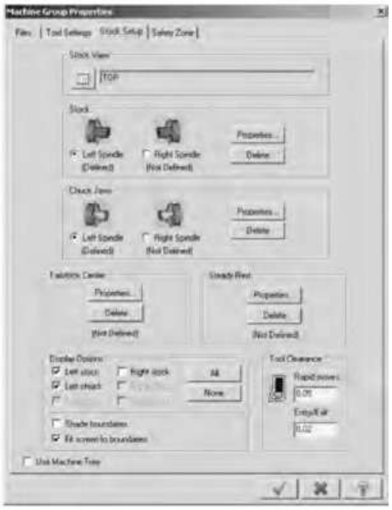


Fig. 16.65 *Job set-up window in Mastercam version X3*

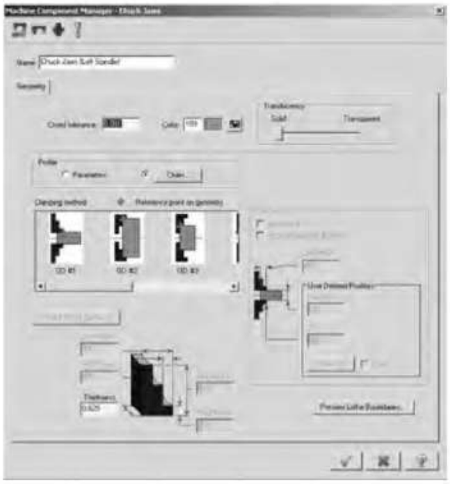


Fig. 16.66 *Chuck specification in Mastercam version X3*

Tool Geometry The complete tool geometry in terms of the tool holder as well as the insert can be completely defined in the tool library for turning centres. Mastercam provides the complete tool catalogues of Sandvik, Kennametal, Valenite and Iscar, from which the user can customise the tool library. It is also possible to define new tools or edit the existing tools from the library. The Lathe Tools dialog box is accessed from the Operations manager from the operation parameters and Def Tools as shown in Fig. 16.67. From Fig. 16.67, select the type of tool that is to be defined, so that all the other screens are customised for that particular option. The tool insert details are specified as shown in Fig. 16.68. Here, the specification is based on the ISO coding system for the carbide inserts. Then the toolholder shape is specified as shown in Fig. 16.69. That completes the geometric specifications of the tool. Based on this information and the details specified in the job set-up, Mastercam will calculate the feed and speed values to be used which can be seen in the parameters window as shown in Fig. 16.70. The user can modify any of these parameters as required.

Fig. 16.67 *Lathe-tool parameter selection window in Mastercam version X3*

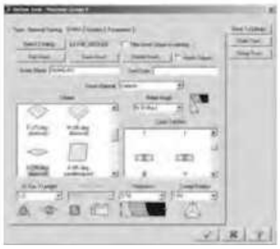


Fig. 16.68 *Lathe-tool insert specifications in Mastercam version X3*

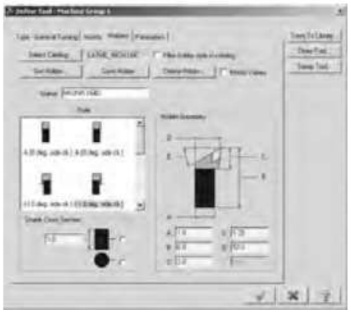


Fig. 16.69 *Lathe-toolholder specifications in Mastercam version X3*

The various tool paths available for turning centres are

- Facing
- Roughing (external and internal turning)
- Finishing (external and internal turning)
- Groove
- Canned
- Quick
- Thread

Fig. 16.70 *Lathe-tool parameters specifications in Mastercam version X3*

For creating the geometry for turning centres, care has to be taken to see that the coordinate system is properly defined. Since the lathe part is symmetrical, it is necessary only to create half the part in two axes (*X* and *Z*). The Cplane (construction plane) used for turning centres is a flat, two-dimensional plane on which the geometry is created. The Cplane settings +*DZ* and –*DZ* are used for constructing geometry with the *Z* axis and a diameter value (the *D* value represents the diameter of *X*). The centreline forms the *Z* axis. The Cplane settings +*XZ* and –*XZ* are used when the *X*-axis specifies the radial distances. The origin of the part is normally at the front face and centreline of the part.

Facing-tool Path Face tool paths are used to cut the material on the face of the part. Selecting the face-tool paths option from the menu opens the facing tool parameters window as shown in Fig. 16.71. After selecting the appropriate tool and its parameters as shown in Fig. 16.71, the face parameters tab is to be clicked. This opens the face parameters window through which all the necessary parameters for the facing operation are entered as shown in Fig. 16.72.

Roughing Tool Paths Roughing tool paths are used to coarsely cut the part geometry to be followed by a finish tool path. The geometry of the roughing path can be selected normally by means of chaining similar to the milling tool paths. Once selected, the rough parameters dialog box opens as shown in Fig. 16.73. The selection of tool is similar to the facing. The roughing parameters dialog box is shown in Fig. 16.73. The options to be selected are given below.

(a) Overlap The Rough Overlap Parameters is used to specify the amount by which the tool overlaps the previous cut before making the next cut.

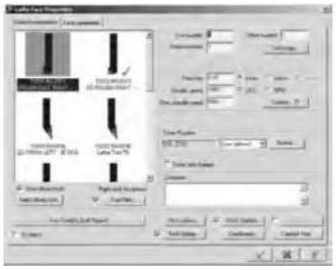


Fig. 16.71 *Tool parameter selection for facing operation in Mastercam version X3*

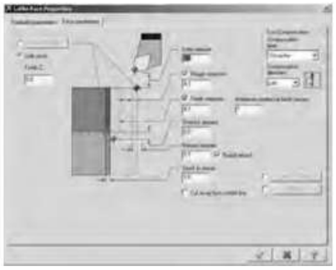


Fig. 16.72 *Facing parameter specifications in Mastercam version X3*

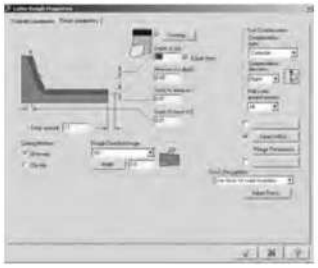


Fig. 16.73 *Rough turning parameter specifications in Mastercam version X3*

(b) Cutting method The options available are either one-way cutting or zigzag. The zigzag option is to be used for neutral insert, otherwise one-way is to be used for either left- or right-hand tools.

(c) Roughing direction/Angle of cut The roughing angle is the angle at which the tool cuts into the part, and is relative to the rough direction.

Plunge Parameters This option allows to specify the method in which the undercutting is to be performed by the tool while it does the rough machining operation. The options available are (Fig. 16.74) given below

Fig. 16.74 *Plunge cutting parameter specifications in Mastercam version X3*

(a) No Plunging Allowed The tool skips over any undercuts in the chained path. In such a case, a separate tool path to cut the undercut is to be defined.

(b) Allow Plunging In Both Directions The tool cuts into all undercuts in the chained path. Activates the Tool Width Compensation parameters.

(c) Allow Plunging in Relief The tool cuts into undercuts on the side cutting edge of the insert.

(d) Allow Plunging in Undercut The tool cuts into undercuts on the end cutting edge of the insert. Activates the Tool Width Compensation parameters.

Finish Tool Paths Finish tool paths are used to follow the part geometry in order to make the final cuts on the part and clean the rough surface generated during the roughing tool path. The process is similar to the roughing tool paths. Additional parameters that need to be specified are (Fig. 16.75) given below.

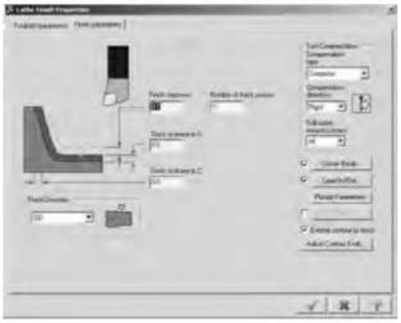


Fig. 16.75 *Finish turning parameter specifications in Mastercam version X3*

(a) Lead In/Out These parameters help to control the direction in which the tool approaches the part at the start of each pass in the tool path. Using these parameters eliminates the need to create extra geometry for this purpose. This can be done (Fig. 16.76) by

- Extending/shortening the geometry in the chained contour
- Adding a line to the start of the chained contour
- Creating a tangent arc move to the start of the tool path
- Manually defining an entry vector, or letting the system calculate an entry vector

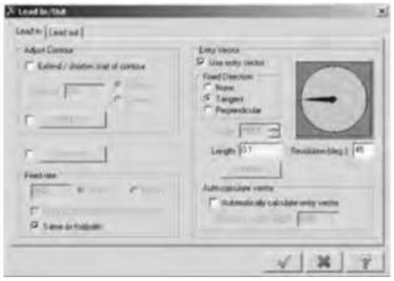


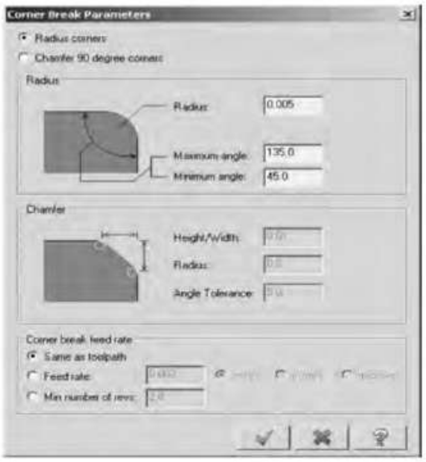
Fig. 16.76 *Lead in/out parameter specifications in Mastercam version X3*

(b) The Corner Break Parameters (Fig. 16.77) allow for the creation of radii or chamfers on all outer corners of the tool path, as well as determination of the feed rate when the tool creates the radii or chamfers. While creating a radius on the outer corners, the angle of the original corner must fall within the maximum and minimum range that is entered. When creating a chamfered corner, the angle of the outer corner(s) must equal 90 degrees. If not, compensation needs to be added by adjusting the angle tolerance. The angle tolerance determines how far the corner can vary from 90 degrees and still have a chamfer added.

Groove Tool Paths A groove is a long, narrow cut or indentation in a surface. Rectangular grooves can be machined by defining a single point, two points, or three lines for groove geometry. It is also possible to create non-rectangular grooves by defining an inner and outer boundary.

Canned Tool Paths Mastercam's canned tool path options make it easy to use the CNC machine controller's canned cycle programs. In this case, only the parameters that are required for the CNC machine's canned cycles are entered. Mastercam writes these parameters to the NCI file. Once the NCI file is converted by the postprocessor into the appropriate NC format, the controller generates the tool path using the parameters entered in the canned tool path dialog box. The type of parameters depends upon the individual controller's capability.

Quick Tool Paths These are used for generating a tool path with very little input. These are used for uncomplicated tool-path creation. Quick tool paths allows to create rough, finish, and groove operations for basic jobs where only a few essential parameters to complete the job need to be specified.



● Radius corners

○ Chamfer 90 degree corners

Radius

Radius: 0.005

Maximum angle: 135.0

Minimum angle: 45.0

Chamfer

Height/Width:

Radius:

Angle Tolerance:

Corner break feed rate

● Same as toolpath

○ Feed rate:

○ Min number of revs:

Fig. 16.77 *Corner break parameter specifications in Mastercam version X3*

Thread Tool Paths Thread tool paths shape and determine how the tool cuts the thread. The parameters to be specified depend upon the type of thread to be created such as ID, OD, or Face/Back. A thread form determines the shape of the thread, and different thread forms are used for different applications. Mastercam supplies a thread table so you can choose from standard combinations of common thread forms. These combinations include common basic major diameter and lead values. There are 11 different thread forms supplied with the software.

Summary

- The use of computer-aided part-programming methods for more complex shapes used in industrial practice is required. Initially, the need started with aerospace applications, but has now been universally applied to all the manufacturing industries. The use of computer-aided programming systems reduces the part-program development time, while increasing its reliability to optimise the machining resource utilisation.
- APT (Automatically Programmed Tools) is the first computer aided programming system that was developed to help in the programming of complex shapes used in aerospace applications. However, its use quickly moved to other applications because of its usefulness for all the industries.
- To make the computer-aided part-programming systems universally applicable to all types of machine tools, the processing is done in two stages. The processor does the various arithmetic calculations required to generate the geometric tool path of the cutter that is independent of the machine-tool type. A postprocessor then converts that information to machine-specific codes to be used for running the CNC machine tool.
- APT geometry has facilities for defining a large number of geometric elements directly from the information available in the part drawing.
- APT motion statements will allow for simple English-like statements to generate the programs for point-to-point applications as well as contouring for complex profiles.
- The complete APT part program has in addition to geometry and motion components, compilation control and postprocessor statements.

- Mastercam is more user-friendly and has a geometric user interface for the user to see the development of the part program.
- The geometry of the part can be easily constructed by the variety of menu commands available from the data available for the part drawing.
- The tool-path generating modules utilise the geometry and the built-in cutting-tool database to generate the optimum tool path which can be verified before finalising the part program.
- Tool-path modules are available for machining centres as well as turning centres to take care of the differences between the types of machining methods employed.

Questions

1. Explain the concept and need for a postprocessor as used in computer-assisted part-programming systems such as APT. Describe the functions of a postprocessor.
2. What are the basic assumptions made while programming in the APT language?

Problems

1. For the following components (Fig. 16.78 to 16.83), develop the part programs using the APT language.
2. For the following components (Fig. 16.84 to 16.90), develop the part programs using any of the CAM software available in your laboratory.

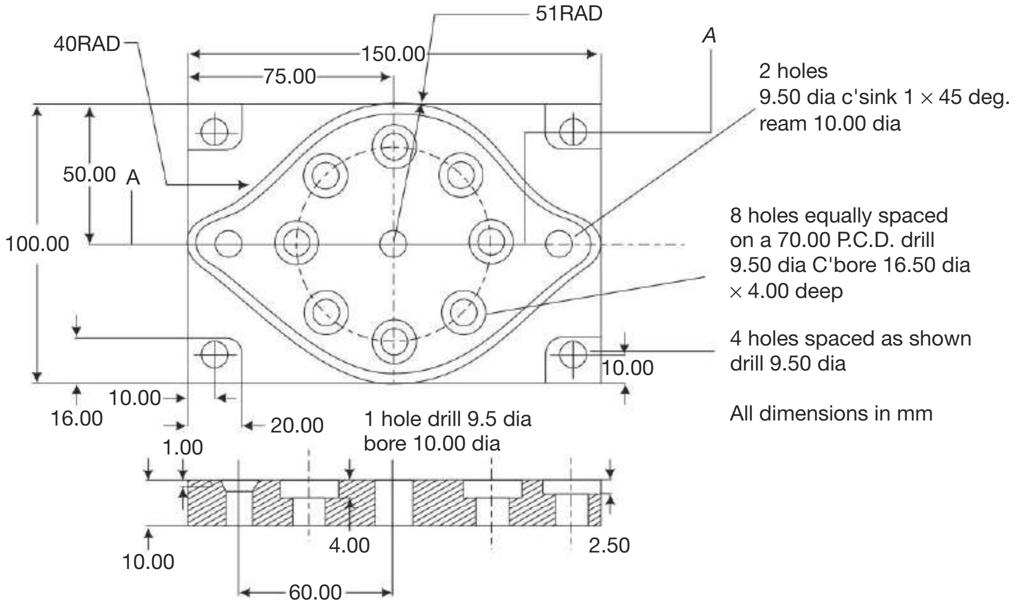
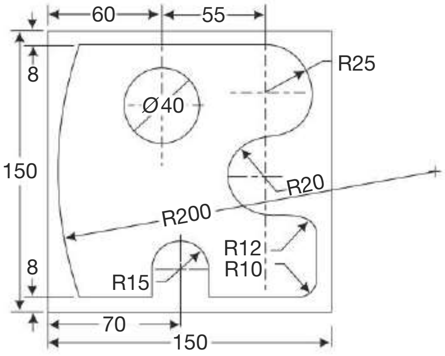


Fig. 16.78



Tool: slot drill Ø16 mm
milling depth: 6 mm

Fig. 16.79

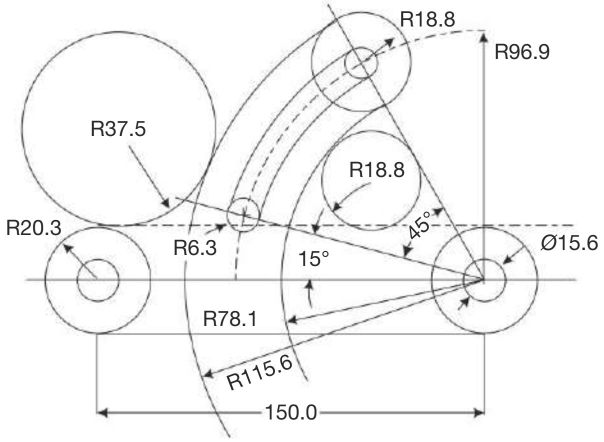


Fig. 16.80

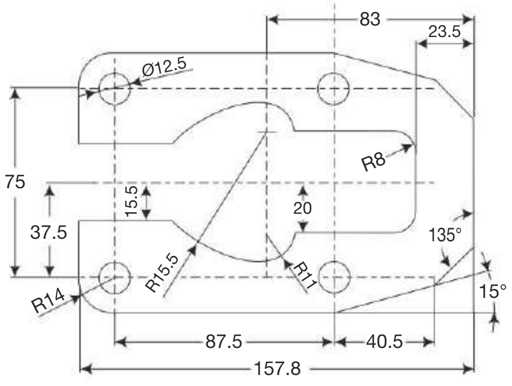


Fig. 16.81

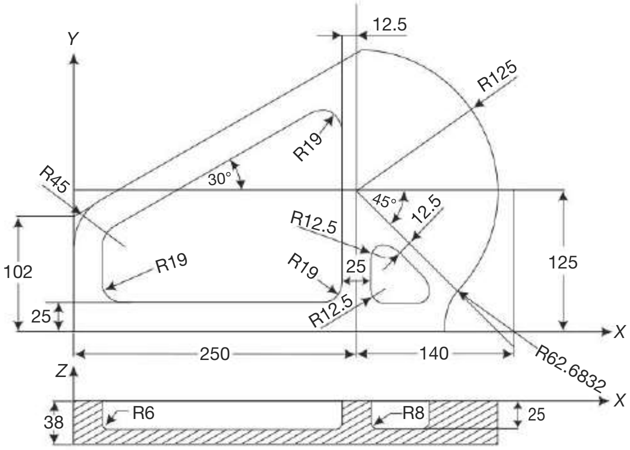


Fig. 16.82

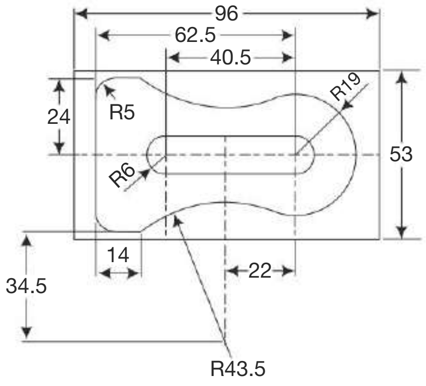


Fig. 16.83

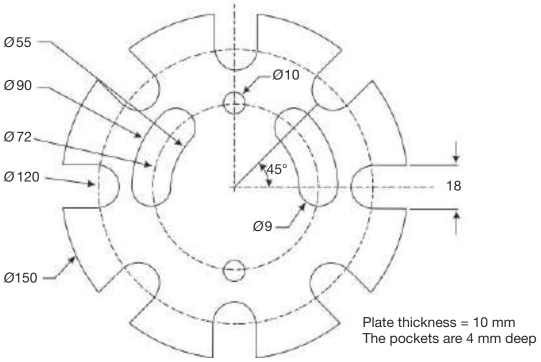


Plate thickness = 10 mm
The pockets are 4 mm deep

Fig. 16.84

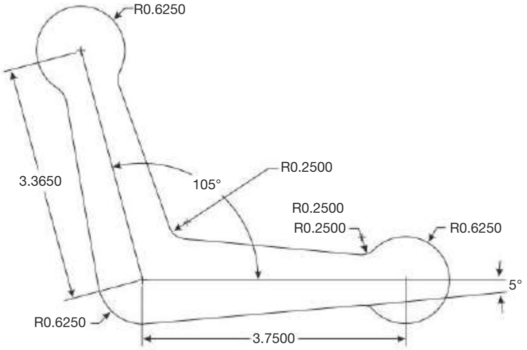


Fig. 16.85

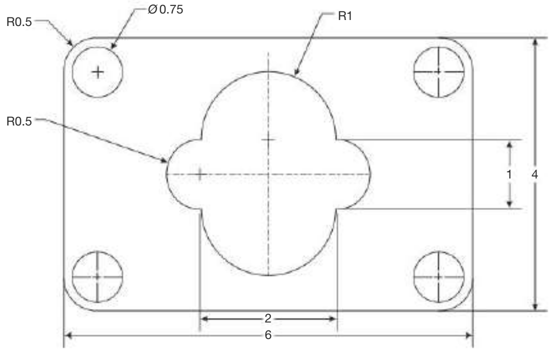


Plate thickness 0.75 in
Pocket depth 0.5 in
Tool 1: Drill 0.75 dia
Tool 2: Endmill 1 dia
Tool 3: Endmill 0.375 dia

Mill Exercise 01

Fig. 16.86

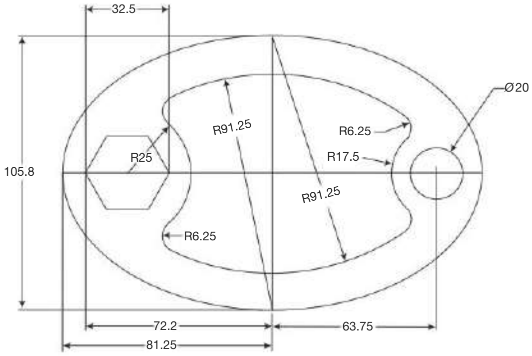


Fig. 16.87

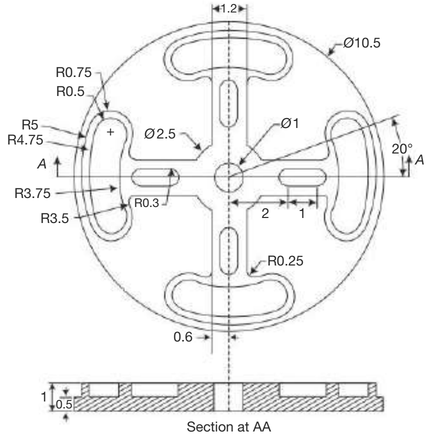


Fig. 16.88

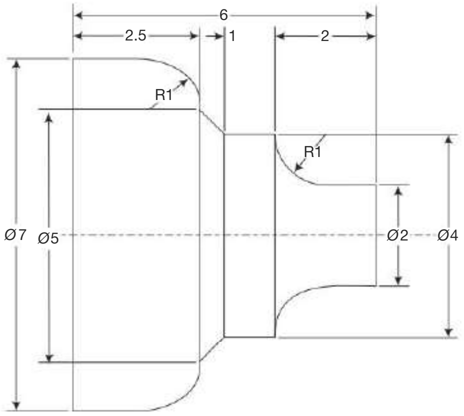


Fig. 16.89

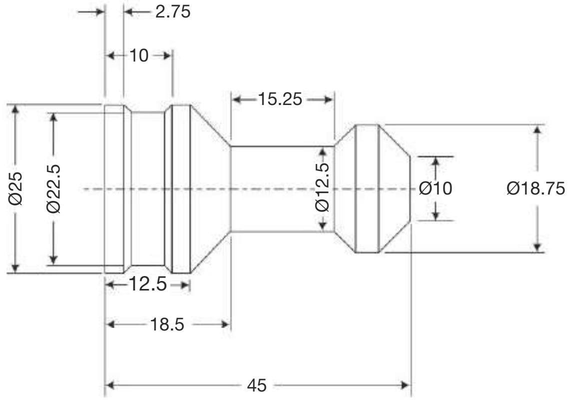


Fig. 16.90

Part – IV

ROLE OF INFORMATION SYSTEMS

17

INFORMATION REQUIREMENTS OF MANUFACTURING

Objectives

Parts II and III of this book dealt specifically with the design and manufacturing aspects of industrial products. In order to successfully produce the products for the market, it is required that the overall manufacturing activity be controlled in such a way that the manufacturing becomes cost-effective. In order to have a proper control of the overall production process, it is necessary to have a complete understanding of the information that needs to be made available at various stages of manufacturing.

For the successful running of an enterprise, the availability of the right type of information at the right time and place will be a major requirement. Hence, understanding of the information requirement in a manufacturing enterprise forms the basis of this chapter. After completing the study of this chapter, the reader should be able to

- Understand the needs of discrete part manufacturing
- Organise the information into various modules within the total manufacturing operations
- Understand the manufacturing strategies and the need for integration

17.1 DISCRETE PART MANUFACTURE

The product cycle with and without the use of computers is discussed in Chapter 1. In that context, the various categories of manufacturing activities in terms of mass, batch and job-shop production were also discussed. Though that discussion is centred around the conventional manufacturing methods, the same is generally true with respect to the use of advanced manufacturing technologies as well. The actual manufacture can be broadly classified into the following two types.

- Continuous production or process manufacturing
- Discrete part manufacturing

Continuous production or process manufacture involve producing products through a continuous line with a single end product coming from the line, whereas the discrete part manufacturing involves the products which require many components which need to be produced separately and then assembled.

The sequence of operations involved in the production of a given product are the following.

- Design
- Planning
- Manufacturing
- Assembly

The assembly stage may involve the sub-assemblies as well as testing. In the conventional manufacturing each of these stages proceed sequentially as shown in Fig. 17.1. Thus it can be seen that the lead time for a given product in such a case is the total time taken for the individual stages.

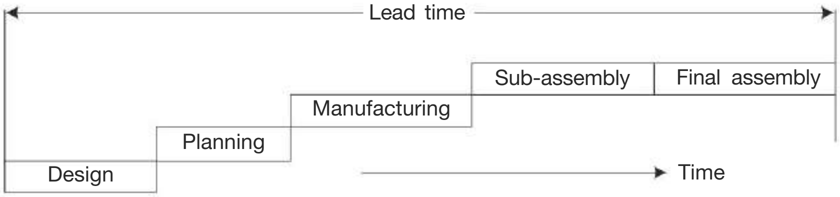


Fig. 17.1 *Various stages involved in the manufacturing of a product*

However, in a concurrent engineering approach some of these operations are done in parallel as shown in Fig. 17.2 such that the total lead time can be reduced. It may not be possible for some operations to be carried out simultaneously, e.g., testing of the full product can be done only after completing the assembly. However, the sub-assemblies can be tested individually for their functions. The concurrent engineering can be achieved by having a close cooperation between all the participating sections of the industry.

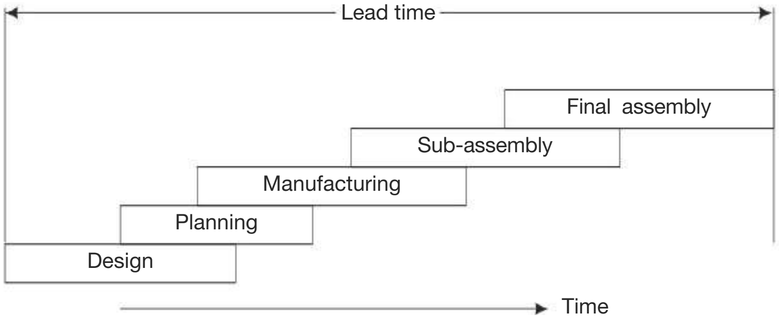


Fig. 17.2 *Various stages involved in the manufacturing of a product in concurrent Engineering Environment*

17.2 INFORMATION REQUIREMENTS OF A PRODUCTION ORGANISATON

The information requirements of a manufacturing organisation are varied and it needs to study the same to understand the requirements of its integration. In this book, we are interested in the manufacturing activity alone and therefore the other aspects will be ignored. A clear understanding of the terminology used will help in understanding their role in successful running of a manufacturing organisation. A typical set of components that can be found in a production information arrangement are shown in Fig. 17.3.

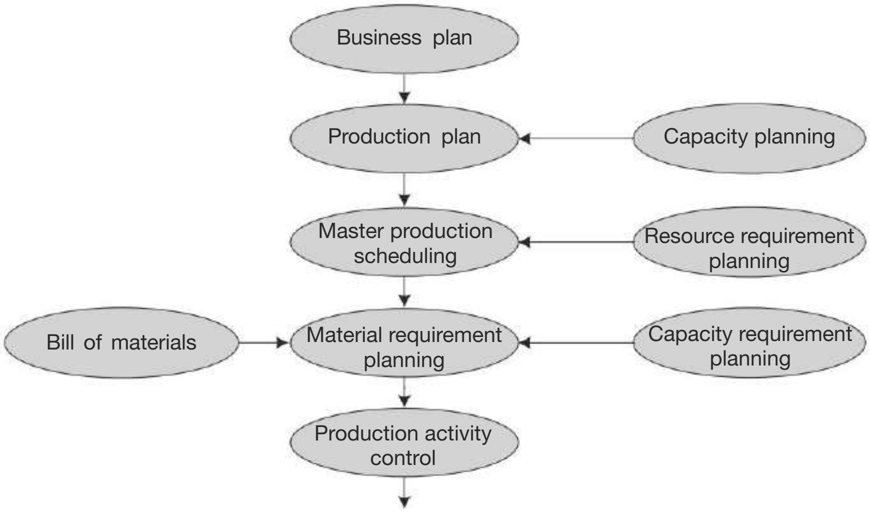


Fig. 17.3 *Components of total operation planning system for manufacturing organisations*

Business Plan It is the statement of the organisation at the highest level in terms of what business activity the company would like to carryout in the next plan period (typically one year). It is normally expressed in gross terms in monetary value (e.g. turnover to be Rs 300 crores in the next financial year). The gross value may also be expressed in terms of individual product groups (product A, B or C) with monthly or quarterly turnover. The business plan is calculated based on the forecast of demand for the products. This becomes the input for the lower order planning departments as shown in Fig. 17.3 to see how it can be executed.

Production Planning This develops the production part of the business plan. It tries to convert the business plan into a more concrete form in terms of the various product groups to be produced in the different facilities and divisions in the plan period. It gives the

- Quantities of each product group
- Desired inventory levels
- Resources required

The aggregate output may be broken into smaller time slots, such as weekly or monthly instead of the gross period of a quarter or an year as used in the business plan. Also many of the production variants may be suppressed to make the problem simpler.

Capacity Planning The desired production plan is meaningful only if there is capacity or means within the plant to realise it. The capacity planning therefore tries to balance the production with capacity at aggregate level. Sometimes it is also called aggregate capacity planning. The actual production capacity available within may be augmented by the addition of temporary workers, additional shifts, overtime payments or subcontracting. Thus the capacity planning will be able to identify the capacity constraints and specify the necessary adjustments needed to achieve the required production.

Master Production Scheduling (MPS) This is the most detailed form of the production plan wherein all the individual parts will be disaggregated into their component level. This is the point where all other departments in the information and planning systems will be able to link with. For example, the linking of actual sales orders with corresponding production timing as well as when these should be delivered. This may also have to take into account the backlog in orders which needs to be fulfilled.

Resource Requirement Planning It is also called *rough cut capacity planning* and is done in conjunction with the master production schedule. This in a way is actually validating the master production schedule. This will be able to show clearly that any of the manufacturing resource (department, work centre or machine) is not overloaded. Though this can be done for all the work centres within the plant, sometimes only the critical areas that are likely to be the bottlenecks may be considered in case of large establishments.

Material Requirements Planning (MRP) MRP allows for the time-phased requirements for releasing the materials or receiving the material in order to fulfill the requirements laid down in the MPS. This is a very important element and is discussed in greater detail in Chapter 19.

Capacity Requirement Planning This is also called *detailed capacity planning*, and needs to be worked in conjunction with the MRP. This will be able to identify at the lowest level, what is the actual capacity required to implement the MPS.

Production Activity Control (PAC) This is also called Shop floor control, this activity controls the actual production activity on a daily basis. The total activity involves the loading of individual machines and work centers with the parts, sequencing of the parts on individual machines in terms of the start times of each job and coordinating the flow of materials in the shop floor. The schedules generated may sometimes have to be changed because of the last minute amendments. The amendments may be in terms of priority changes, machine breakdowns, worker absenteeism, etc.

As noted above, the level of detail and the planning horizon are different for various modules in the information system of a manufacturing organisation. A qualitative representation of these is shown in Fig. 17.4.

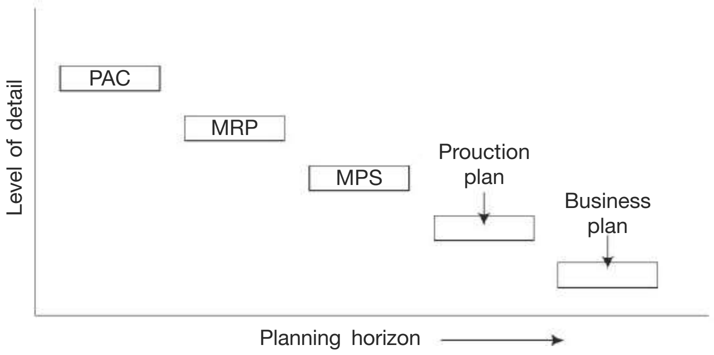


Fig 17.4 *Level of detail versus planning horizon for various modules in the information system of a manufacturing organisation*

17.3 MANUFACTURING STRATEGIES

Many of today's manufacturing companies have to deal with the following aspects.

- Greatly reduced product lifecycles
- High product varieties
- Unpredictable demand patterns
- High customer's lead time
- Large numbers of components with high lead times and usage rate

These requirements are part of the challenge to the companies to gain competitive advantage. The challenge of gaining competitive advantage is made more difficult due to the uncertainty of the market environment. The approach to managing this properly is effective materials and manufacturing resources management and control. This involves the management of variety, flexibility of suppliers and a responsive Master Production Scheduling (MPS) system. Manufacturing is moving away from making products for stocking purposes and is becoming increasingly customer's driven in its approach. This involves some rethinking as regards the production management system (PMS). There are four classic types of manufacturing environments. These are the following.

- Make to stock (MTS)
- Assemble to order (ATO)
- Make to order (MTO)
- Engineer to order (ETO)

MTS categories the manufacturing of products based on a well known and relatively predictable demand mix. The MTS assembly has the advantage of having quick delivery time, but inventory costs are large and customers are unable to express preferences to product design. Further, it is categorised by reasonably long and predictable product life cycles. Thus MTS is followed when

- Demand is fairly constant and predictable
- There are few product varieties
- Delivery times demanded are shorter than the product manufacturing time
- Product has a long shelf life

ATO involves having the same core assemblies for most products and the ability to vary all other components of the final assembly. The delivery time is of medium length and is based on the availability of major sub assemblies. Assembly takes place on receipt of an order and buffers of modules or options may exist. No final inventory buffer exists and customer has limited choice in the design of the product. The typical example is the personal computers.

MTO involves having all the components available along with engineering designs but the product is actually specified. Manufacturing of the product begins with the receipt of an order and the configuration of the product is likely to change from the initial specification during the course of processing. Interaction with the clients is extensive. The delivery time ranges from medium to long. Thus make to order is followed when

- Products are made as per the customer specification
- Inventory holding costs are high
- A large number of product options are made available
- Customer is willing to wait

ETO is an extension of MTO system with the engineering design of the product being done almost totally on the basis of customer's specifications and customer's interaction is greater than in the case of MTO.

The manufacturing strategies and their effect on lead time is depicted in Fig. 17.5.

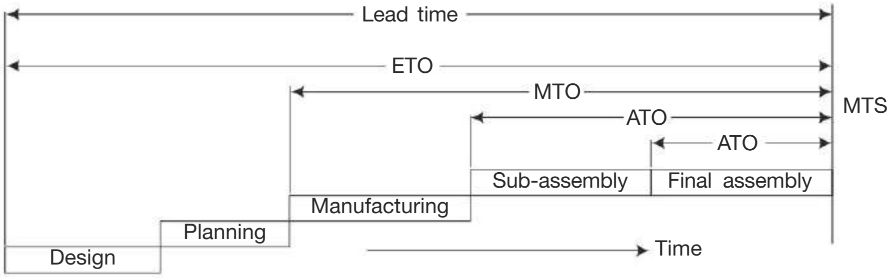


Fig. 17.5 *Manufacturing strategies and their effect on the lead time*

17.4 INTEGRATION REQUIREMENTS

As can be seen the various elements of the information systems require that a large amount of information is required in order to correctly assess the situation and come to the correct decision. All the data is available in the form of databases in the various modules in addition to the centralized databases. Most of this information may be based on past experience or assumptions. Unfortunately in a realistic situation the incorrect information will lead to sub optimal or faulty decisoins.

Therefore it is necessary that many of these modules in the information system of a manufacturing organisation need to provide inputs to other modules in such a way that the decisions are taken based on actual position rather than the assumptions. It therefore becomes necessary to link all these modules at the database level. A typical structure of information flow for a manufacturing organization tion is shown in Fig. 17.6.

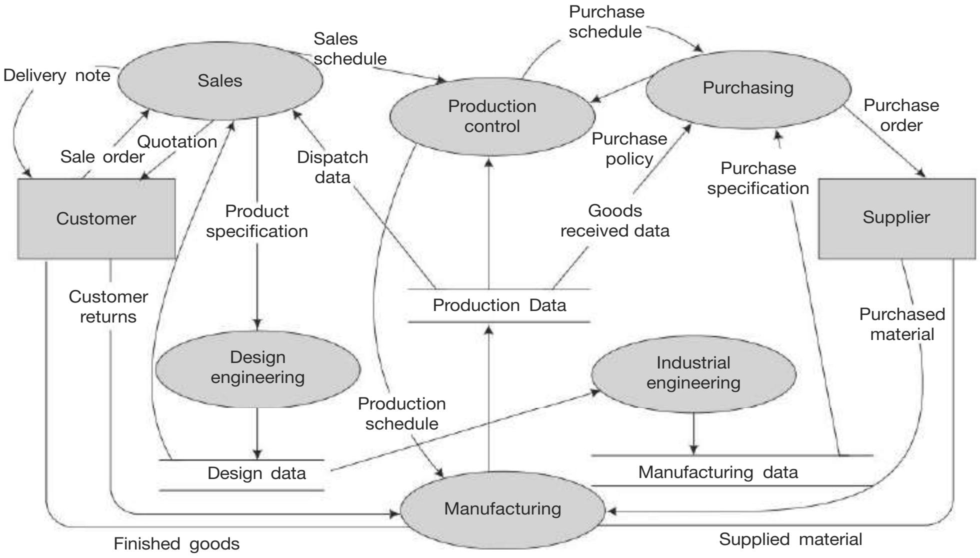


Fig. 17.6 *Typical information flow in a manufacturing organisation*

Summary

- In order to maintain proper control of manufacturing operations, there is need to peruse a lot of information. The manufacturing industries therefore need to organise and channel this information properly, so that there is efficient operation of the various components in the production operations.
- Discrete part manufacture involves the design, planning, manufacturing and assembly as the components parts. The way these are organised help in reducing the lead times for total operation.
- The information requirements of a manufacturing organisation can be identified into modules depending upon their functionality. Some of the modules are production planning, capacity planning, material requirement planning, etc.
- Manufacturing strategies refer to the strategies adopted depending upon the type of product in question. These are make to stock, assemble to order, make to order and engineer to order.
- Integration of various modules of information is important to gain the synergies in operation.

Questions

1. What are the different classes of manufacturing? Give examples for each.
2. Define lead time. Explain how it changes with the concurrent engineering practice.
3. Broadly speaking, what are the various components present in the information system for production planning of an industrial organisation.
4. Briefly write about production planning process in discrete part manufacturing.
5. What do you understand by the following?
    (a) Master production scheduling
    (b) Resource requirements planning

   How are they related?
6. What are the differences between capacity planning and resource requirement planning? Explain the function of each.
7. What are the various manufacturing strategies used by product manufacturers? Explain their application.
8. Briefly write about the importance of integration in manufacturing information systems.

18

GROUP TECHNOLOGY AND COMPUTER AIDED PROCESS PLANNING

Objectives

Converting the part CAD model into a product requires a number of operations that need to be identified and organised systematically so that proper control can be established. This chapter will explain some of the aspects that deal with this part of identifying and organising the manufacturing processes for the parts. After completing the study of this chapter, the reader should be able to

- Understand the need for Group Technology (GT) as a means of bringing the benefits of mass production to the relatively smaller production that is required in a majority of the present-day manufacturing industries
- Learn about the coding and classification methods and schemes used in manufacturing
- Use production-flow analysis as a method of applying GT for manufacturing applications
- Learn about the different types of flow possible in manufacturing cells
- Understand the methods used to design GT cells
- Appreciate the need for Computer Aided Process Planning (CAPP)
- Understand the different approaches used in CAPP application
- Learn details about the techniques utilised in developing CAPP systems

18.1 GROUP TECHNOLOGY

The increasing market competition worldwide has dramatically increased the number and variety of products and parts for today's industries. Among the traditional manufacturing systems, mass production accounts for only 30% of all manufacturing. The remaining 70% of manufacturing is accomplished in job shops where it is done on relatively general-purpose machine tools arranged normally in functional layout. This situation leads to uneconomic and inefficient production and the search for methods to improve the productivity of traditional intermittent production industry has motivated various technological improvements.

Group Technology (GT) is the most important technological improvement reported for the batch-processing industries. The GT approach is based on the principle of sameness which refers to the grouping of parts to be manufactured according to similarities derived using various characteristics and processing them on their requisite machines placed close together in a cell. The concept of Cellular Manufacturing System (CMS) has been reported to have resulted in the largest manufacturing productivity increases. Since its evolution about half a century ago, GT principles have caught attention of many researchers and practitioners.

18.1.1 Introduction

The term *group technology* was first used by Professor Mitrofanov of Leningrad University in the early 1950s, but the genesis of group technology can be traced back to the work carried out in the USSR and Germany, to extend formal standardisation. With the publication of the first work by Prof. Mitrofanov in 1952, the fundamentals of group technology emerged with the development from the single-family machine approach to multiple families on multi-product lines and moreover, the introduction of group methods into the forming sectors of manufacture.

Group Technology (GT) is a manufacturing philosophy which can be used to group parts based on similarities in design or manufacturing process so as to reduce the overall manufacturing cost. By grouping the parts, small batches of different parts can be produced as a large batch, thereby bringing in the advantages of mass manufacture to batch manufacturing. In addition, there are many other advantages which are discussed later.

To be grouped into a single group, the parts need not have similar features—in fact even different looking parts having the same production process can also be classified into a single group. Looking at Fig. 18.1, it can be noticed that in (a) all the parts have similarities in shape, geometry and dimensions. However, in the case of (b), the part geometries are quite different, but the processing methods followed are similar and hence they are grouped together.

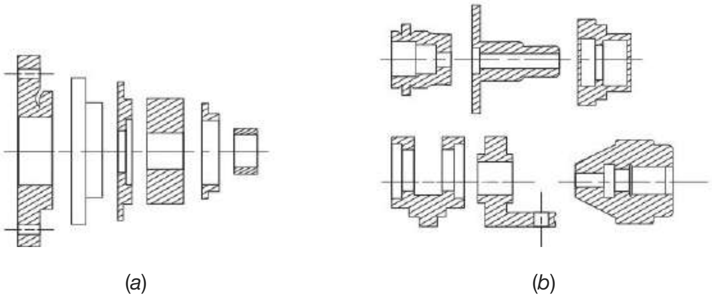


Fig. 18.1 *Examples of part families that can be grouped by geometry or processing methods*

To understand the philosophy better, consider the plant layouts that are generally followed. The two most common plant layouts used are process layout and product layout. For small batch and job-shop manufacture, generally process layout is followed. In the case of *process layout,* all the machine tools of the same process are grouped in a single department and placed together as shown in Fig. 18.2. However, with such an arrangement it can be noticed that the parts need to move through various departments to complete the task. This calls for a fairly large amount of travel for the part and is wasteful.

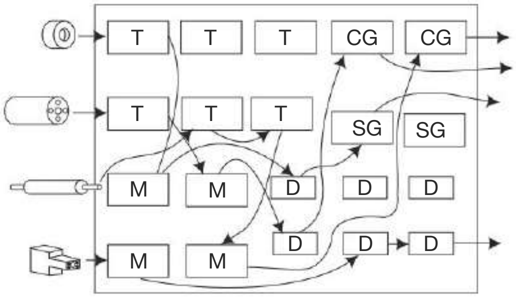


Fig. 18.2 *Layout of machines with process grouping in a conventional job shop, T - Turning, M - Milling, D - Drilling, CG - Cylindrical grinding and SG - Surface grinding*

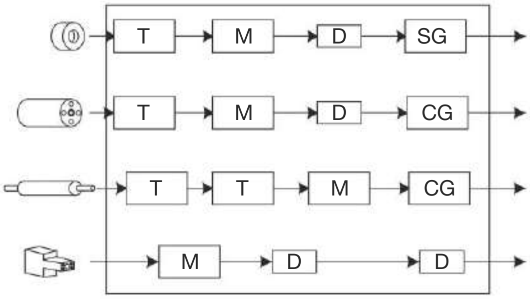


Fig. 18.3 *Layout of machines with functional grouping as per product requirements*

In the case of *product layout,* the machines are arranged in the sequence in which the operations are to be carried out, such that the part movement is smooth. The relaying out of the facility in Fig. 18.2 is done in Fig. 18.3 for product form. It can be noticed that the parts don't have to travel a lot in the shop since the next machine tool is always placed by the side. However, this type of layout can be justified only when there are enough number of parts to be done on these groups of machine tools so that they are occupied all the time. When the production is of small or medium batch, it is difficult to fully load the machine tools and that makes the machine tools used in that particular line to be under-utilised. However, by grouping the parts together into a family, it is possible to make the batch sizes large by combining the production requirement of the individual parts together.

The concept can be easily explained with the help of a composite part concept. Let us imagine that there are a number of components as shown in Fig. 18.4a which all have similar machining operations to be done. Then it is possible to construct a new composite part that has all the features identified in the parts from Fig. 18.4a and shown in Fig. 18.4b. It is then possible to develop the optimised process plan for the composite part, which can then be tailored for the individual parts by removing the processes for the features that are not present in the particular part.

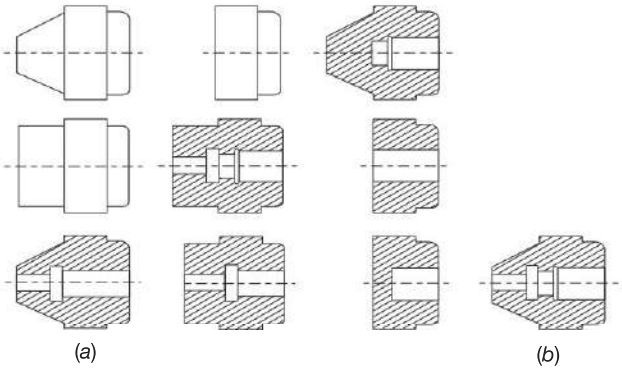


Fig. 18.4 *Composite part and its possible variants*

18.1.2 Advantages and Limitations of GT

It has been found that GT has a large number of advantages depending upon the way it is implemented and based on the type of industry in use. In fact, most of the flexible manufacturing systems that will be studied in later chapters rely heavily on the GT concepts. Some of the advantages are outlined below.

1. Group technology allows similar designs to be easily modified from the existing designs from the database instead of starting from scratch.
2. Standard process plans can be developed for the groups. Greater efforts can be applied in optimising the process plans.
3. Standard tooling can be developed for a part family, and then a standard a set-up procedure can be used.
4. The use of GT allows faster production, therefore there is less inventory, and Work in Process (WIP).
5. The throughput time gets reduced.
6. Material handling and movement is reduced.
7. Improvement in quality and reduction in scrap results in increase in production.
8. There is improved utilisation of machines, and as a result lesser number of machines are required. This increases the floor space available.
9. There is increase in output per employee and hence increase in productivity.
10. Manufacturing lead time is reduced leading to reduction in overdue orders.
11. There is improved ability to respond to market changes.
12. Increase in job satisfaction and greater management–worker harmony results.

Though there are a large number of advantages claimed for GT, there are as many problems that need to be considered before implementing GT. They are the following:

1. The cost of implementation is generally high with an outside consultant often being necessary since in-house expertise on GT is rarely available. It requires a long set-up time and painful debugging.
2. It may not be suitable for a factory with a very large variety of products.
3. The entire production of the company cannot be put under GT and hence GT will have to coexist with conventional layouts.
4. There are too many GT codes in use and there is no one GT code that suits all applications.
5. It is often difficult to conceive all the operations for a group of components being taken care of in the cell created for it.
6. The range of product mix in a plant may be under constant change in which case, the GT cells may need constant revision, which is impractical.

18.1.3 Part-Family Formation

In order to develop the part families in group technology, the most common methods adopted are

- Visual inspection
- Classification and coding
- Production flow analysis

Visual Inspection Visual inspection is the least sophisticated and least expensive method. The classification of parts into families is done by visually inspecting either the physical parts or their drawings and arranging them into groups having similar features from either the design point of view or the processing point of view. For this purpose, shop-floor experience is an added asset for the planning engineers. The part families thus arrived at will form the basis for work cell. This method is fast and simple and is useful when the part mix is not complex.

18.2 CLASSIFICATION AND CODING

Classification is the process of separating the parts into groups or families depending on the characteristic attributes based on a set of rules or principles. *Coding* is the process of providing a symbol to the component. These symbols should have meanings that reflect the attributes of the part, thereby facilitating further analysis. In order to arrive at the coding scheme for the parts, it is necessary first to complete the study of all the features present in the total part spectrum. There are a large variety of coding systems is use, each of which are developed for specific applications. No system has not yet received universal acceptance. The reason for this is that the information that is to be represented in the classification and coding system varies from one company to another. Also, the type of code to be used depends upon the application that is being envisaged for it. Though there are commercial systems available for coding and classification, they need to be further customised for the company to take care of the individual requirements of the company and the purpose for which the code is being developed.

The functions served by classification and coding systems are the following:

- **Identifying and filing data for retrieval purpose** The system should be able to retrieve the requisite information in a timely manner.
- **Forming a basis for inference and prediction about group members** The coding system should be able to predict certain coexistent properties of a family based on the likenesses and differences of the attributes used to form the family. This capability is based on the scientific theory of the system itself. The predictive worth of the classification and coding system depends on the homogeneity of the group formed, based on the clear and concise selection of attributes that eliminates arbitrary and random groupings.
- **Providing explanatory information about the family groupings** There should be a clear rationale for the joint attributes of items within a family, for the formation of groups, and for the hierarchy of the attributes that are classified.

18.2.1 Types of Coding Systems

There are a large number of coding systems that were developed and in use. All these systems can be broadly classified into three groups:

- Hierarchical, or monocode
- Attribute, or polycode
- Hybrid, or mixed

Hierarchical or Monocode The monocode comes closest to matching a decision tree. In this type of code, the meaning of each character is dependent on the meaning of the previous character, i.e., each subsequent character amplifies the information of the previous character. The structure of such a code is shown in

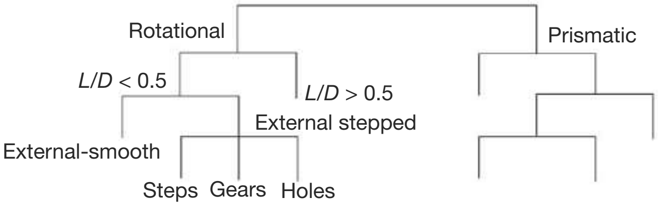


Fig. 18.5 *Sample monocode*

Fig. 18.5. The monocode can be used to rapidly subdivide a population into small groups with relative ease. However, the meaning of any particular digit in the code is difficult to determine.

This type of code is particularly preferred in design departments for part retrieval because this type of system is very effective for capturing shape, material, and size information. However, in manufacturing departments, the information needs are based on processing requirements. For this purpose, a hierarchical structure is much useful.

Attribute or Polycode The polycode is not structured like logic trees. The meaning of each character in an attribute code is independent of all other digits. Each digit is used to completely to classify some feature of the item. An example of an attribute code is given in Table 18.1. As can be seen, each digit of a polycode has a significance of its own and can be used to extract a definition of the item from the database independent of any other digit.

Table 18.1 *A sample polycode*

Digit	*Feature*	*Possible values*			
		1	*2*	*3*	*4*
1	External shape	Shape 1	Shape 2	Shape 3	Shape 4
2	Number of holes	0	1 – 3	4 – 6	> 6
3	Type of hole	Axial	Cross	Axial/cross	Other
4	Gear teeth	Internal spur	External spur	Helical	Worm
5	Splines	—	—	—	—

In polycode, the features are usually listed and queried in order from the most important to the least important without any logic as can be seen from Table 18.1. Since there is no logical transition between digits, to describe every conceivable item in the population in detail becomes difficult. To cover all aspects of the population, polycodes can become quite long and the coding can be very tedious. Retrieval of information becomes easy. For example, to retrieve all spur gears, identify all parts with a ‘2’ in the position 5 of the associated code. As a result, the attribute code system is popular with manufacturing organisations since it makes it easy to identify parts that have similar features that require similar processing.

Hybrid, or Mixed Code Both the above codes cannot fully serve the functions of a manufacturing organisation. To take advantage of both the coding systems, most codes that are used in industry are neither monocode

Table 18.2 *Some general classification and coding systems developed (Gallagher and Knight, 1986).*

Coding System	*Developing Country*	*Number of Digits*
VUOSO	Czechoslovakia	4
VUSTE	Czechoslovakia	4
Brisch	UK	4 – 6
KC1	Japan	5
Part Analog	USA	4 – 6
IAMA	Yugoslavia	8
Opitz	West Germany	9
PGM	Sweden	10
CODE	USA	8
Pittler	West Germany	9
Gildemeister	West Germany	10
Toyoda	Japan	10
MICLASS	The Netherlands	12
TEKLA	Norway	12
NIITMASH	Russia	15
ZAFO	West Germany	21+
VPTI	Russia	Variable
KK3	Japan	20
DCLASS	USA	8

nor polycode, but are a hybrid of the two. Initially, one or two digits form the monocode part to divide the population into small groups, which is followed by a polycode series of attributes that have significance to the group in the branch. A hybrid code is relatively more compact than a pure attribute code while retaining the ability to easily identify parts with specific characteristics.

Some general classification and coding systems developed by various countries is listed in Table 18.2. This is not a complete list, but only a representative sample to give some idea of the different systems attempted in the past. A few of the coding systems that are widely used are described in greater detail here.

18.2.2 Opitz Coding System

The most common and widely used coding system is that developed by Professor H Opitz of Aachen Technical University, Germany. It is a 9-digit code with the first 5 forming the primary code while the last four, the secondary code. The concept is shown in Fig. 18.6.

The first five digits are used as given in Table 18.3.

The classification proceeds further with the different first digit values of 3 to 9. The full details can be seen in the book by Opitz. As can be seen, each digit has a meaning which needs to be ascertained from the coding system.

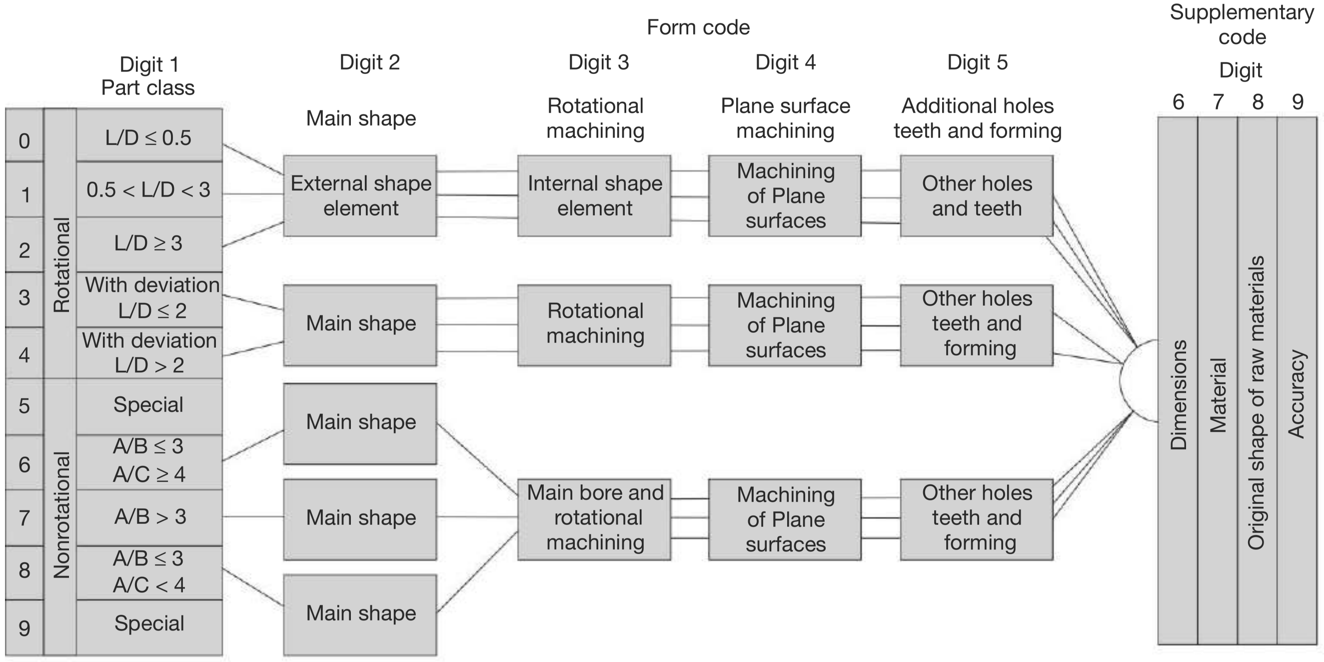


Fig. 18.6 *Opitz coding system*

Table 18.3 *Opitz coding system*

	Digit 1		*Digit 2*		*Digit 3*	
		Part class	*External shape, external shape element*		*Internal shape, internal shape elements*	
0	R o	$L/D \leq 0.5$		Smooth, no shape element		No hole no break through
1	t a	$0.5 < L/D < 3$ $L/D \geq 3$	Smooth or Stepped to one end	No shape elements	Smooth or Stepped to one end	No shape elements
2	t i			Thread		Thread
3	o n			Functional grooves		Functional grooves
4	a		Stepped both ends	No shape elements	Stepped both ends	No shape elements
5	l			Thread		Thread
6	N o			Functional grooves		Functional grooves
7	n r			Functional cone		Functional cone
8	o			Operating thread		Operating thread
9	t			All others		All others

Table 18.3 *Opitz coding system* *(contd.)*

	Digit 4	*Digit 5*	
	Plane surface machining	*Auxiliary holes and gear teeth*	
0	No surface machining	No gear teeth	No auxiliary hole
1	Surface plane and/or curved in one direction, external		Axial, not on pitch circle diameter
2	External plane surface related by graduation around a circle		Axial, on pitch circle diameter
3	External groove and/or slot		Radial, not on pitch circle diameter
4	External spline (polygon)		Axial and/or radial and/or other direction
5	External plane surface and/or slot, external spline		Axial and/or radial on PCD and/or other directions
6	Internal plane surface and/or slot	With gear teeth	Spur gear teeth
7	Internal spline (polygon)		Bevel gear teeth
8	Internal and external polygon, groove and/or slot		Other gear teeth
9	All others		All others

Example 18.1 Develop the Opitz form code (first 5 digits) for the component given in Fig. 18.7.

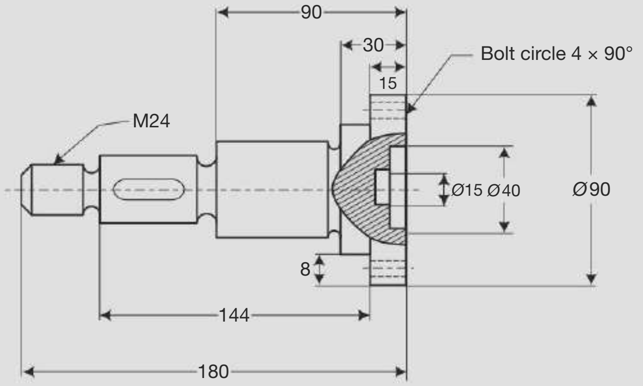
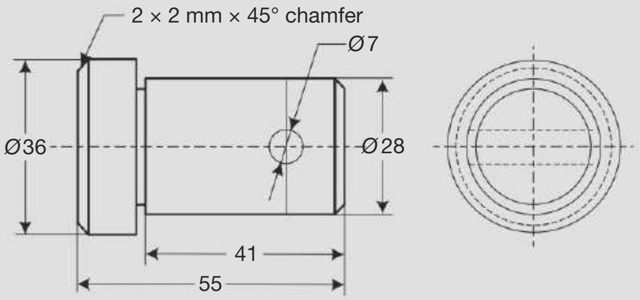


Fig. 18.7 *Part for coding Example 18.1*

Solution

1	*2*	*1*	*3*	*2*
Part class: Rotational part, *L*/*D* = 2	External shape: Stepped to one end, thread	Internal shape: Stepped to one end, no shape element	Surface machining: External groove	Auxiliary holes: Axial on pitch circle diameter

Example 18.2 Develop the Opitz form code (first 5 digits) for the component given in Fig. 18.8.



All dimensions in mm

Fig. 18.8 *Part for coding Example 18.2*

Solution

1	*3*	*0*	*3*	*3*
Part class: Rotational part, *L*/*D* = 1.52	External shape: Stepped to one end, smooth, no shape elements	Internal shape: No hole	Surface machining: External groove	Auxiliary holes: Radial, not on pitch-circle diameter

18.2.3 VUOSO–PRAHA Coding System

It is a 4-digit code developed by VUOSO. Each of these 4 digits characterise the part by kind, class, group and material. Brief details of the code system are given in Table 18.4.

Table 18.4 *Simplified part of Vuoso–Praha coding system for kind and class*

		KIND								
		Rotational workpieces					*Flat and irregular*		*Boxlike*	*Other mainly non-machined*
		Hole in axis			*Geared and splined: Hole in axis*					
		None	*Blind*	*Through*	*Blind*	*Through*				
		1	*2*	*3*	*4*	*5*	*6*		*7*	*8*
		D	*L/D*	*Rough form*			*Rough form*	L_{max}, *mm*	*Rough weight*	*Made of*
C	0	0–40	< 1				Gib like L/B > 5	0 – 200	0—0 kg	Extruded forms
L	1		1 – 6					200+	30–200 kg	Bars
A	2		> 6				Platforms	0-200	200–500 kg	Tubes
S	3	40–80	< 1					200+	500–1000 kg	Sheets
S	4		1–4				Lever like	0-200	1000+ kg	Wires
	5		> 4					200+		
	6	80–200	> 3				Irregular	0-200		
	7	80	> 3					200+		
	8	200	> 3				Prism like	0-200		
	9	Various	> 30					200+		

18.2.4 MICLASS Coding System

The MICLASS (Metal Institute Classification) was developed by the Organisation for Applied Scientific Research in the Netherlands (TNO) in the 1960s and 1970s to develop a system for both design and manufacturing needs for OIR (Organisation for Industrial Research). The various functions MICLASS was developed for are

- Standardise engineering drawings
- Retrieve drawings based on classification
- Standardise process routing
- Automate process planning
- Select parts to be processed on a group of machine tools

MICLASS is an expandable hybrid code system of up to 30 digits, while the first twelve digits have been standardised. These digits relate to shape, form, dimensions, tolerances, and materials, as shown below:

1	2	3	4	5	6	7	8	9	10	11	12
Basic Form				Primary Dimensions				Tolerances		Material	

The system can be enlarged to thirty digits to cover any classification attribute desired by the user. The following is an example of such an extension.

13	14	15	16	17	18	19	20	21	22	23	24
Lot size		Secondary dimension		General manufacturing operations		Supplementary design and manufacturing information					

Computer software is provided by OIR for deriving the part code, after the user goes through a series of questions and answers them interactively. The built-in logic is in the form of decision trees.

18.2.5 KK-3 Coding System

This is a code developed by the Japan Society for the Promotion of the Machine Industry (JSPMI) and was presented first in 1976 [Chang, Wysk and Wang].

Table 18.5 *KK-3 coding system structure for rotational components [Chang, Wysk and Wang]*

Digit	*Items (Rotational components)*		
1	Parts name		General classification
2			Detail classification
3	Materials		General classification
4			Detail classification
5	Chief dimensions		Length
6			Diameter
7	Primary shapes and length diameter ratio		
8	Shape details and kinds of processes	External surface	External surface and outer primary shape
9			Concentric screw threaded parts
10			Functional cut-off parts
11			Extraordinary shaped parts
12			Forming
13			Cylindrical surface
14		Internal surface	Internal primary shape
15			Internal curved surface
16			Internal flat surface and cylindrical surface
17		End surface	
18		Non concentric holes	Regularly located holes
19			Special holes
20		Non-cutting process	
21	Accuracy		

18.2.6 DCLASS Coding System

DCLASS (Design and Classification Information System) was developed in the Computer-Aided Manufacturing Laboratory of Brigham Young University in 1976. Although its primary use to date has been in the university environment, many companies are using it for prototype development. It is not a fixed code classification and coding system, but uses a computer software system to rapidly and efficiently traverse decision tree logic similar to MICLASS.

The DCLASS part family code is comprised of eight digits partitioned into five code segments, as shown below:

1	2	3	4	5	6	7	8
Basic shape			Form features	Size	Precision	Material	

Three digits that form the first set are used to denote the basic shape. The form features code which is one digit in length is entered in the next segment. It is used to specify the complexity of the part such as holes and slots, heat treatment and special surface finishes and is determined by the number of special features. The third segment which is one-digit-long specifies the overall size of the part. Precision of the part is indicated in the fourth segment as a single digit. The last two digits of the code specify the material type. An example of DCLASS code for the part shown in Fig. 18.9 is shown below:

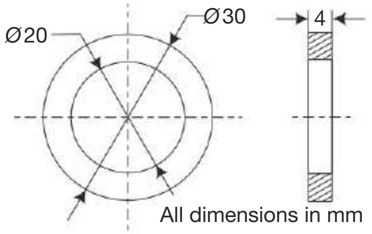


Fig. 18.9 *Part for coding by DCLASS*

B	0	1	1	2	3	A	7
Basic shape: Round with single outside diameter and single bore diameter			Form features	Size: maximum dimension ≤ 50 mm	Precision, no special processing	Material: Stainless steel	

18.2.7 CODE MDSI System

Manufacturing Data Systems, Incorporated (MDSI) has developed this classification and coding system called CODE. It is an eight-digit hybrid code used primarily to classify and code mechanical piece parts. The typical code structure is shown below:

1	2	3	4	5	6	7	8
Major division	Outer diameter or section	Centre hole	Holes (other than centre hole)	Grooves, threads	Miscellaneous	Maximum outer diameter, or section across flats	Maximum overall length

Each of these digits uses hexadecimal characters (0 to 9, and A through F).

18.3 PRODUCTION FLOW ANALYSIS

One of the main advantages of using group technology is that the machine tools can be reorganised into cells to follow the operation sequence closely. This ensures that the parts are grouped based on the production requirements, and only those machines that are required for the group manufacture are in the cell, thereby reducing the material-handling requirements for the group. Since all the parts in the group have similar machining processes, the cutting tools and fixtures required can be rationalised which reduces the set-up times for the parts, bringing down the overall machining cost. The main requirement for this is that all the parts of a group should have similar routings.

This can be accomplished by a process called *production-flow analysis* or PFA, a procedure developed by Burbidge. Production-flow analysis helps in identifying the part families as well as grouping the machine cells. In order to carry out the PFA, it is necessary to collect all the information related to the parts and their processing to be made as part of the system.

Data Collection The main data required to perform PFA is the information about the route taken by the part through the shop to complete all the processing operations. This is normally present in a sheet called *route sheet* or *process sheet* or some similar name in the industrial engineering (planning) department. Normally, in this sheet, each processing operation to be performed is associated with a machine tool. From the route sheet, the part number and machine routing (operation sequence) for every part can be obtained. Additional information that is useful is lot size, time standards, annual demand, etc., which could be used to identify the capacity required in terms of cells and number of machines. A typical route sheet is given in Table 18.6.

Table 18.6 *Typical route sheet*

Op. No.	*Operation*	*Description*
01	Lathe01	Chuck, face one end, centred rill, drill, ream, straight turn, chamfer, cut-off
02	Milling03	Mill the keyway
03	Inspection05	Inspect dimensions and finish

Sorting of Process Routing The data collected during the first step needs to be grouped in this step so as to achieve similarity in the process routings. To help with this step, all the processing operations are coded into numbers or characters (e.g., lathe–01 or A; mill–02 or B; etc.). Using these codes, each part is listed as a series of codes representing the sequence in which these operations are to be carried out. Then a sorting procedure is applied on these routings such that parts with identical routes are grouped together. The actual number of parts in a particular group may vary from one (unique part) to many.

PFA Chart The route sheets of all the parts are then organised in the form of a matrix as shown in Fig. 18.7. This chart is for a total of six parts, all of which need to be processed on six different machines. This chart is called the *part machine incidence matrix*. In this matrix, the entry of '1' represents where the part needs to visit that particular machine. When no visitation is required, a '0' is placed or for convenience in reading, it is left blank.

Table 18.7 *PFA Chart, original and rearranged*

Part → *Machine ↓*	*1*	*2*	*3*	*4*	*5*	*6*
A	1		1		1	
B		1		1		1
C		1		1		1
D	1		1		1	
E		1		1		1
F	1		1		1	

Part → *Machine ↓*	*1*	*5*	*3*	*6*	*2*	*4*
A	1	1	1			
D	1	1	1			
F	1	1	1			
C				1	1	1
B				1	1	1
E				1	1	1

While forming the matrix, care has to be taken to see that grouping is possible. For example, if an operation is required by only one part, there is no need to include it in the group-formation process. Similarly, if all or most of the parts require a particular operation then also it should not be included. For example, painting or heat-treatment operations are generally required by many parts and hence are arranged separately and not in any group.

Analysis From the raw part machine incidence matrix, the columns and rows are rearranged in such a way that all the ones come into one grouping as shown in the right-hand side matrix of Table 18.7. Each such block can be considered as a machine cell (Cell1 = machine 1, 4, 6; Cell1 = machine 3, 2, 5). This being a hypothetical case, the cells were easily formed. However, in real life, there are many problems which need to be properly addressed by the user.

A point that should be noted here is that PFA utilises the existing parts, process plans and routings which might have been prepared by different planners at different times, and as such may not be optimal. It allows a process layout of the plant to be converted into a more efficient product layout with the current data.

Once the incidence matrix is organised, there are a number of methods available for making the groups. A few of these methods are described below:

18.3.1 Rank-Order Clustering

King [1980] presented a simple algorithm for converting the part machine incidence matrix into a nearly block diagonal structure. A block diagonal matrix is one where boxes on the main diagonal contain ‘0’s and ‘1’s, while the off-diagonal boxes contain all ’0’s. In this procedure, the rows and columns are considered as binary strings, left to right for rows, and top to bottom for columns. The decimal equivalent is calculated for each row and column. Considering rows, if two are similar then their binary number value should be the same. Now, all the rows are ranked in decreasing order. Next, the procedure is repeated for the columns. See if the row order has changed. If so, continue the procedure until a time when there is no change in the order occurs, which gives the solution. An example will help clarify the procedure.

Example 18.3 Obtain the part families for the incidence matrix given in Table 18.8 using rank-order clustering.

Table 18.8 *Initial incidence matrix for Example 18.3*

Parts → Machines ↓	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
A	1		1	1	1			1	1	
B				1						
C	1		1		1			1	1	
D		1								1
E		1								
F						1	1			1
G	1									
H										1

Solution

Step 1 Assign 8 to the top row and the corresponding binary positions. Their values are shown in the two extreme right columns of the incidence matrix in Table 18.9. Based on these, the binary values of columns are calculated. Based on the value, the ranks of the columns are shown.

Table 18.9 *Incidence matrix with column ranks calculated for Example 18.3*

Parts → *Machines ↓*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*W_j*	*Value*
A	1		1	1	1			1	1		2^7	128
B				1							2^6	64
C	1		1		1			1	1		2^5	32
D		1								1	2^4	16
E		1									2^3	8
F						1	1			1	2^2	4
G	1										2^1	2
H										1	2^0	1
W_j	162	24	160	192	162	4	4	160	160	21		
Rank	2	7	4	1	3	9	10	5	6	8		

Step 2 Based on the ranks calculated in the previous step, rearrange the columns in the descending order from left to right as shown in Table 18.10. Assign 10 to the leftmost column and the corresponding binary positions and their values are shown in the two bottom-most rows of the incidence matrix shown in Table 18.10. Based on these, the binary values of rows are calculated. Based on the value, the ranks of the rows are shown.

Table 18.10 *Incidence matrix for Example 18.3 after Step 2*

Parts → *Machines ↓*	*4*	*1*	*5*	*3*	*8*	*9*	*2*	*10*	*6*	*7*	*W_j*	*Rank*
A	1	1	1	1	1	1					1008	1
B	1										512	2
C		1	1	1	1	1					496	3
D							1	1			12	5
E							1				8	6
F								1	1	1	7	7
G		1									256	4
H								1			4	8
W_j	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
Value	512	256	128	64	32	16	8	4	2	1		

Step 3 Based on the rank calculated in Step 2, the rows are rearranged and shown in Table 18.11. Since the rows are rearranged, the ranks of the columns are calculated as in Step 1.

Table 18.11 *Incidence matrix for Example 18.3 after Step 3*

Parts → *Machines ↓*	*4*	*1*	*5*	*3*	*8*	*9*	*2*	*10*	*6*	*7*	W_j	*Value*
A	1	1	1	1	1	1					2^7	128
B	1										2^6	64
C		1	1	1	1	1					2^5	32
G		1									2^4	16
D							1	1			2^3	8
E							1				2^2	4
F								1	1	1	2^1	2
H								1			2^0	1
W_j	192	176	160	160	160	160	176	14	8	8		
Rank	1	2	4	5	6	7	3	8	9	10		

Step 4 Based on the rank calculated in Step 3, the columns are rearranged and shown in Table 18.12. Since the columns are rearranged, the ranks of the rows are calculated as in Step 2. Since the row ranks are not altered, this is the final incidence matrix, and the two cells formed are shown in Table 18.12.

Table 18.12 *Final incidence matrix for Example 18.3*

Parts → *Machines ↓*	*4*	*1*	*5*	*3*	*8*	*9*	*2*	*10*	*6*	*7*	W_j	*Value*
A	1	1	1	1	1	1					1008	1
B	1										512	2
C		1	1	1	1	1					496	3
G		1									256	4
D							1	1			12	5
E							1				8	6
F								1	1	1	7	7
H								1			4	8
W_j	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
Value	512	256	128	64	32	16	8	4	2	1		

18.3.2 Direct Clustering Method

An improvement of the rank-order clustering was given by King and Nakornchai [1982], and Chan and Milner [1982]. In this method, the cells in the incidence matrix are termed positive (those containing 1) and negative (those containing 0 or blank). Proceed sequentially through the matrix and move the columns with the topmost negative cells to the left and rows with leftmost positive cells to the top. As this shifting takes place, the machine and the part should move together. Repeated application of this procedure makes all the ones come close to the diagonal, thus forming the machine blocks. A typical incidence matrix for 10 different parts with a total of 10 different operations is shown in Table 18.7 which is utilised to demonstrate the direct clustering method.

Example 18.4 Obtain the part families for the incidence matrix given in Table 18.13 using the direct clustering method.

Table 18.13 *The initial incidence matrix*

	Part 1	*Part 2*	*Part 3*	*Part 4*	*Part 5*	*Part 6*	*Part 7*	*Part 8*	*Part 9*	*Part 10*
Oper 1		1		1		1	1			1
Oper 2		1								1
Oper 3	1		1						1	
Oper 4		1								1
Oper 5				1		1	1			
Oper 6	1							1	1	
Oper 7	1							1		
Oper 8	1		1		1				1	
Oper 9	1				1				1	
Oper 10						1	1			

Solution

Step 1 Calculate the total sum of each row and column numbers as shown in Table 18.14.

Table 18.14 *The incidence matrix after Step 1*

	Part 1	*Part 2*	*Part 3*	*Part 4*	*Part 5*	*Part 6*	*Part 7*	*Part 8*	*Part 9*	*Part 10*	
Oper 1		1		1		1	1			1	5
Oper 2		1								1	2
Oper 3	1		1						1		3
Oper 4		1								1	2
Oper 5				1		1	1				3
Oper 6	1							1	1		3
Oper 7	1							1			2
Oper 8	1		1		1				1		4
Oper 9	1				1				1		3
Oper 10						1	1				2
	5	3	2	2	2	3	3	2	4	3	

Step 2 Sort the rows and columns in the descending order as shown in Table 18.15.

Table 18.15 *The incidence matrix after Step 2*

	Part 3	*Part 4*	*Part 5*	*Part 8*	*Part 2*	*Part 6*	*Part 7*	*Part 10*	*Part 9*	*Part 1*	
Oper 1		1			1	1	1	1			5
Oper 8	1		1						1	1	4
Oper 3	1								1	1	3
Oper 5		1				1	1				3
Oper 6				1					1	1	3
Oper 9			1						1	1	3
Oper 2					1			1			2
Oper 4					1			1			2
Oper 7				1						1	2
Oper 10						1	1				2
	2	2	2	2	3	3	3	3	4	5	

Step 3 For each of the row (from top to bottom), move all the columns which have a positive value to the right, maintaining the order of the previous rows. An example is shown in Table 18.16 for the row 1.

Table 18.16 *The incidence matrix after Step 3*

	Part 3	*Part 5*	*Part 8*	*Part 9*	*Part 1*	*Part 4*	*Part 2*	*Part 6*	*Part 7*	*Part 10*	
Oper 1						1	1	1	1	1	5
Oper 8	1	1		1	1						4
Oper 3	1			1	1						3
Oper 5						1		1	1		3
Oper 6			1	1	1						3
Oper 9		1		1	1						3
Oper 2							1			1	2
Oper 4							1			1	2
Oper 7			1		1						2
Oper 10								1	1		2
	2	2	2	4	5	2	3	3	3	3	

Step 4 For each of the columns (from right to left), move all the rows which have a positive value to the top, maintaining the order of the previous columns. An example is shown in Table 18.17.

Table 18.17 *The incidence matrix after Step 4*

CELL 1

	Part 3	*Part 5*	*Part 8*	*Part 9*	*Part 1*	*Part 4*	*Part 2*	*Part 6*	*Part 7*	*Part 10*
Oper 1						1	1	1	1	1
Oper 5						1		1	1	
Oper 2							1			1
Oper 4							1			1
Oper 10								1	1	
Oper 7			1		1					
Oper 9		1		1	1					
Oper 6			1	1	1					
Oper 3	1			1	1					
Oper 8	1	1		1	1					

CELL 2

Step 5 If the current matrix is same as the previous matrix, stop or else go to Step 3.

The above procedure is amenable for computer coding. The example taken above is relatively simple and hence a solution is obtained in a very small number of steps. In actual problems, the matrix size is very large and the final solution may require a number of iterations as shown in Step 5. Further, if there are some bottleneck machines, the solution gets halted, in which case user intervention is required to proceed with the solution.

The Production Flow Analysis (PFA) is a systematic procedure for finding the families and is used by many companies, but it involves subjective judgements and becomes cumbersome for large numbers of components and/or machines. Even though the methods were simple, logical and efficient, the limitation of these procedures is that the incidence matrix considers only the information on the relationship between the parts and the machines. It doesn't consider the unique processing time and demand requirements of each part on different machines but implicitly assumes uniform load. As a result, the production cell formed may not be balanced. If the block diagonal form is not obtained, resolution of the exceptional parts is required.

A few reasons why PFA has succeeded as given by Burbidge [1993] are the following:

1. PFA starts with Factory Flow Analysis (FFA) that forms departments (major groups), which complete all the parts they make, before attempting to form groups and families.
2. PFA only plans the change from process organisation GT. It doesn't consider any changes in plant, product design, or processing methods, or any sub-optimisations such as cost minimisation, for example. Some of these may be desirable, but they are best left to new projects after GT. This follows the principle of restricting change to bite-sized chunks.
3. PFA does not accept the total machine/part matrix as found. It modifies it, before using it to find groups, into a module/machine matrix, based on a ranking of the machines, to give precedence to those that can only be in one group.
4. PFA never expects to find a pure division into groups with the existing process routes. It accepts that it will normally be necessary and possible to reallocate some processing operations from one machine to another, and it includes a formal step for doing this in Group Analysis (GA).

18.4 CELLULAR MANUFACTURING

Once parts have been grouped into families, and the machines that are required for complete processing have been identified, it is necessary to organise these machines into a cell. A *cell* is where all the parts of the group are completely manufactured and these machines are organised so that material and parts smoothly move through the cell. Manufacturing with GT cells is called *cellular manufacturing*.

The proximity of workstations, their limited number and the product similarities make it easy to schedule the jobs, provided the part mix remains the same. When the part mix changes, cellular-manufacturing system tends to be much less efficient because of the imbalance in machine loading. Also, it may not be possible for a GT cell to be entirely independent. Sometimes, parts must visit more than a single GT cell since it is judged too expensive to duplicate machines simply to increase GT cell independence. The main purpose served in a GT cell is that a product layout for the part spectrum encompassed in the part family is achieved with the requisite number of machine tools and processing stations arranged in proper layout.

18.4.1 Machining Cell Designs

Important consideration for the design of a machining cell is the part and material flow through the cell. The materials and semi-finished parts flow in only one direction through the cell in the order in which they are to be performed as per the process planning.

The basic cell-flow patterns that can be present in any GT cell are

- Straight through cell or inline layout
- U-shape cell
- L-shape cell

Straight Through Cell or Inline Layout As shown in Fig. 18.10, the machines are arranged in a line and generally integrated with an inexpensive, mechanised work-handling system, such as a conveyor. This type of layout is easy to understand, control, and schedule the parts and labour operating the machines. It has easy access from both sides, and avoids congestion of the point of delivery for the finished parts from the cell.

U-shape Cell The more common arrangement found in GT cells is the U-shape cell as shown in Fig. 18.11. In this arrangement, the work-in and work-out points is the same allowing for convenient handling for integrating with other elements of the factory. All the workers are inside and as such can help each other when necessary. When the number of machines in the cell is small, it may be provided with only manual

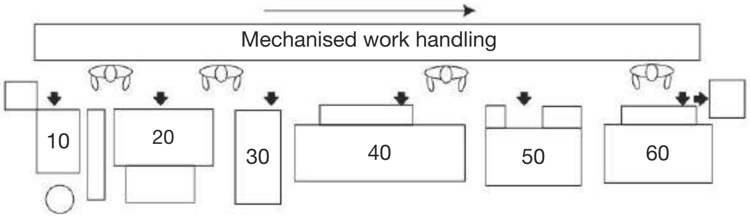


Fig. 18.10 *Arrangement of a straight through cell*

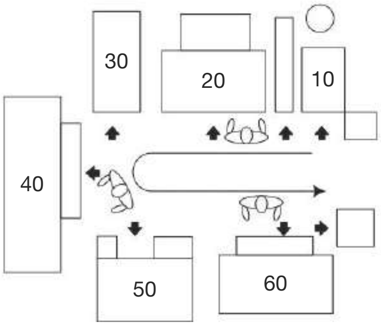


Fig. 18.11 *Arrangement of a U-shape cell*

work-handling arrangement, while with large number of machines, it can be provided with mechanised work-handling system. Since the machines are in close proximity to the workers, it is possible to assign multiple machines to an individual worker, thereby improving the productivity. It is easy for line balancing.

L-shape This type of arrangement, shown in Fig. 18.12, allows for fitting lengthy series of operations into a limited space. It can be conveniently located at the points in the shop that are convenient from the raw material to come to the cell as well as feed other cells with the finished parts. In view of the segregation of inflow and outflow of material from the cell, it is easy to organise the supplies such as materials, products or special services.

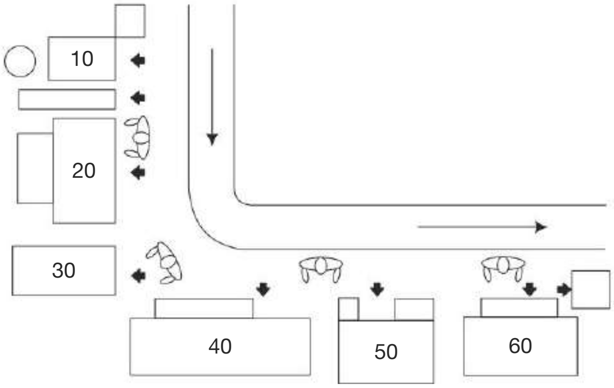


Fig. 18.12 *Arrangement of an L-shape cell*

For designing the cell layout, another piece of information that needs to be obtained is how the part moves through the workstations in the cell. That will be indicated in the operation sequence present in the routing sheet. The choice of the type of layout depends upon the various part moves in the cell. Theoretically, there can be four different types of moves that could be envisaged in a cell which are indicated in Fig. 18.13. They are the following:

- **Repeat operation** Consecutive operations are repeated in the same workstation. This means effectively there is no part move.
- **In-sequence move** In this, the part moves to the workstation that is next to it in the forward direction.
- **Bypassing move** In this, the part moves in the forward direction, but not to the immediate neighbour but to a station after one or two workstations.
- **Backtracking move** The part moves to the next workstation that is in the reverse direction.

Ideally, the parts should all move in only one direction and preferably in-sequence moves. However, the other moves are sometimes inevitable and therefore an account has to be made in the cell for these expected moves. For example, if the cell is provided with an automatic material-handling system such as a conveyor then the backtracking moves may have to be made manual.

Additional information required for finalising the cell layout includes

- **Workload** of the cell which is identified by the number of parts to be produced in the cell for a given time period, together with the actual time spent by the part at each of the workstation.
- **Part characteristics** to finalise the type of material handling to be incorporated in the cell. The characteristics include the part material, size, shape, mass, and other physical attributes.

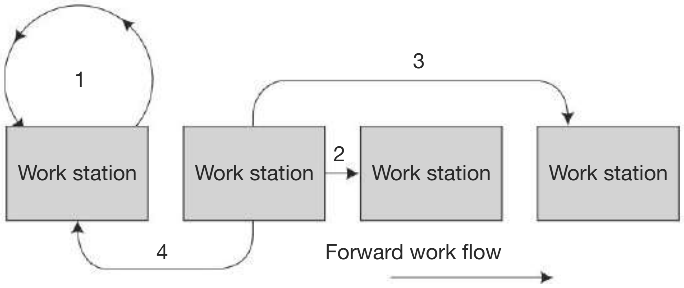


Fig. 18.13 *Types of part moves in a cell from one workstation to another workstation; 1 - Repeat operation, 2 – in-sequence move, 3 – bypassing move, and 4 – backtracking move*

A key machine in a GT cell is a machine that is more expensive to operate or performs certain critical operations in the plant [Groover, 2001]. In view of its importance, care has to be taken to see that this machine is utilised to the maximum extent possible, even when other machines in the cell are less utilised. When a GT cell with a key machine is formed, other machines in the cell are so selected that all the parts that need the key machine for their processing are completed in the cell. Obviously, cell utilisation will be different from the utilisation of the key machine which needs to be considered while designing the cell.

18.4.2 Machining Cell Planning

At the end of PFA, the groups of components and the workstations that have to be used to process them have been identified. The next step in laying out the cell is arranging these machines in an order such that part movement is minimised within the cell. A number of heuristics have been suggested by Hollier [1963] which help in this regard. The method described here considers the 'From – To' chart which provides the volume of part movement between the machines. The heuristic tries to maximise the in-sequence move of parts in the cell [Groover, 2001].

The method can be outlined as follows:

Develop the From–To Matrix The information present in the routing sheet is converted in the form of part movement volume between the various machines present in the cell and arranged as a matrix.

Compute the 'From' and 'To' sums for Each Machine Add all the trips taken by each machine 'From' rows as well as 'To' columns. This is essentially the summation of each row and column in the matrix.

Assign Machines to the Cell Based on Minimum 'From' or 'To' sums The machine having the smallest sum in either 'From' or 'To' sum is selected. In case the minimum value happens to be from 'To' sum, the machine is placed at the beginning of the sequence, while in the case of 'From' sum, it is placed at the end of the sequence. When there is a tie for the minimum value, the following rules are applied:

- Calculate the ratio of 'From/To' for all the machines. Choose the machine that has the lowest 'From/To' ratio.
- If the same machine has the smallest 'From' and 'To' sum, skip that machine and select the next lowest machine.
- If two different machines have the smallest sum, one for 'From' and the other for 'To', select both of them keeping one in the beginning and the other at the end of the sequence.

Rearrange the 'From – To' Matrix Remove the machine(s) selected from the previous step from the matrix by eliminating the rows and columns corresponding to the machine(s). Repeat steps 2 and 3 until all the machines are allocated.

An example will clarify the procedure.

Example 18.5 A GT cell processes a number of parts on four workstations. The part-move data is given in Table 18.18. From the given data, arrange the workstations in the proper sequence. Show the output in the form of a flow diagram.

Table 18.18 *From–To chart for Example 18.5*

To → From ↓	1	2	3	4
1	0	10	0	30
2	35	0	0	20
3	15	45	0	0
4	15	0	0	0

Solution Calculate the 'From' and 'To' sums from the given data as shown in Table 18.19. Since the minimum 'To' sum for the station 3 is '0', it is placed at the beginning of the sequence. Next, the station 3 is removed from the matrix as shown in Table 18.20 and 'From' and 'To' sums are recalculated. In Table 18.20, the station 2 has the lowest sum and is placed next to the station 3. Repeat it for other stations as shown in Table 18.21. The final chosen workstation sequence is 3 → 2 → 1 → 4. The actual flows are shown in Fig. 18.14.

Table 18.19 *From–To sums for Example 18.5*

To → From ↓	1	2	3	4	From sum
1	0	10	**0**	30	40
2	35	0	**0**	20	55
3	15	45	**0**	0	60
4	15	0	**0**	0	15
To sum	65	55	0	50	170

Table 18.20 *From–To sums for Example 18.5*

To → From ↓	1	2	4	From sum
1	0	**10**	30	40
2	35	**0**	20	55
4	15	**0**	0	15
To sum	50	**10**	50	110

Table 18.21 *From–To ratio for Example 18.5*

To → *From ↓*	*1*	*4*	***From sum***	***Ratio***
1	0	30	30	2
4	**15**	**0**	**15**	**0.5**
To sum	15	30		

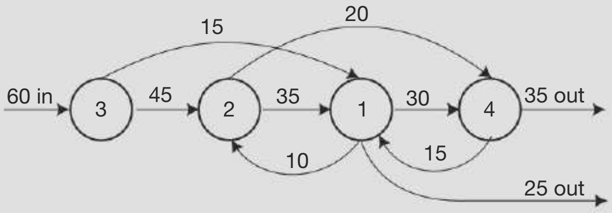


Fig. 18.14 *Flow of material between the workstations in Example 18.5*

18.4.3 Guidelines for Implementing GT

The following are some guidelines for implementing group technology [Rao, Tewari and Kundra, 1993].

1. Collect a complete variety of components being manufactured in the company.
2. Get an estimate of the quantity to be produced for each variety over a period of time. This period will depend on the policies of manufacture and stocking the requirements. Due to fluctuations in demand, it may not be very certain.
3. Obtain the process sheet for each component.
4. For implementation of GT, exclude special operations, e.g., heat treatment, painting, forging, etc.
5. Try to plan the production of all the parts of a family in one cell. This may mean deliberate under-utilisation of secondary machines. This would be acceptable if these machines are inexpensive, otherwise such machines could be utilised by other cells too. This would enable flexibility in labour.
6. The layout should be such as to permit a redistribution of load amongst various cells whenever fluctuations in load arise.
7. Study the data on operations, their sequences for various components, the quantity required, the machine-tool capacities, the set-up time, and the machining times. Based on this, calculate the workload on each machine tool.
8. Collect the components, which use (a) the same sequence of machine tools, (b) same machines, or (c) similar machines.
9. Form cells on the basis of the above information. Such a layout enables an easy transfer of load amongst cells when the load varies. Inexpensive machines may be duplicated.
10. There are machines that are required for a large number of components but their use in each of the cells is not much. It would be justified to keep them separate and made available to all the cells requiring them. This is particularly true for expensive machines. For example, forging machine and heat-treatment equipment can be formed into a single cell or two separate cells.

18.5 COMPUTER AIDED PROCESS PLANNING

Traditionally, process planning is performed manually by highly experienced planners who possess in-depth knowledge of the manufacturing processes involved and the capabilities of the shop-floor facilities. Because of the experience factor involved in planning for the physical reality of the product and in the absence of standardisation of the process, conventional process planning has largely been subjective. Moreover, this activity is highly labour-intensive and often becomes tedious when dealing with a large number of process plans and revisions to those plans. Rather than carry out an exhaustive analysis and arrive at optimal values, which would be too time-consuming, process planners often tend to play safe by using conservative values and this situation invariably leads to non-optimal utilisation of the manufacturing facilities and longer lead times. They also would not be in a position to see whether a similar component has already been planned in view of the difficulties involved in going through all the old process plans.

The need of shorter lead times, satisfying varied customer demands on the product variety and the optimum use of the manufacturing facilities, prompted research organisations and industries to automate many functions in the product cycle. Harnessing the power of the computer is extremely advantageous in process planning since a vast amount of data needs to be used for arriving at the right decision for planning the manufacturing operations.

Computer Aided Process Planning (CAPP) is a means to automatically develop the process plan from the geometric image of the component. The key to development of such CAPP Systems is to structure the data concerning part design, manufacturing facilities and capabilities into categories and logical relationships. CAPP thus appears to fully integrate CAD and CAM (Fig. 18.15).

18.5.1 Approaches to CAPP

There are two basic approaches to Computer Aided Process Planning: variant, and generative—which are briefly discussed below.

Variant Approach Variant approach, which is also called retrieval approach, uses a Group Technology (GT) code to select a generic process plan from the existing master process plans developed for each part family and edits to suit the requirement of the part (Fig. 18.16). The variant approach is commonly implemented with GT coding system. Here, the parts are segmented into groups based on similarity and each group has a master plan.

However, this approach is impractical in situations where small batches of widely varying parts are produced. Moreover, this method fails to capture real knowledge or expertise of process planners, and there is a danger of repeating mistakes from earlier plans that were stored in the database.

Generative Approach In the generative approach, a process plan is created from scratch for each component without human intervention. These systems are designed to automatically synthesise process information to develop the process plan for a part (Fig. 18.17). These systems contain the logic to use manufacturing databases and suitable part-description schemes to generate a process plan for a particular part. Most of the contemporary CAPP systems being developed are generative in nature. The generative approach eliminates disadvantages of the variant approach and bridges the gap between CAD and CAM.

18.6 CAPP: IMPLEMENTATION TECHNIQUES

Logical decision is a traditional implementation technique used in Computer Aided Process Planning. The simplest approach is to code the process capability in a computer program. A tree-structured classification can be used in the system and each process can be coded as a branch of a decision tree.

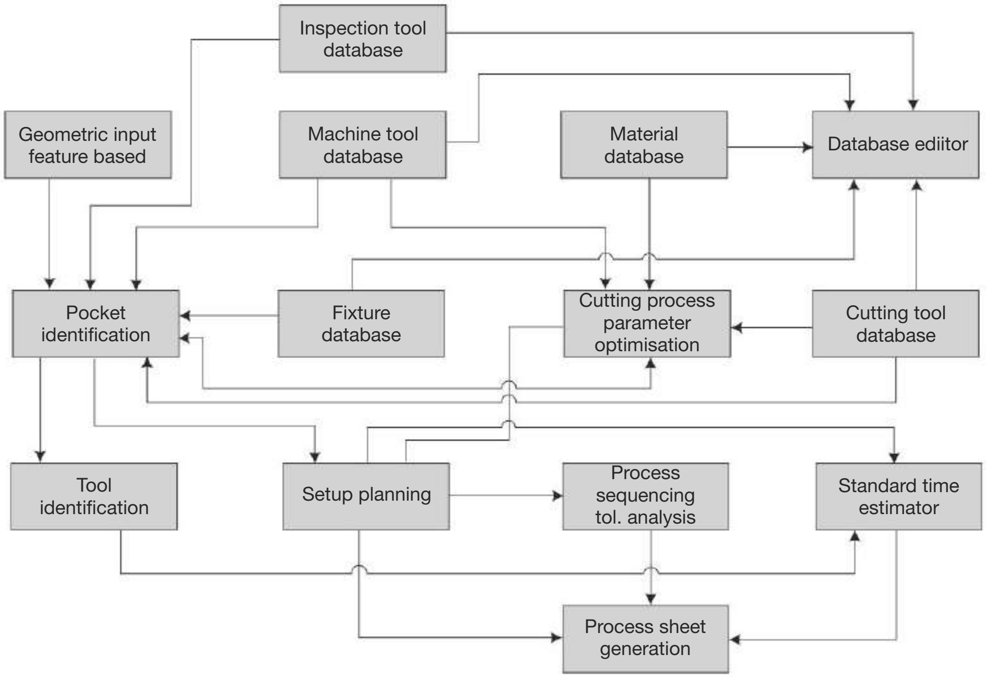


Fig. 18.15 *Architecture of a CAPP system*

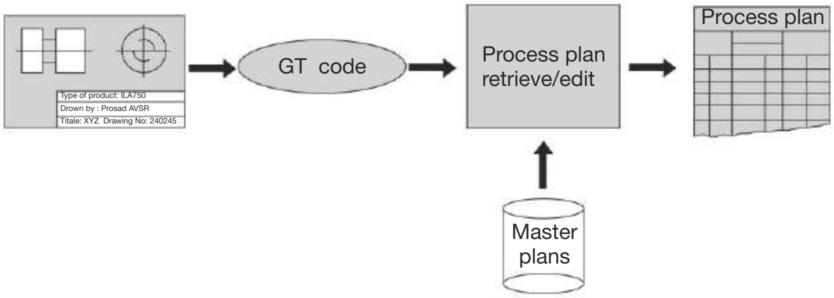


Fig. 18.16 *Variant approach to CAPP*

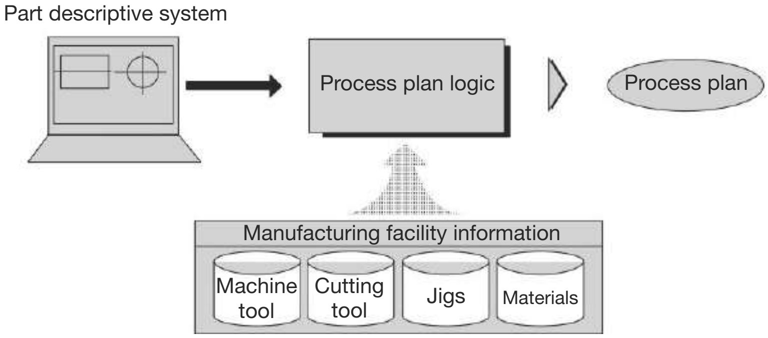


Fig. 18.17 *Generative approach to CAPP*

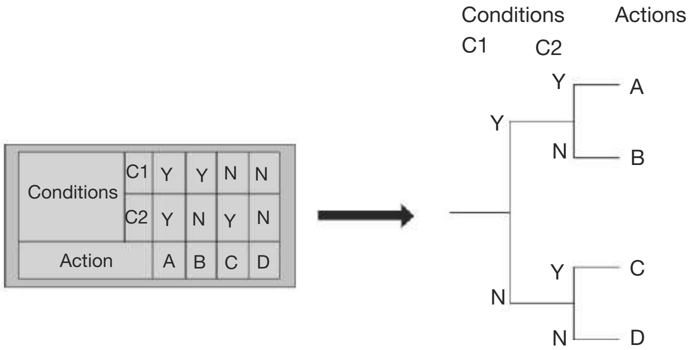
The decision logic should be present in a format which is easy to visualise and check for completeness, contradictions and redundancy. The information should be easily coded and debugged and full program documentation should be provided to help in further modification. Generally, each manufacturing process is defined as a separate entity based on the capabilities to generate or modify geometric features or properties. Only the values of decision variables change over time or between companies. The techniques for structuring the decision logic are numerous and varied.

The objective of decision logic in a CAPP system is to match the process capabilities with design specifications in an optimal way. Generally, the most common decision logic can be classified as one of the three methods: decision tables, decision trees and AI.

Decision Tables A decision table is partitioned into conditions and actions and is represented in a tabular form. A decision table is a program structuring tool, which provides readable documentation as an automatic by-product. Also, a decision table can be used with a preprocessor to eliminate some program coding, and to provide automatic checks for completeness, contradiction and redundancy.

Decision Trees A decision tree is a graph with a single root and branches emanating from the root. Decision trees are easier to customise, update, maintain, visualise and develop. Decision trees can be represented as computer codes or data. The tree as a computer code is converted to a flowchart. The starting node is the root, and every branch represents a decision statement, which is either false or true.

Decision trees can be more easily updated and maintained compared to decision tables. The flexibility in expanding and contracting the tree if necessary is another advantage compared to decision tables. Figure 18.18 illustrates the decision table and tree.



Conditions	C1	Y	Y	N	N
	C2	Y	N	Y	N
Action		A	B	C	D

Decision table

Decision tree

Fig. 18.18 *Decision table and tree as used in CAPP*

Expert System Techniques (AI) The solution to the process-planning task depends mainly on the empirical knowledge relevant to the organisation based upon the existing facilities. The popularity of expert systems in CAPP is due to this qualitative, subjective, imprecise and company specific nature of the process-planning knowledge. The expert systems are designed to cope with such knowledge characteristics. They are also much easier to modify and customise than the fixed logic conventional systems, because the knowledge in expert systems is explicitly represented and segregated from the planning (inference) mechanism.

In general, problems in a production system formalism can be represented by the following: 1. an initial state, 2. a goal state, 3. a set of operators, and 4. a control structure. One of the frequently used methods to solve problems in AI is the 'theorem proving technique'. Using this technique, we can proceed from the initial state to the goal state. Proving that the goal state can be reached from the initial state will involve the applications of the operators, thereby providing a solution to the problem.

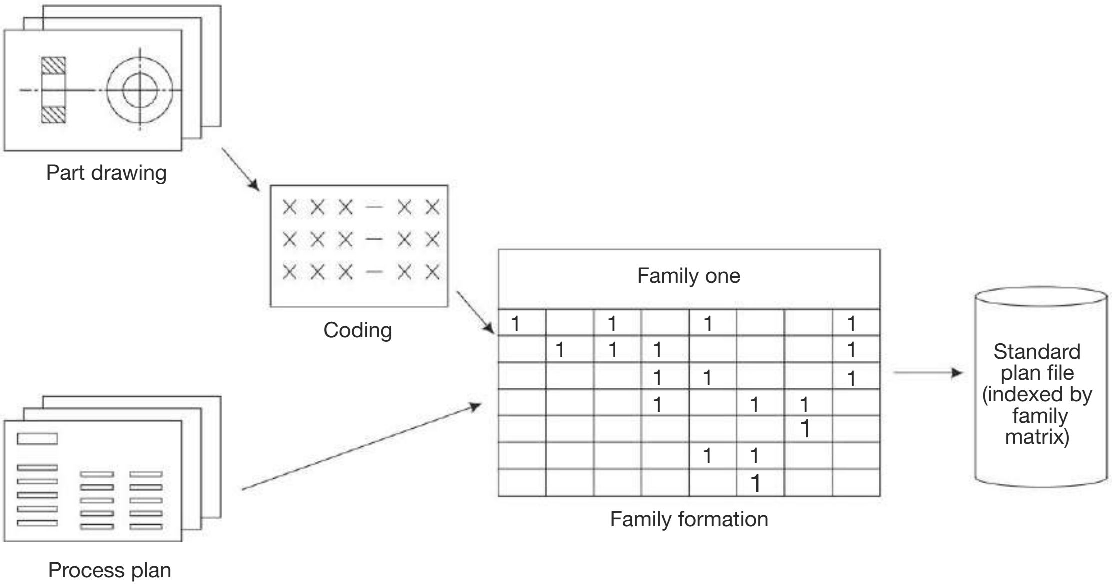
The process-planning problem falls into this category. In the part manufacturing problem, the initial state is the raw material or workpiece from which the part is to be produced and the goal state is the finished part. The set of operators comprise the available machine tools, cutters, etc. Processing of the part involves proceeding from the raw material to the finished part. Application of the operators (machine tools and cutters) moves the problem from one state to another. The various stages represent the workpiece in 'work in progress' condition.

$$S_{\text{initial}} \rightarrow S_1 \rightarrow S_2 \rightarrow \ldots . \rightarrow S_{\text{final}}$$

These stages are non-reversible as the material once removed cannot be added back. However, in process planning, as the actual material removal has not been done, the stages can be reversible, leading to the development of alternative process plans. This strategy is being made use of in the development of expert CAPP systems.

18.6.1 Essential Elements in a Retrieval-Type CAPP System

Steps in developing a retrieval-type CAPP system are presented in Fig. 18.19. The first step in the development is the identification of the part families for which the process plan library is to be developed. With the help of group technology coding scheme, all the components that are likely to be manufactured in the shop are coded. This forms the part-family matrix. For each of the parts in the family, the process plan is manually coded and verified through practice if possible. All such manually developed process plans are stored into the process-plan database with the part number in the part-family matrix as the key.



1		1		1			1
	1	1	1				1
			1	1			1
			1		1	1	
						1	
				1	1		
					1		

Fig. 18.19 *Essential elements of a retrieval-type CAPP system*

Whenever a new part is required, first the GT code for the part is developed. Then from the process-plan database, the process plan is retrieved matching the group code. The process plan is a generic one with possibly many features not required in the current part. Then this process plan is edited to match the actual part requirements. However, if the part falls into a group for which no process plan exists, a new plan is manually coded and entered into the database.

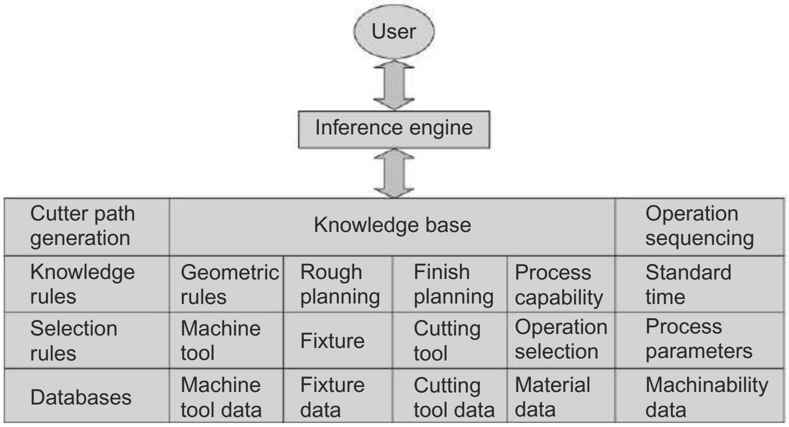
18.6.2 Essential Elements in a Generative CAPP System

Most of the generative CAPP systems need to have some amount of artificial intelligence built into them. A typical generative CAPP system with an expert system (inference engine) has been shown in Fig. 18.20. The various elements and parts as shown are required to provide the necessary functionality.

A number of steps are involved in the development of generative CAPP systems. Suggested methodological steps are detailed below:

(i) Identify the machinable volumes called pockets by taking the difference of blank size and the finished component size. For each of the pockets, attach the necessary technological details relevant for manufacturing. The blank size if not given directly can be identified as the largest volume that completely encloses the finished component.

(ii) Do a preliminary sorting of the pockets in order of levels that clearly indicate the likely sequence in the final process plan.



Cutter path generation	Knowledge base				Operation sequencing
Knowledge rules	Geometric rules	Rough planning	Finish planning	Process capability	Standard time
Selection rules	Machine tool	Fixture	Cutting tool	Operation selection	Process parameters
Databases	Machine tool data	Fixture data	Cutting tool data	Material data	Machinability data

Fig. 18.20 *Essential elements of a generative type CAPP system*

(iii) Examine the pockets for any possibility of combining so that the machining operations could be reduced.

(iv) Select the machine tool that can be used for each of the identified pockets. Minimise the total number of machine tools required. This may have to be modified recursively based on the operation sequence selected.

(v) Identify the process sequence required for the machining of each of the pockets based on the technological requirements. Help may be obtained in the shape of canned sequences based on the technological parameters. Any possible alternate plans can also be identified at this stage.

(vi) For each of the pockets and the operation decided, select the cutting tool required.

(vii) Obtain the optimum cutting-process parameters (speed, feed, number of passes and depth of cut) for each of the pocket, tool and the operation combination identified in the earlier steps.

(viii) Sort the operations on the basis of machine tool and cutting tool. Sequence the operations on the basis of machine tool and cutting tool by making use of the heuristic rules for the purpose.

(ix) Evaluate the machining time and idle time involved in the production of the component. Select the final process plan based on the lowest cost or machining time.

(x) Present the final results in any suitable form such as

- Process sheet (alpha numeric)
- Process pictures
- Machining simulation steps
- CLDATA or CNC part program

One of the important steps to be considered while developing the CAPP program package is that the ultimate decision be left to the process planner (user). This means that the results that are coming at the various points in the planning procedure be displayed for the possible verification and approval by the planner. This helps in the better appreciation of the package as well as the possibility of any modifications, should the expert planner detect some apparent problem in the solution offered by the system. One problem is that the time taken for complete planning of the part increases because of the interaction, but is desirable.

Summary

- Most of the modern-day manufacturing is characterised by medium-volume operations that cannot employ the low-cost benefits of large volume production. Group Technology (GT) has been developed to fill this need. Also, Computer Aided Process Planning (CAPP) is becoming increasingly important to form part of the design cycle in view of its ability to quickly generate optimum plans directly from the product model data.
- GT as a philosophy started in the early 1950's and brings the advantages of mass manufacture to small batch manufacture by combining batches based on certain similarity principles.
- GT employs product layouts to reduce the travel time and waiting time of parts unlike the process layouts.
- GT utilises coding and classification system of parts to be able to group them into similar groups by means of certain characteristics such as similarity in shape, size or processing method used. Different types of coding systems such as monocode, polycode and hybrid coding systems are possible. There are a number of coding schemes in use such as Opitz, MICLASS, KK-3, etc.
- To organise the processing machines into product layouts, production flow analysis, which identifies the similar operations into a single group of machines to form a cell, is used. The grouping process can be done efficiently using either rank cluster algorithm or direct clustering algorithm.
- There are many arrangements of machines in the GT cell, such as in-line, U or L-layout, of which U-cell is the more common and widely used.
- Key machine in a GT cell is an important concept to identify a single machine that is used for critical operations.
- The identification of the machine layout in a GT cell can be easily accomplished by utilising simple heuristic rules from the available data such as 'From–To' matrix.
- It is necessary to follow the guidelines to develop efficient manufacturing cells utilising GT.

- CAPP is a method to develop efficient process plans directly from the CAD model of the part by considering all the manufacturing resources unlike the manual process-planning methods.
- CAPP utilises two major approaches, viz., variant approach using GT for classifying process plans, and generative approach to develop process plans directly from scratch every time.
- Implementing a variant CAPP system is straightforward since it makes use of the existing process plans that are already proven in the form of a catalogue from which the required process plan needs to be selected and edited.
- Implementing a generative CAPP system requires far more effort. This involves the codification of the various knowledge related to the manufacturing resources of the enterprise and build the algorithms for identifying the resources and plans based on the component geometry.
- Artificial intelligence techniques have often been found to be useful in generative CAPP system development.

Questions

1. What do you understand by the term 'group technology'?
2. Why is group technology more important in the present manufacturing scenario?
3. What is the basis for forming groups in group technology?
4. Explain the benefits of group-technology layout compared to a process-type layout.
5. What are the various methods available for forming groups in group technology? Explain briefly.
6. Explain the types of coding systems possible for group technology.
7. Explain the concept of composite part with an example.
8. Briefly explain about the Opitz coding system generally used in group technology.
9. Explain the MICLASS coding system as used in group technology.
10. Explain the DCLASS coding system as used in group technology.
11. Explain the CODE coding system as used in group technology.
12. Explain the KK-3 coding system as used in group technology.
13. Give a brief description of production flow analysis.
14. What is the method used for forming cells in group technology. Briefly explain with an example.
15. Mention the advantages to be gained by the adoption of group-technology methods.
16. What are the limitations of group technology?
17. What do you understand by the term 'key machine' in group technology? Explain its relevance.
18. Explain different types of cell designs that are used in group-technology cells.
19. What is cellular manufacturing? Explain its relevance in modern manufacturing.
20. Briefly explain the guidelines for implementing group technology.
21. Briefly explain the need for computer-aided process planning.
22. What are the various approaches available for computer-aided process planning.
23. Give a brief description about the retrieval type of computer-aided process planning method.
24. What are the difference between retrieval and generative type of computer-aided process planning? Which is better? Explain your answer.
25. What methods are available for taking decisions in the process of developing a process plan?
26. Briefly explain the methodology to be followed for developing a retrieval type of computer aided process planning system.
27. Briefly explain the methodology to be followed for developing a generative type of computer-aided process planning system.

Problems

1. Develop the form code (first 5 digits) of the Opitz code for rotational components for the component shown in Fig. 18.21.
2. Develop the form code (first 5 digits) of the Opitz code for rotational components for the component shown in Fig. 18.22.
3. Develop the form code (first 5 digits) of the Opitz code for rotational components for the component shown in Fig. 18.23.
4. Develop the form code (first 5 digits) of the Opitz code for rotational components for the component shown in Fig. 18.24.

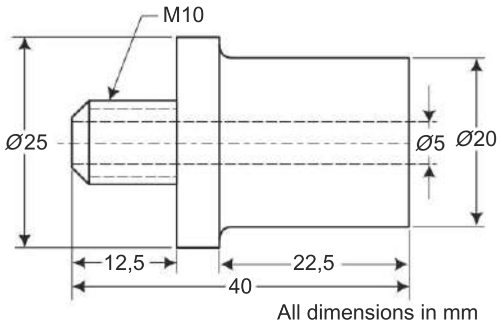


Fig. 18.21 *Example component*

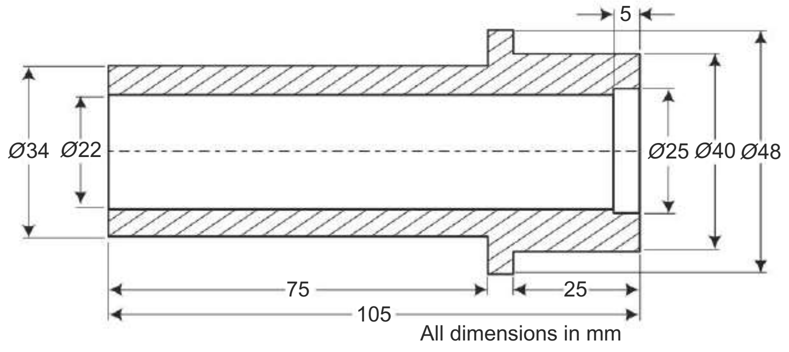


Fig. 18.22 *Example component*

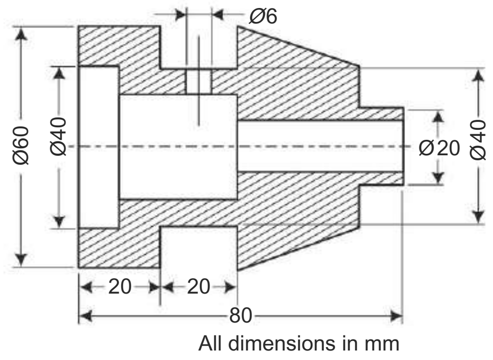


Fig. 18.23 *Example component*

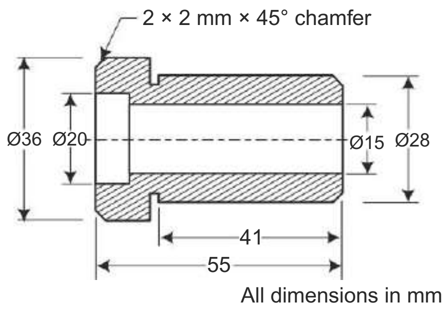


Fig. 18.24 *Example component*

5. Obtain the part families for the incidence matrix given in Table 18.22. Use rank-order clustering method.

Table 18.22 *Initial incidence matrix for Problem 18.5*

Parts → *Machines ↓*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
A	1			1	1			1	1	
B				1		1				
C		1	1		1			1	1	
D						1	1			1
E	1		1							
F		1					1			1

6. Obtain the part families for the incidence matrix given in Table 18.23. Use direct clustering method.

Table 18.23 *Initial incidence matrix for Problem 18.6*

Parts → *Machines ↓*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
A	1			1	1			1	1	
B						1				1
C	1	1		1				1		
D			1			1	1			
E		1		1	1				1	
F			1				1			1

7. Obtain the part families for the incidence matrix given in Table 18.24. Use direct clustering method.

Table 18.24 *Initial incidence matrix for Problem 18.5*

Parts → *Machines ↓*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
A	1			1			1	1		1
B		1	1		1	1			1	
C	1	1		1						1
D			1		1			1	1	
E	1	1					1		1	
F		1	1			1				1

8. The routings of 5 parts are given in Table 18.25 which are to be manufactured in a GT cell. Develop the part-machine incidence matrix. Apply rank-order clustering or direct clustering method to obtain the logical machine groups and the corresponding part families.

Part	*Weekly quantity*	*Machine routing*
1	50	C → B → G
2	20	F → A
3	75	F → E
4	10	F → E→ A
5	12	C → B
6	35	F → E → A
7	45	C → B → A
8	40	C → B → G → D

9. A GT cell processes a number of parts on four workstations. The part-move data is give Table 18.25. From the given data arrange the workstations in the proper sequence. Show the output in the form of a flow diagram.

Table 18.25 *From-to chart for Problem 18.9*

To → *From ↓*	***1***	***2***	***3***	***4***
1	0	15	0	35
2	40	0	0	25
3	20	50	0	0
4	20	0	0	0

10. A GT cell processes a number of parts on four workstations. The part-move data is give Table 18.26. From the given data, arrange the workstations in the proper sequence. Show the output in the form of a flow diagram.

Table 18.26 *From–to chart for Problem 18.10*

To → *From ↓*	***1***	***2***	***3***	***4***
1	0	20	0	50
2	0	0	0	0
3	60	0	0	30
4	0	60	0	0

11. A GT cell processes a number of parts on four workstations. The part-move data is give Table 18.27. From the given data arrange the workstations in the proper sequence. Show the output in the form of a flow diagram.

Table 18.27 *From-to chart for Problem 18.11*

To → *From ↓*	***1***	***2***	***3***	***4***	***5***
1	0	20	90	0	0
2	0	0	0	95	0
3	0	0	0	0	0
4	80	0	30	0	0
5	0	85	0	30	0

19

PRODUCTION PLANNING AND CONTROL

Objectives

The need for a variety of information for successful operation of manufacturing enterprise was discussed earlier. In this chapter, a part of that information that will be forming the actual production operation is discussed. After completing the study of this chapter, the reader should be able to

- Learn about production planning as it converts the business plan into more detailed production rates for individual items
- Appreciate capacity planning as a link between master production schedule and the material requirement planning
- Understand the needs of Master Production Schedule (MPS) and methods to develop it
- Appreciate the place of Material Requirement Planning (MRP) in the manufacturing operations and develop methods to link Bill Of Materials (BOM) and MRP
- Plan and execute the Production Activity Control (PAC), which actually deals with operations in the shop floor
- Understand the information integration methods that are commonly used in industries such as MRP II, OPT and JIT

19.1 INTRODUCTION

As discussed in Chapter 17, the production plan is derived from the business plan which becomes the starting for the production planning and control functions of an enterprise as shown in Fig. 19.1. From this, the rest of the functioning modules need to be derived. In this chapter, a brief presentation of these functioning modules are described.

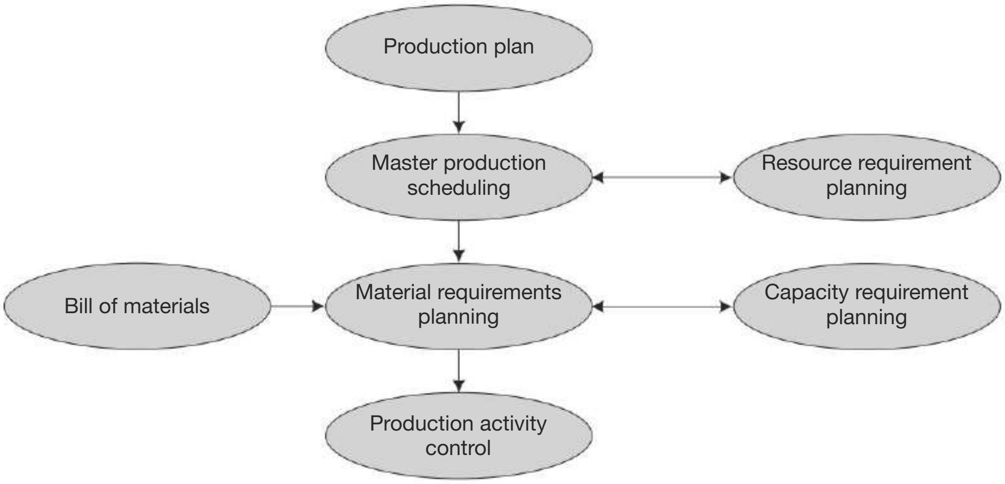


Fig. 19.1 *The various components of the production planning and control system*

19.2 PRODUCTION PLANNING

Production plan for a manufacturing company will have to focus on production rates and inventory holdings based on the varying customer requirements and capacity limitations. The business plan specified by the top management of the organisation for a year or two is used to develop the aggregate plan, which in turn is used to develop the master production schedule as described later. The production plan specifies the product-family production rates, inventory levels and workforce levels.

In order to come up with an appropriate aggregate production plan for the company based on the business plan, a number of inputs need to be considered. The overall idea of the number of inputs from various departments of the company that are to be considered for aggregate planning is shown in Fig. 19.2. The objectives that should be considered by the aggregate planning team are the following:

- **Maximise profits and minimise costs** This is the overall goal. Minimising the costs without affecting the production quantities always maximises profit, assuming the customer demand remains the same.
- **Maximise customer service** In the current market-driven economy, satisfying the customer is the most important criterion for success. This means improving delivery time and on-time delivery performance. However, this means additional workforce, machine capacity and inventory levels.
- **Minimise inventory** Inventory being cost, reducing this always leaves the capital which could be better spent on productive investments.
- **Maximise utilisation of plant equipment**
- **Minimise changes in production environment** Changes in production rates or workforce levels require a lot of adjustments which may not always improve productivity.

The various alternatives that the team should consider while balancing the above objectives are adjusting the workforce, adjusting the inventory levels between lean and heavy demand periods, overtime or undertime to employees, adjusting the vacation schedules, hiring subcontractors, provision for backlogs and backorders, promotional programs to promote demand, and having a complementary production program to even out the demand.

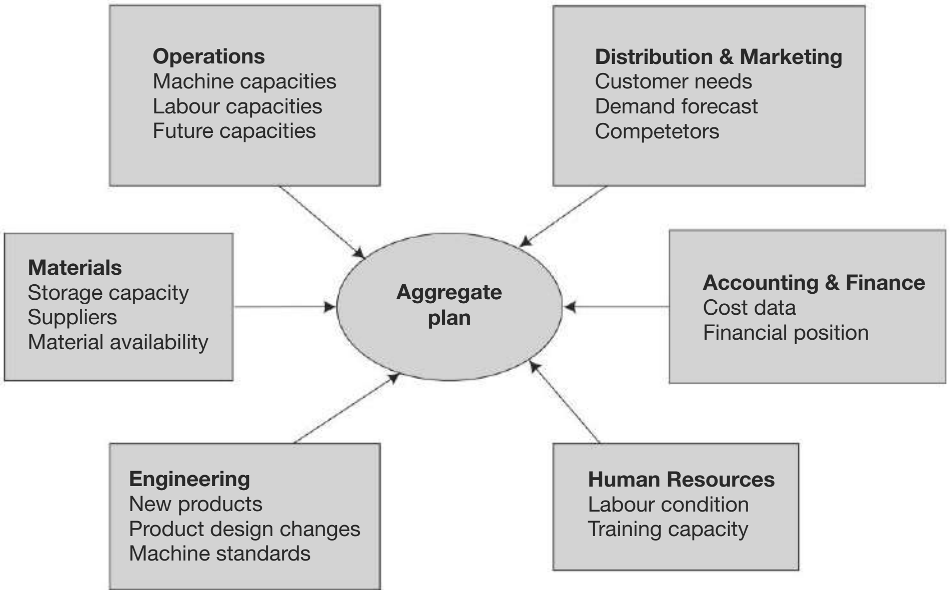


Fig. 19.2 *Various inputs from different functional components for aggregate planning*

Based on the demand forecast and other information on hand (replacement requirements, backlogs, distributor requirements, etc.), planners can prepare a production plan for the planning horizon. Based on the planned production as above, it is necessary to identify the resources required to achieve this production. Some of the inputs required to establish this are regular time costs, overtime costs, hiring and layoff costs, inventory holding cost and backorder and stock-out costs.

The planning process is always an iterative one. Start with a tentative plan, and evaluate this plan against the available resources and satisfying the stated objectives. If not acceptable, make changes and then repeat the evaluation. The final acceptable plan may only be achieved after several revisions and adjustments.

Using an electronic spreadsheet is an easier way to complete the planning process. For this purpose, the required information is demand forecast for the planning periods and various costs and limits on workforce availability. Assuming all costs are linearly proportional to the volume produced and using the following notation, the spreadsheet formulae are shown in Table 19.1 for 4 planning periods.

h = holding cost per unit per period
r = cost per unit to produce on regular time
c = cost per unit to produce on overtime
s = cost per unit to produce on subcontract
u = undertime cost per unit
b = back order cost per unit per period
I_0 = beginning inventory level
I_4 = desired inventory level at the end of period 4
R_t = regular time capacity in period t
O_t = over time capacity in period t

S_t = subcontracting capacity in period t
D_t = forecast demand for period t
U = total unused capacities

Table 19.1 *Production planning using transportation model*

Alternatives		*Time period*				*Unused capacity*	*Total capacity*
		1	*2*	*3*	*4*		
Period	*Beginning inventory*	*0*	*h*	*2h*	*3h*	*4h*	I_0
1	Regular time	r	$r + h$	$r + 2h$	$r + 3h$	u	R_1
	Over time	c	$c + h$	$c + 2h$	$c + 3h$	0	O_1
	Subcontract	s	$s + h$	$s + 2h$	$s + 3h$	0	S_1
2	Regular time	$r + b$	r	$r + h$	$r + 2h$	u	R_2
	Overtime	$c + b$	c	$c + h$	$c + 2h$	0	O_2
	Subcontract	$s + b$	s	$s + h$	$s + 2h$	0	S_2
3	Regular time	$r + 2b$	$r + b$	r	$r + h$	u	R_3
	Overtime	$c + 2b$	$c + b$	c	$c + h$	0	O_3
	Subcontract	$s + 2b$	$s + b$	s	$s + h$	0	S_3
4	Regular time	$r + 3b$	$r + 2b$	$r + b$	r	u	R_4
	Over time	$c + 3b$	$c + 2b$	$c + b$	c	0	O_4
	Subcontract	$s + 3b$	$s + 2b$	$s + b$	s	0	S_4
Requirements		D_1	D_2	D_3	$D_4 + I_4$	U	

Start with a value for R_t and constraints on O_t and S_t and get a plan. Evaluate the plan. Then make suitable changes and repeat the evaluation until the objectives are satisfied.

19.3 CAPACITY PLANNING

The goal of capacity planning is to transform the manufacturing requirements, as set forth in the MRP stage, into a detailed machine loading plan for each machine or group of machines in the plant. The decisions in this stage are confined to the demands of the MRP stage, and the optimisation criteria are capacity balancing, meeting due dates, minimum level of work-in-process and manufacturing lead time based on the available plant capacity, tooling, on-hand material and workforce. The available plant capacity is also dependent upon its utilisation for each manufacturing unit which can be defined as

$$\text{Utilisation} = \frac{\text{Average output rate}}{\text{Maximum capacity}} \times 100$$

A bottleneck is an operation that has the lowest effective capacity of any operation in the process and thus will affect the final output of the total process.

Capacity requirements can be estimated as the ratio of the total processing hours required for one year's demand to that of the actual hours available from one machine per year after deducting the desired cushion. Capacity planning is a complex process involving many interlinked disciplines within the production organisation such as competitive priorities, quality management, capital intensity, resource flexibility, inventory and scheduling. Identify capacity gaps (difference between the projected demand and available capacity) and find ways in which these gaps could be filled. There may be many alternatives to fill the gap, at which time it is necessary to examine the alternatives that fit the needs of the organisation. The mathematical techniques used frequently for capacity planning process are the waiting line models and simulation.

19.4 MASTER PRODUCTION SCHEDULE

As per the APICS, a master production schedule is *the anticipated build schedule for those items assigned to the master schedule. It represents what the company plans to produce expressed in specific configurations, quantities and dates.*

The MPS provides a link between planning and actual manufacturing stages by reconciling with the resource constraints. It helps in keeping the priorities valid and also allows to make promises which can be kept. It forms the basis and drives MRP, which is seen later. The interaction of the various components of information with the master production schedule is shown in Fig. 19.3. The MPS relies on the following information:

- The production plan conveyed by the top management
- Long-term forecasts of the individual items
- Actual orders received from the customers for the plan period
- Present inventory levels of the individual items
- Resource constraints

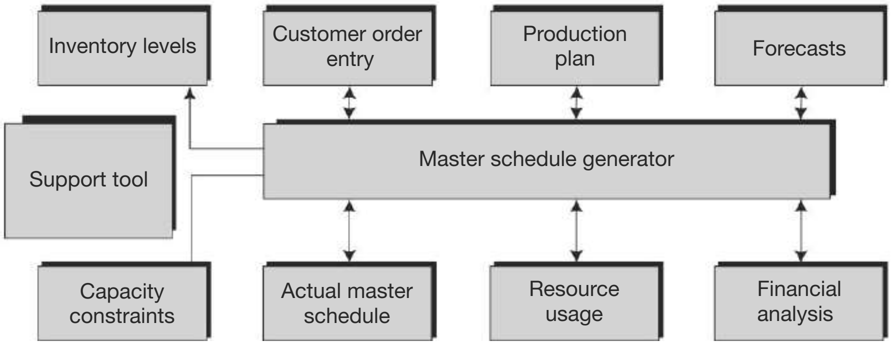


Fig. 19.3 *The various stages involved in the manufacturing of a product in concurrent engineering environment*

The objectives of the master production schedule are the following:

- Make the best use of the resources available in terms of equipment, material and labour.
- Maintain the inventory as low as possible to commensurate with the objectives of the company.
- Maintain the finished goods delivery as per the expectations of the customers.

In order to achieve these objectives, the framework to be followed is as follows:

- Use qualitative data for strategic levels, e.g., full customer order completeness.
- Direct linkage with customer order entry.
- Take over all product structure to evaluate the resource requirements.

- Use lower-level constraints and historical data in the generation of MPS, such as machine breakdowns.
- Make use of the support tools, e.g., identification of bottlenecks, identification of critical items, etc.

Based on the above framework, develop a preliminary MPS. Then check this MPS against the capacity constraints. If there is any difference in the capacities required and capacities available, explore the possibility of resolving through various means available at the disposal of the planning department. The final MPS should be arrived at based on the following goals:

- Best use of all the resources in all the plan periods.
- The cost incurred is not in excess including the overtime payment, subcontracting, additional manpower, etc.
- All the due dates for the orders can be met with.

As an example, consider the inputs for an MPS as shown in Table 19.2. The forecast for the first 5 plan periods is 5 units while for the last 3 periods, is 21.

Table 19.2 *Initial data for preparing the MPS*

	Period number							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Forecast	5	5	5	5	5	21	21	21
Available								
MPS								
Inventory on hand	20							

To develop the preliminary MPS divide the total forecast into equal amounts in the plan periods.

$$\text{Total demand} = 88 \text{ units}$$

$$\text{Production per plan period} = \frac{\text{Total forecast}}{\text{Number of plan periods}} = \frac{88}{8} = 11$$

Fill the MPS column with 11 as shown in Table 19.3. Then calculate the items available as follows:

$$\text{Available} = \text{Inventory on hand} + \text{MPS} - \text{Forecast}$$

Inventory on hand for the second plan period will be the available of the first plan period. Accordingly, fill the table as shown in Table 19.3. The same step is continued for all the plan periods as shown in Table 19.4.

Table 19.3 *Calculating the MPS for the first plan period*

	Period number							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Forecast	5	5	5	5	5	21	21	21
Available	26							
MPS	11	11	11	11	11	11	11	11
Inventory on hand	20	26						

Table 19.4 *Calculating the MPS for all the plan periods*

	Period number							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Forecast	5	5	5	5	5	21	21	21
Available	26	32	38	44	50	40	30	20
MPS	11	11	11	11	11	11	11	11
Inventory on hand	20	26	32	38	44	50	40	30

However, in cases where the economic lot size exists for manufacture then the MPS will have to be calculated differently. For example, if 30 is the economic lot size then the total production is divided into 3 economic lots and then distributed as shown in Table 19.5. The rest of the table is filled as before.

Table 19.5 *MPS calculation with economic lot size*

	Period number							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Forecast	5	5	5	5	5	21	21	21
Available	45	40	35	60	55	34	43	22
MPS	30			30			30	
Inventory on hand	20	45	40	35	60	55	34	43

19.5 MATERIAL REQUIREMENT PLANNING (MRP)

The objective of the MRP is to get the right materials to the right place at the right time minimising the inventory cost. In order to do this, MRP utilises the master production schedule for the purpose of material explosion. The major objectives of MRP in an organisation are the following:

- *Reduction in inventory*, by specifying exactly when material is to be ordered to meet the master production schedule.
- *Coordinate production and purchase* and thereby reduce the manufacturing lead time.
- *Realistic commitment* for the delivery of the finished products.

In order to apply MRP, it is necessary to have the full inputs from the following as shown in Fig. 19.4.

- Master production schedule
- Bill Of Materials (BOM)
- Present inventory status

Change is continuous within the manufacturing environment.

- The master production schedule changes.
- The inventory status changes.
- Engineering activity modifies bill of materials.
- Orders are released to the shop floor or purchasing.

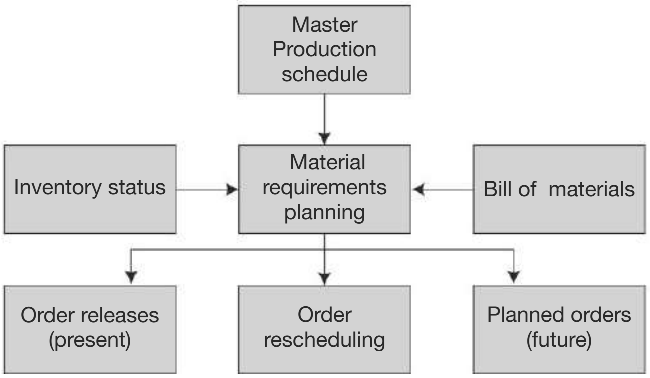


Fig. 19.4 *The various stages involved in the manufacturing of a product in concurrent engineering environment*

Some of these events are planned and some are unplanned. The MRP system therefore should have the capability to adapt to the changes.

A few definitions of the terms are given that need to be understood before we proceed with MRP calculations.

Allocated Quantity The actual amount of items in the inventory, which are committed for a particular use and hence are not available for allocation.

Gross Requirements In order to meet the planned output levels, the actual quantity required at the end of the planned period.

Scheduled Receipts The amount of items to be received from the suppliers at the beginning of the planned orders as a result of the orders placed earlier.

Available Quantity The amount of items that is available at the end of the planned period to be allocated in the succeeding periods.

Net Requirements The additional quantity to be obtained to meet the planned production for the period—calculated by subtracting the scheduled receipts and the available quantity from the gross requirements.

Planned Order Receipt The amount of items that are planned to be ordered such that they are received at the beginning of a period to meet its net requirements; the order has not been placed yet.

Planned Order Release The amount of items that is planned to be released such that they are received when needed; the order precedes the receipt in time by an amount equal to the lead time.

Bill of Materials (BOM)Bill of materials is the list of materials that is used in the final assembly of the product. The structure of the BOM is such that it is possible to know how the part is made, starting from the component level to the sub-assemblies to the final assembly. A typical BOM structure is shown in Fig. 19.5. All the parts that are required to make the part *A* are shown in Fig. 19.4 in structured form. It shows all the parts that are required for making parts *A* and *B*, *C* and *D*. *C* and *D* are the subassemblies, which in turn are made from the components *E*, *F*, *G* and *H*. An actual bill of materials for making the gear shift lever is given in Table 19.6. The bill of materials is generally shown in multiple levels to indicate the type of product structure involved. That shows clearly the method followed in the assembly of the product.

Inventory Status File The MRP calculations require that up-to-date file of the inventory status of all the components that go into the assembly of the part should be available. The file contains the up-to-date

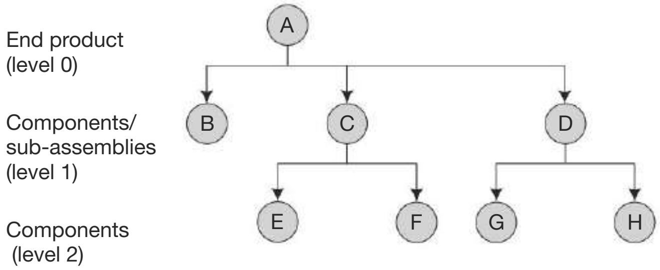


Fig. 19.5 *The Bill of Material (BOM)*

Table 19.6 *Bill of materials for the gear shift lever*

Level #2	*Level #1*	*Level #0*
	Bushing rubber Distance pierce Bushing #1 Bushing #2 Snap ring Inner pipe Bolt M8 × 100 Pipe spacer Seat Return spring Nut M8 Washer Washer plain M8 Washer spring Grease Bracket →	Gear shift lever assembly
Plate cable end Bolt weld Plate assembly → Plate select Pin	Bracket assembly	
Pin select Pipe *A* → Bracket *A*	Lever A assembly	
Rod shift Pipe lever shift → Pin	Lever shift assembly	
Pipe *B*1 Pipe *B*2 → Pin bracket	Lever *B* assembly	

information about the part name, item code, level in BOM, on- hand stock, safety stock, allocated, lot size for manufacture and the manufacturing or procurement lead time. A typical file is shown in Table 19.7.

Table 19.7 *Inventory status file for the gear shift lever*

No	*Part name*	*Item code*	*Level*	*On hand*	*Safety stock*	*Allocated*	*Lot size*	*Lead time days*
1	Lever assembly		0	276	100	0	1000	1
2	Bracket assembly		1	1160	500	0	1000	4
3	Lever *A* assembly		1	1160	500	0	1000	4
4	Lever shift assembly		1	1160	500	0	1000	4
5	Lever *B* assembly		1	1160	500	0	1000	4
6	Bushing rubber		1	26050	500	0	1000	4
7	Distance pierce		1	5000	1000	2500	5000	14
8	Bushing #1		1	29300	3000	5000	10000	4
9	Bushing #2		1	74000	5000	15000	10000	4
10	Snap ring		1	17400	5000	8000	3500	4
11	Pipe spacer		1	2300	500	1000	2500	7
12	Inner pipe		1	1700	500	0	1000	7
13	Bolt M8		1	8700	2000	6000	3000	4
14	Nut M8		1	11900	2500	5000	10000	4
15	Washer M8		1	36500	5000	10000	6000	7
16	Washer plain M8		1	9500	1000	2500	4000	4
17	Washer spring		1	23500	3000	5000	10000	4
18	Seat		1	24000	5000	6000	8000	4
19	Return spring		1	18750	5000	5000	10000	4
20	Bracket *m/t* cable		1	7800	1000	0	2500	7
21	Grease		1					4
22	Plate cable end		2	2500	2000	0	3000	1
23	Bolt weld *M*8		2	10000	3000	6000	5000	7
24	Plate assembly		2	4750	2000	0	2000	1
25	Plate select		2	5330	2000	0	4125	1
26	Pin		2	1130	500	0	1000	14
27	Pin select		2	2500	500	0	1000	14
28	Pipe *A*		2	250	100	0	1000	4
29	Bracket *A*		2	9000	2000	0	3500	1
30	Rod shift		2	2000	500	0	900	14
31	Pipe lever shift		2	230	100	0	950	7
32	Pin		2	1200	250	0	800	14
33	Pipe *B*1		2	1300	500	0	1500	7
34	Pipe *B*2		2	1615	500	0	1500	7
35	Pin bracket		2	1500	500	0	1500	7

MRP Processing Logic The MRP processing requires that each of the part in the MPS be exploded using the BOM structure, giving the total component requirement to satisfy the MPS for each time period. From the gross requirements, it calculates the planned order receipts by subtracting the inventory already available. From this, it further calculates the planned orders to be released by taking into account the lead time for the component. An example is given below to explain the procedure to be followed for the MRP calculations.

There are a few terminologies, which need to be understood with reference to MRP calculations. The current time is the beginning of the first period. The time periods are also called *time buckets*, and generally taken as one week. However, it can be any value convenient to the plant. With an increase in the computing power, many people are using daily buckets as well. The number of periods in the MRP record is called the planning horizons which simply specify the future periods for which plans are being made.

Table 19.8 gives the requirements of the part as obtained by the explosion of the MPS. The lead time for the part is 1.

Table 19.8 *MRP Calculation sheet - Inputs*

		Planned period							
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Gross requirements		350	320	340	450	310	280	380	380
Scheduled receipts		174							
Available on hand	176								
Net requirement									
Planned order receipt									
Planned order release									

Calculation for planned period 1

$$\text{Available on hand} = \text{available on hand} + \text{scheduled receipts} - \text{Gross requirements}$$

$$= 176 + 174 - 350 = 0$$

$$\text{Net requirements} = \text{Gross requirement} - \text{available on hand}$$

$$= 350 - 176 = 174$$

This is shown in Table 19.9.

Table 19.9 *MRP Calculation sheet—Period 1*

		Planned period							
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Gross requirements		350	320	340	450	310	280	380	380
Scheduled receipts		174							
Available on hand	176	0							
Net requirement		174							
Planned order receipt									
Planned order release									

Calculation for planned period 2 (Table 19.10)

Available on hand = available on hand + scheduled receipts − Gross requirements
= 176 + 174 − 350 = 0

Net requirements = 320 − 0 = 320

Table 19.10 *MRP Calculation sheet—Period 2*

		Planned period							
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Gross requirements		350	320	340	450	310	280	380	380
Scheduled receipts		174							
Available on hand	176	0	0						
Net requirement		174	320						
Planned order receipt									
Planned order release		320							

The same procedure is being continued for all the periods and the same is shown in Table 19.11. This is relatively simple since the lead time is one period.

Table 19.11 *MRP Calculation sheet*

		Planned period							
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Gross requirements		350	320	340	450	310	280	380	380
Scheduled receipts		174							
Available on hand	176	0	0	0	0	0	0	0	0
Net requirement		174	320	340	450	310	280	380	380
Planned order receipt			320	340	450	310	280	380	380
Planned order release		320	340	450	310	280	380	380	

Table 19.12 gives the MRP calculations for another part with the following details.

The lead time for the part is 1.

Lot size = 1000

Table 19.12 *MRP Calculation sheet*

		Planned period							
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Gross requirements		174	320	340	450	450	280	380	380
Scheduled receipts									
Available on hand	1200	1026	706	366	916	466	186	806	426
Net requirement					84	74		194	
Planned order receipt					1000			1000	
Planned order release				1000			1000		

The output from the MRP software is the planned releases but they may not be automatically released. The material planner needs to release the order.

The reasons for adopting MRP in general are the following:

- Demand for component items is dependent on the demand for assemblies of which they are a part. Consequently, dependent time-phased requirements for components can be calculated.
- Lot-sizing, component commonality, etc., give rise to lumpy demand at the component level. Therefore, order point techniques which are generally used in conventional inventory control are invalid.
- A computer makes extensive application of the MRP technique feasible since a large amount of calculations are needed. In the example, we have shown only one component for only 8 weeks. In a conventional company there may be thousands of parts which may need to be planned for a longer horizon, all of which calls for large computational capacity which is manually impossible.

19.6 PRODUCTION ACTIVITY CONTROL (PAC)

The orders to be released by the MRP can be either to the purchase department or to the production department. The production activity control takes care of the actual production activity to be undertaken in the plant. PAC, therefore, describes the principles and techniques used by management to plan in the short term, control and evaluate the production activities of the manufacturing organisations.

Production management typically is considered as a job of 'pushers' or 'chasers' through human interaction and is also, sometimes paraphrased as 'Too Many People Chasing Too Few Jobs'. The actual operations of production activity control, which is also called 'shop floor control', may be classified into three sections:

- Planning
- Execution
- Control

19.6.1 Planning

By far, this is the most complex of the production activity since it deals with a large amount of inaccurate data and has to come up with an optimum schedule of activities to be performed as per the production orders. Based on this, release orders and allocate resources as per the MRP. In order to do this activity, it is necessary first to create a production schedule.

Shop-floor Scheduler There is a large amount of literature pertaining to production scheduling in view of the complexity of the operations involved. In the process of scheduling, the available options are so large that it is impossible to examine all these in order to arrive at the best possible option. The typical inputs for a scheduling operation are routings, operation standards, MRP, available capacity and status of manufacturing resources. Based on the above inputs, the scheduler will generate as output the schedule of activities to produce components when needed in the correct quantity.

While arriving at an appropriate schedule, there are a number of objectives that can be considered. It is not possible to optimise all these objectives at a given time and hence care has to be taken to define the objectives first before attempting a scheduling function. The various objectives or performance measures in scheduling are as follows:

- **Flow Time** The total amount of time spent by the job in the shop is called flow time.
- **Make Span** The total amount of time required for a group of components in the shop is called make span.

- **WIP Inventory** Work-in-process inventory includes all the jobs in the shop which are in waiting queues, moving, getting processed, or assembled.
- **Utilisation** This may be expressed as a percentage of the time actually utilised for productive work compared to the total time available.
- **Past Due** The amount of time by which a job has missed its due date is called past due. Sometimes, it may also be expressed as the percentage of the jobs from among a group that have missed the due date.
- **Total Inventory** The total inventory gives a count of all the items in the inventory. This includes those in the inventory as well as the scheduled receipts.

Since the variables in the scheduling system are so large, there are no analytical techniques suitable for applying to the more general production scheduling problem. As a result, it is very common to use heuristic scheduling rules for the purpose of scheduling. A scheduling rule or a despatch rule allows selecting the next job to be processed from a set of jobs awaiting service. It is also possible to use these rules for the purpose of

- Introducing parts into the system
- Routing parts in the system
- Assigning parts to facilities
- Allocating resources

These rules are heuristic and, as a result, no one rule is found to perform well in all situations. The factors which affect the suitability of a despatch rule are

- The distribution of the processing times of the tasks to be despatched
- The distribution of the job due dates
- The distribution of job arrivals over time
- The performance measures to be optimised
- The nature of the process plans of the jobs (e.g., presence of assembly tasks, number of tasks per job, difficult facilities, etc.)

Some of the despatch rules used in scheduling of job shops are given in Table 19.13. By their very nature, none of these rules are suitable for all the situations. Hence, it is necessary to examine a number of rules for a given situation and the planner will decide based on that output which is good for a given situation.

Once having selected the performance measures to be optimised and chosen the despatch rule to be applied, the scheduling system will create the job sequence list, which can be presented in many forms. Before accepting the created schedule, it is necessary to analyse it to examine its validity. For this purpose, simulation is the most convenient form. The generated sequence of events will be simulated to visually check the performance as well as generate a large number of statistics of the performance for a proper analysis.

One common form of output that is generally used is the *Gantt chart*. It is a visual tool, which graphically displays the machines and jobs in the time frame as shown in Fig. 19.6. The chart shows three machines with different jobs scheduled for the full day. The blank space denotes the machine free and the shaded portion represents when the machine is occupied with a job. Gantt chart is a very convenient visual tool to see the machine utilisation. It is also possible to superimpose two Gantt charts with one representing the scheduled jobs while the other, the actual operations on the machine. This helps in the monitoring of the progress in the shop and will be discussed later.

Table 19.13 *Despatch rules used in production scheduling*

Despatch rule	*Application*
SPT, shortest processing time	The job with the shortest processing time will be allocated at the workstation.
LPT, longest processing time	The job with the longest processing time will be allocated at the workstation.
EDD, earliest due date	The job which has the earliest due date will be processed next.
FIFO, First in first out	The job which has arrived first at the workstation will be scheduled next.
LWR, Least work remaining	The job with the least remaining processing time will be allocated at the workstation.
CR, critical ratio	It is the ratio of remaining time for the job till the due date with the remaining process time for the job. The job with the lowest CR will be scheduled next. Remaining processing time includes all the times such as setup, waiting, moving and processing times. CR < 1.0 means the job is behind schedule and > 1.0 means ahead of schedule.
S/RO, slack per remaining operations	Slack is the difference of remaining time for the job till the due date with the remaining process time. The priority for scheduling is arrived at by dividing the slack with the number of remaining operations to be done on the job. The job with the lowest S/RO will be scheduled next.

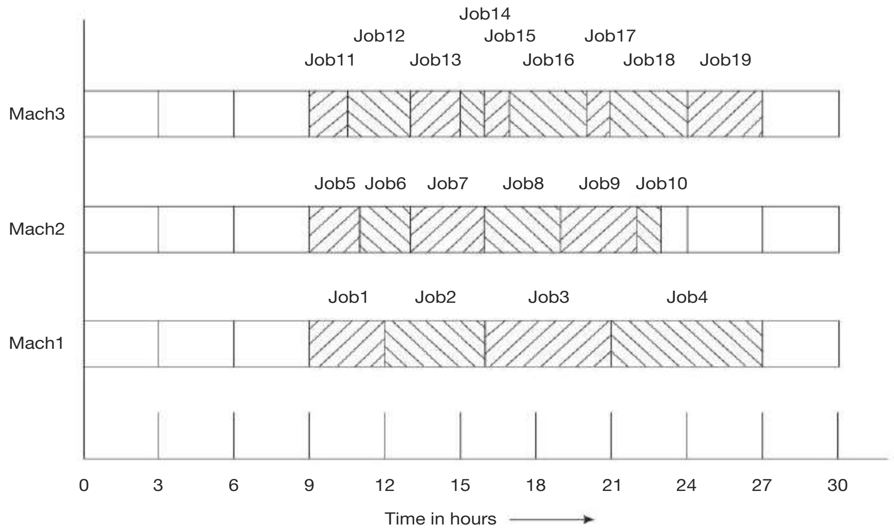


Fig. 19.6 *Typical Gantt chart showing the jobs scheduled on different machines*

19.6.2 Execution

This part of the production activity control is to see that the schedule, which was generated by the scheduler is executed. In order to do this, this module will have to

- Consolidate all the necessary information to execute the tasks that were identified
- Despatch the orders to the shop floor for the purpose of actual production
- Coordinate closely with the status of the shop-floor resources

The execution phase requires that all the information required for the purpose of manufacturing of the parts such as engineering drawings, bill of materials, route sheets, raw material, tools required, etc. ,along with the authorisation to withdraw any material is passed on to the workstation concerned. In addition to this, it may also coordinate actual material handling and storage function (AGV, AS/RS, etc.). Further for any other computer-controlled equipment present in the system, the necessary instructions such as part programs for CNC machines are downloaded (DNC function).

Another function that may have to be done is the necessity for dynamic rescheduling to take care of disturbances within the shop floor. Similarly, any priority changes that need to be incorporated such as the rush orders, can be handled by the rescheduling function.

19.6.3 Control

The major functions served by the part of the production activity control are

- Track actual performance of the work orders released
- Evaluate the production efficiency in terms of work in process, lead times, delays, etc.
- Report to the management about the resource utilisations, productivity and other production related statistics

This module collects the data from the various manufacturing resources such as process times, part status, material availability, inspection data, scrap and rework data, etc., and analyses this data to give real-time feedback to other modules. Thus, it is able to compare whether the schedule given is in control or any corrective action is necessary. The data collection and reporting to be done should be as extensive as possible to keep the schedules and actual production in close control.

Schematically, the total functions of production activity control with the various modules and the databases that are necessary are shown in Fig. 19.7.

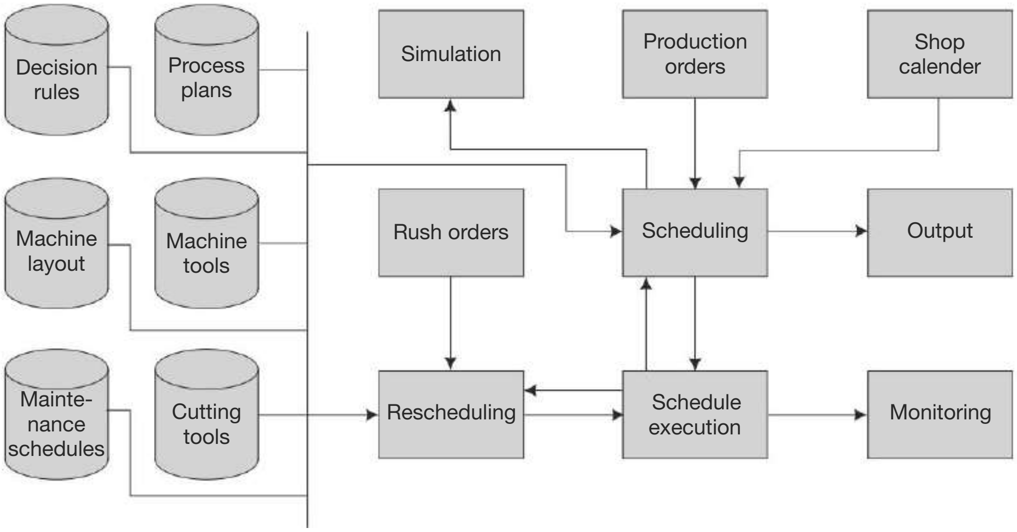


Fig. 19.7 *Typical components of a production activity control function*

19.6.4 Shop-floor Data Collection

There is a need for having a large amount of data that has to be sent to the shop floor as well as receiving a large amount related to the progress. All this information is required to take appropriate shop-floor control decisions. Some of the types of data that is either sent to or generated in the shop floor are as follows:

- Order releases
- Scheduling information
- The jobs or parts that are actually in production
- Number of pieces that have been scrapped
- Labour utilisation
- Machine utilisation
- Machines in production, set-up, idle or breakdown
- Work or production stoppages
- Cost of downtime
- Various other reports to help the management to better plan production

Most of this information can be generated and transmitted electronically. However, there is need for interaction with the labour on the shop floor at some point of time to get or feed information in hard-copy form, for which a number of methods are normally utilised in the shop floor. Some of the methods used for data transmission or collection can be accomplished by having semi-automated data terminals that are located throughout the plant. These can download product data related to materials, process plans, tooling, labour content, etc. At the same time, shop-floor data such as part completion, inspection data, timing, etc., can be collected at this point. These terminals are semi-automatic, since plant personnel interact with them. The degree of automation present in these terminals depends upon the type of overall information-handling protocols and systems practiced in the plant.

Data-Collection Methods

Major automated data input technologies practiced in the shop floor are

- Barcode technologies
- Optical Character Recognition (OCR)
- Vision or image processing
- Radio Frequency Identification (RFID)

The principles of some of these technologies are described here.

(a) Barcode Technologies Barcode technology allows for information input at a very high rate with very little error rate compared to keyboard input. The input method used by barcodes is a simple and inexpensive method for encoding textual information that can be easily read by inexpensive electronic readers. It was invented in 1949 and is now extensively used.

A barcode consists of a series of parallel, adjacent bars and spaces. The symbols used in barcodes follow some specific predefined patterns. They generally consist of spaces, narrow bars and wide bars. The coded information depends upon the sequence of these bars. A barcode reader decodes a barcode by scanning a light source across the barcode and measuring the intensity of light reflected back by the white spaces. A photodiode incorporated into the barcode reader generates the electronic signal which is then decoded by the software present in the reader. The way the barcodes are designed does not make any difference in scanning a barcode from right to left or from left to right.

There are a large variety of barcodes that were developed and used by different users. Code 39 is a general-purpose code and is widely used in the world including manufacturing organisations. The structure of a barcode consists of a leading and trailing quiet zone, a start pattern, one or more data characters, optionally one or two check characters and a stop pattern. The barcode may be of any length, though rarely more than 25 characters are used.

There are a number of barcode reader types that are used in the industry. The type of technology used by each of them is slightly different for reading and decoding a barcode. There are pen-type readers (e.g., barcode wands), laser scanners, CCD readers and camera-based readers.

A photodiode and a light source are placed close to each other in a pen-type reader at the tip similar to a pen. The tip of the pen is dragged across the barcode in a steady motion to read it. The light reflected back from the light source is measured by the photodiode to measure the bar widths and white spaces in the barcode. The scanner then decodes this information.

The operation of a laser scanner is very similar to the pen-type reader explained above except that it uses a laser beam as the light source. A photodiode inside the scanner measures the reflected light to generate the pattern of the barcode. The light emitted by the reader is tuned to a specific frequency as also the photodiode is designed to detect only this same frequency light in both pen readers and laser scanners. The head of a CCD (Charge Coupled Device) reader has a sensor consisting of an array of a large number of tiny light sensors similar to a digital camera. The sensor is able to generate the pattern of the barcode by measuring the reflected light from the code. The CCD reader measures emitted ambient light from the barcode compared to a pen or a laser scanner which measures the reflected light of a specific frequency originating from the scanner itself.

An improvement of the CCD reader is the camera-based reader that uses a small digital camera to capture an image of a barcode. It uses a sensor that has an array of sensing points compared to a row of elements in the case of a CCD reader to generate an image. The software in the reader utilises sophisticated digital image-processing techniques to decode the barcode.

The major element present in any barcode system is the scanner utilising one of the technologies as described above. Stationary models of barcode readers are placed at the fixed locations in the plant depending upon the requirements. These are normally online so that when a data is entered, it is automatically captured by the real-time data-collection system. In addition, there can be a number of handheld scanners used in the plant. These are normally battery powered and either download to the nearest computer or transmit using a FM transceiver.

(b) Radio Frequency Identification (RFID) Radio frequency identification tag, or RFID tag, incorporates a microchip and antenna. The antenna in the tag enables it to transmit the data present in the tag to a reader which without any line of sight requirements like barcode scanners can read it. RFID is not necessarily ‘better’ than barcodes. The main advantage of RFID is that it will be automatically read once it comes into the field of view of the reader. The main benefit provided by the use of RFID is the ability to seamlessly integrate the new information captured by RFID, without disruption, into existing and proven information infrastructure.

19.7 OPTIMISED PRODUCTION TECHNOLOGY (OPT)

As mentioned earlier, scheduling is one of the most difficult and tedious task in a job shop requiring to manufacture a large number of components. OPT was developed in Israel with a proprietary algorithm which allows for better scheduling to take care of particular cases with bottlenecks. The philosophy of Optimised Production Technology is based on bottlenecks in manufacturing. A bottleneck is defined as *a point or storage* *in the manufacturing process that holds down the amount that a factory can produce. It is where the flow of materials being worked on, narrows to a thin stream.*

The ten rules which form the basis for OPT [Brown et al 1989] are the following:

1. The level of utilisation of a non-bottleneck is determined not by its own potential, but by some other constraint in the system.
2. Utilisation and activation of a resource are not synonymous.
3. An hour lost at a bottleneck is an hour lost for the total system.
4. An hour saved at a non-bottleneck is just a mirage.
5. Bottlenecks govern both the throughput and the inventory in a system.
6. The transfer batch may not, and many times should not be equal to the process batch.
   Transfer batch—the lot size from the parts point of view
   Process batch—the lot size from the resources point of view
7. The process batch should be variable and not fixed.
8. Capacity and priority should be considered simultaneously not sequentially.
9. Balance flow not capacity.
10. The sum of local optima is not equal to the optimum of the whole.

More details about Optimised Production Technology can be seen in Brown et al 1989.

19.8 MANUFACTURING RESOURCE PLANNING (MRP II)

Manufacturing Resource Planning (abbreviated as MRP II), or closed-loop MRP, is an integrated production information system that synchronises all aspects of the business. Thus, while planning for production, the total manufacturing resources are taken into account. This means that it involves the interconnected activities of all the following:

- Business planning
- Production planning
- Capacity planning
- Master production scheduling
- Resource requirement planning
- Material requirement planning
- Capacity requirement planning
- Production activity control

A schematic of the operation of MRP II system is shown in Fig. 19.8. All the other related activities are also linked for proper functioning of the total system of MRP II. One of the main advantages of using MRP II is that a large number of options can be tried with the system and the optimum option can be decided.

In spite of the extensive application of MRP II in industries, a large number of problems have been identified during the running of the system. They are given here.

- MRP ignored the question of redesign of the manufacturing process.
- The BOM driven mentality tended to encourage the development of many separated process stages.
- MRP II sought sophistication but has instead achieved complexity in many areas.
- The notion of leaving capacity management to the user never worked very well.
- MRP II has grown too large. Significant modularisation is required.

However, a number of lessons were learned from the large number MRP II implementations in the industries. They are the following:

- The fallacy of applying order-point control systems to dependent demand items.
- Hierarchical planning with multiple levels of representational detail of the manufacturing process (e.g., MPS, MRP and PAC), is a highly effective means of coping with the complexity and variety of manufacturing systems.
- Through a computer and a manufacturing database, the work of people in many different manufacturing functions can be better coordinated.
- The role of the planner is paramount. Planner education and responsibility are essential for system success.

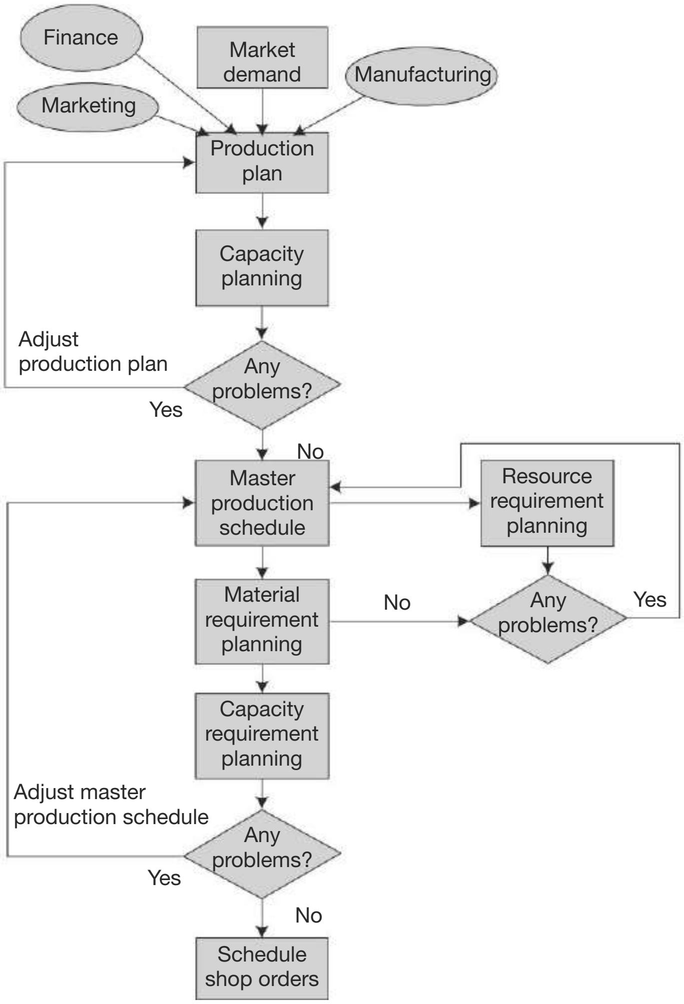


Fig. 19.8 *Manufacturing Resource Planning (MRP II) system*

19.9 JUST IN TIME (JIT)

As the developments in MRP and MRP II are taking place, the Japanese have been quietly developing a number of innovations in manufacturing. The Just In Time, or JIT, is one such methodology that has revolutionised the manufacturing scene the world over.

The APICS dictionary defines JIT as *a philosophy of manufacturing based on planned elimination of all wastes and continuous improvement of productivity.*

The seven wastes that have been identified [Hall, 1987] by the Japanese manufacturers, notably Toyota, that lead to the continuous improvement in production processes are the following:

Waste of Overproduction Eliminate by reducing set-up times, synchronising quantities and timing between processes, compacting layout, visibility, and so forth. Make only what is needed now.

Waste of Waiting Eliminate through synchronising workflow as much as possible and balance uneven loads by flexible workers and equipment.

Waste of Transportation Establish layouts and locations to make transport and handling unnecessary if possible. Then rationalise transport and material handling that cannot be eliminated.

Waste of Processing Itself First question why this part or product should be made at all, then why each process is necessary. Extend thinking beyond economy of scale or speed.

Waste of Stocks Reduce by shortening set-up times and reducing lead times, by synchronising workflows and improving work skills, and even by smoothing fluctuations in demand for the product. Reducing all the other wastes reduces the waste of stocks.

Waste of Motion Study motion for economy and consistency. Economy improves productivity and consistency improves quality. First improve the motions, then mechanise or automate. Otherwise there is danger of automating waste.

Waste of Making Defective Products Develop the production process to prevent defects from being made so as to eliminate inspection. At each process, accept no defects and make no defects. Make processes fail-safe to do this. From a quality process comes a quality product—automatically.

How well this is practised can be gauged from a survey of FMS in United States and Japan conducted in the mid-1980's by Jai Kumar, shown in Table 19.14. This table compares the manufacturing systems between US and Japan and shows the flexibility and economy achieved by the Japanese manufacturers. It can be seen that Japanese manufacturers were able to produce 93 different parts per system compared to that of 10 in the case of US manufacturers. Further, the machine-utilisation rate of Japanese manufacturers is a high of 84% compared to 52% for US. That explains the economy achieved by the Japanese manufacturers using the JIT principles.

Thus, the goals for achieving the Just-In-Time manufacturing are

- Zero defects
- Zero set-up time
- Zero inventories
- Zero handling
- Zero breakdowns
- Zero lead time
- Lot size of one

Table 19.14 *Comparison of US and Japanese FMS use [Jai Kumar 1986]*

	United States	*Japan*
System development time, years	2.5 to 3	1.25 to 1.75
Number of machines per system	7	6
Types of parts produced per system	10	93
Annual volume per part	1727	258
Number of parts produced per day	88	120
Number of new parts introduced per year	1	22
Number of systems with untended operations	0	18
Utilisation rate, two shifts	52%*	84%*
Average metal cutting time per day-hours	8.3	20.2

* Ratio of actual metal cutting time to available time for metal cutting

To achieve these goals, the key elements that one should follow in JIT approach are the following:

High Quality JIT systems eliminate all defects, which eliminates the scrap and rework that help to provide a smooth flow of materials through the plant. The quality is built into the process and is controlled by workers acting as their own quality inspectors.

Small Lot Sizes In conventional wisdom, inventories are used as buffers for inefficiency in production systems. In JIT, small lot sizes are practised which give the benefits of lower work in process inventory, lower lead time and uniform operating-system workload.

Set-up Time Reduction Lower lot sizes calls for larger number of set-ups and hence it becomes necessary to reduce the set-up time.

Manufacturing Methods Use group technology with product families and flow-based manufacturing that simplifies the material flow patterns. Use production-flow analysis and develop flow-based organisation.

Product Design Design the products for ease of manufacturing and assembly. Increase product variety while reducing the process variety. Use modular design such that the same modules can be used in a number of products. This allows for large batch manufacturers bringing in the economy of scale. As far as possible, use off-the-shelf components that provide high quality at low cost and lower lead time. Design products with ease of automation such that mechanisation is possible. Design customised products to a mass market at an economic cost.

Relationship with Suppliers In a JIT environment, build long and enduring relationships with suppliers. Treat suppliers as long-term allies and give them access to information. Also, try to support JIT suppliers with technical information and research.

Summary

- Production planning and control is a part of activity in manufacturing that integrates all the functions that require to be executed in an integral fashion to optimise the resources and achieve economical manufacturing.
- The production plan is generated from the business plan of the management for the organisation. It aggregates all the inputs from the various departments to come up with the net requirements for the business plan so that a master production schedule can be developed.

- The goal of capacity planning is to transform the manufacturing requirements, as set forth in the MRP stage, into a detailed machine-loading plan for each machine or group of machines in the plant.
- The Master Production Schedule (MPS) is the main starting point of all manufacturing operations. This represents the plans of the enterprise for manufactured products in specific configurations, quantities and dates.
- Developing an MPS that is reasonable and realisable within the resource constraints is one of the important elements of a successful organisation.
- Using the MPS, Material Requirement Planning (MRP) produces the requirements of materials to achieve the required production as envisaged in the MPS.
- MRP has to link with the external vendors as well as in-house production facilities to meet the demand of materials as outlined in the MPS.
- MRP also interacts with the bill of materials to explode the MPS into material requirement, as well as the inventory management system to keep track of the storage and lead times for procurements to plan the appropriate strategies and times to order material to satisfy the demand.
- Production Activity Control (PAC) deals with the actual operation of the shop floor and methods that are used to organise and control the shop floor.
- PAC can be modularised into planning, execution and control stages to achieve its objectives.
- Planning part of PAC prepares a schedule of shop-floor operations on a continuous basis to optimise the various indicators of its performance such as work in process, flow time, make span, etc.
- A large amount of information flows in and out of a shop floor. There are a number of methods to automatically capture that information for which some of the technologies used are the barcodes and RFID.
- Optimised Production Technology (OPT) utilises the concept of bottlenecks in manufacturing to integrate the factory operations.
- Manufacturing Resource Planning (MRP II) is a comprehensive system that tries to integrate all the modules in a manufacturing operation to make the information available instantaneously wherever it is needed.
- Just In Time (JIT) is a philosophy that tries to enunciate principles to eliminate the various types of waste that is part of a manufacturing operation.

Questions

1. Give a block diagram indicating the various inputs required for aggregate production planning.
2. Give a brief description of aggregate production planning in a manufacturing organisation.
3. Give a brief description of capacity planning in a manufacturing organisation.
4. Give a description of the objectives of master production scheduling.
5. Define master production scheduling. Explain the factors that need to be taken into account while developing a master production schedule.
6. Give an example how a master production schedule can be developed with (a) an economical lot size of one, and (b) an economical lot size of a small batch (greater than one).
7. What are the objectives of materials requirement planning?

8. Give a block diagram showing the interaction between materials requirement planning and other modules of a production planning system.
9. Briefly explain the functions served by production activity control.
10. How can you decide the planning function in production activity control?
11. What do you understand by the despatch rule? Give some examples of despatch rules used in scheduling with their applications.
12. Why are the heuristic rules used in shop-floor scheduling?
13. What are the functions served by the control function in production activity control?
14. Write a short note on Optimised Production Technology.
15. What is MRP II? How does it benefit the management of a factory?
16. What is the philosophy of Just In Time (JIT)?
17. What are the seven wastes in manufacturing plants? Give a brief description.
18. In order to achieve the goals of JIT, give the approaches to be followed.

Part – V

INTEGRATION OF MANUFACTURING SYSTEMS

20

COMMUNICATIONS

Objectives

A CNC machine tool will not run in isolation. There can be more than one CNC machine tool in a shop. Also it will be interacting with other machine tools as well as other manufacturing machines, such as the material handling equipment. Under such circumstances, it is necessary for the CNC machine tool to maintain communication with other units for smooth operation of the plant. After completing the study of this chapter, the reader should be able to

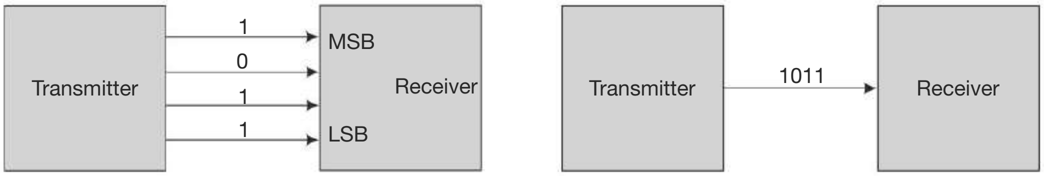
- Understand the basic communications used in computing equipment
- Learn about the direct numerical control (DNC) as used in CNC machine tool communication
- Appreciate the various functions that can be enhanced by the use of DNC and enhanced DNC functions
- Understand the advantages of using DNC
- Learn about the various communication standards that are pertinent to manufacturing systems

20.1 COMMUNICATION METHODS

Data communication is done by transmitting logic 0 and logic 1 signals in the form of high and low voltages through wires between the two interconnecting devices. Any typical computing device (the MCU is also a computing device in view of the presence of the microprocessor inside) communicates with the outside devices in the following two modes.

- Parallel transmission
- Serial transmission

As the name implies the parallel transmission involves the movement of information between the two devices in parallel using a large number of individual wires as shown in Fig. 20.1(a). Parallel interfaces are used for high rate of data flow than is possible with serial data transmission. In the case of parallel transmission, the number of wires connecting will be based on the number of data bits being sent in parallel, such as 8, 16 or 32 bits. As such there have to be different standards for each unlike the serial transmission where there is a single accepted standard RS 232C. In view of the multiple wires being used in parallel transmission, the distances between the communicating equipment has to be small to reduce the cost of the cabling.



(a) Parallel transmission

(b) Serial transmission

Fig. 20.1 *Communication methods between computing devices (LSB—least significant Bit, MSB—most significant Bit)*

In the case of serial transmission, there is a single wire that sends data from one terminal to the other as shown in Fig. 20.1 (b). As a result, the data will move in single bits. However, the data is normally stored in parallel in the computer or MCU in the form of words of length 8, 16 or 32 bits. Therefore the serial transmission will have to convert that data into single bits and then send. At the receiving end, these bits will have to be reassembled to make the necessary word of 8, 16 or 32 bits.

To facilitate the transmission between the two devices, there need to be some form of communicating method to establish the mutual readiness to communicate and receive data which is termed as hand shaking. Typical hand shaking methods used in the devices are shown in Fig. 20.2. The handshaking signals are self-explanatory.

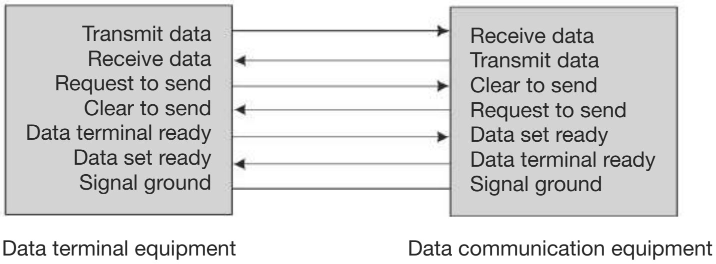


Fig. 20.2 *Handshaking methods between communicating devices*

When the data is being sent as individual characters (using for example the ASCII, American standard code for information interchange code), it is necessary to communicate with the receiving equipment as regards when the character is starting and when it is ending. This is done with the help of a single start bit in the beginning and 1 or 2 stop bits at the end of the actual character transmission. So the full transmission of a single character requires a start bit, data bits (5 to 8), parity bit (odd, even or none to error check the correctness of data transmission) and stop bits (1, 1.5 or 2). This is schematically shown in Fig. 20.3. The actual data shown in Fig. 20.3 is for the character S (1010011), with seven data bits, even parity and one stop bit.

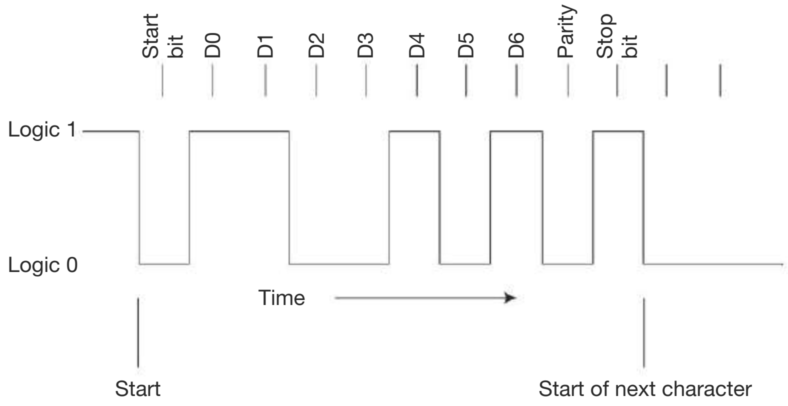


Fig. 20.3 *Serial transmission protocol*

In order for the receiving end to count the bits correctly it is necessary to establish the speed with which the bits are being transmitted, which is termed as baud rate and is measured in bits per second. The baud rates that are generally used are 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, etc.

The accepted standard for serial transmission is the RS 232C of the Electronics Industry Association or EIA. The pin configuration of the connectors used as per the RS 232C is shown in Table 20.1. Though there are a large number of signals identified in the standards, all these are rarely encountered. The actual pins used in typical transmission between MCU and PC are shown in Table 20.2.

Table 20.1 *Standard pin configuration for RS 232C*

Line/Pin number	*Circuit description*
1.	Protective ground
2.	Transmitted data (TD)
3.	Received data (RD)
4.	Request to send (RTS)
5.	Clear to send (CTS)
6.	Data set ready (DSR)
7.	Signal ground
8.	Primary carrier detect
9.	Test circuit
10.	Test circuit
11.	Unassigned
12.	Secondary carrier detect
13.	Secondary clear to send
14.	Secondary transmitted data
15.	Transmission signal timing
16.	Secondary received data
17.	Receiver signal timing
18.	Unassigned

Contd..

Contd..

19.	Secondary request to send
20.	Data terminal ready (DR)
21.	Signal quality detector
22.	Ring indicator
23.	Data signal rate selector
24.	Transmit signal timing
25.	Unassigned

Table 20.2 *Actual pins that are generally used in RS 232C connections*

Line/Pin number	*Circuit description*	*Circuit abbreviation*	*Signal direction*	*Signal function*
2	Transmitted data	TD	From DCE:	Data transmission
3	Received data	RD	To DCE:	Data transmission
4	Request to send	RTS	To DCE:	DTE wants to transmit
5	Clear to send	CTS	From DCE:	Transmission approved
6	Data set ready	DSR	From DCE:	Unit is operational
8	Primary carrier detect	CD	From DCE:	Modem carrier present
20	Data terminal ready	DR	To DCE:	Unit is operational

The signal ground, Pin 7, is used as the reference point for zero volts between the two circuits. The protective ground (Pin 1) connects the two chassis of the DTE and DCE to protect the user from electrical shock similar to the earth pin of a common 3 pin plug for electrical appliances.

Generally the length of the cable connecting the two devices is limited owing to the loss of data and the possibility of noise. When the data needs to be transmitted over long distances, then the signal may be converted to suitable form so that the common telephone lines could be used for the transmission. This is done with the help of a modem *(modulator–demodulator)* as shown in Fig. 20.4. The modem converts the digital data into analog oscillations of two distinct frequencies for communicating over the telephone lines which are generally used for transmitting voice. At the receiving end the modem demodulates and converts it back to digital 0s and 1s.

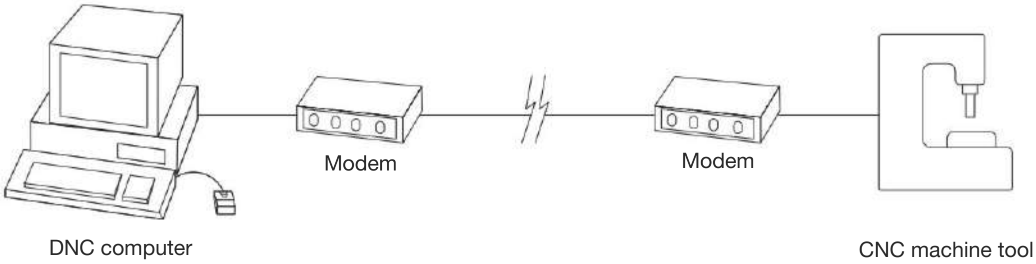


Fig. 20.4 *Typical transmission using a modem over long distances*

20.2 DIRECT NUMERICAL CONTROL

Direct Numerical Control (DNC) is the method used in the shop floor to connect a number of CNC machine tools together and controlled by a single computer (PC). Here when we say controlling, it means that the MCU of each of the CNC machine tool is directly under the control of the DNC computer as shown in Fig. 20.5, while the individual machine tool is controlled by its MCU.

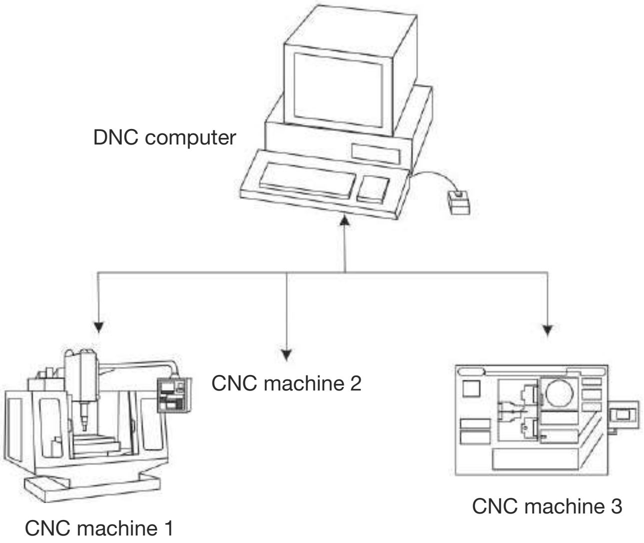


Fig. 20.5 *Typical direct numerical control operation*

Typical functions served by the DNC computer can be categorised into the following three groups.

- Communications
- Part program storage
- Ancillary functions

Communications All those functions wherein the communication takes place between the CNC machine tool and the DNC computer are put in this category. They are the following.

- Sending part programs to the CNC machine tool as and when required, which is normally termed as part-program downloading.
- Receiving the part-programs from the CNC machine tool after it has been edited or prepared on the machine control unit, which is normally termed as part-program uploading.
- Provide a remote program buffer. Generally, the part-program memory size is limited in most of the controllers which is generally less than 1 MB. In case of part programs generated by the CAD/CAM systems particularly in the case of finishing cuts of 3D profiles such as those used for dies and mould cavities, the part programs are generally very long often going beyond 1MB. In such cases, since the part program cannot be fully loaded into the controller memory, the program needs to be split into parts. However, when a program is split, there is a possibility of burr left at the point where the tool finished in the first program and restarted in the second. This is not conducive for the clean finish of

the die. In such cases the program will be fed in parts into the controller, termed as drip feed. The MCU starts executing the program blocks that are already received by the controller without looking for the end of the program. The MCU will flush the blocks of program that were executed by it, thus vacating its part program memory. The DNC computer will send the next segment of the program into the MCU before the complete execution of all the blocks present in the MCU. Thus the machine tool will not have to starve for the program and the tool will continuously be cutting assuring good finish. However for the drip feeding of the program, normally it uses only the motion commands G00, 01, 02 and 03. Also the controller should have the capability of accepting the drip feeding.

- The tool offsets of the tools to be used in the current program can be down loaded into the MCU. This eliminates the necessity for the operator to use the MCU for data entry, which is generally a tedious process and not ergonomical.
- The CNC machine tool can be operated directly from the DNC computer, again eliminating the operator to go to the MCU.
- To bring the CNC machine tool to its datum position remotely.

Part-Program Storage There are a large number of functions that can be carried out by the DNC computer in terms of the part program manipulation that cannot be done inside the MCU. They are the following.

- It can store all the part programs meant for all the CNC machine tools. Theoretically, it is possible to store a large number of part programs since the disk storage cost is relatively small.
- It is possible to input, edit, copy and delete programs at the DNC computer. The type of facilities provided will be with the familiar look and feel of Windows operating system, so that the operation is relatively easy and faster than possible with an MCU.
- Editing function can be provided based on blocks rather than the paragraph (line feed) as is normally used in the conventional editors. This greatly facilitates the editing process of CNC part programs.
- One of the more common editing function done on the part programs is the removal and renumbering of block numbers, which can be easily done in the DNC computers.
- There are times when the same job is to be made on different machine tools, in which case the programs will be different. Instead of storing the different versions of the same part program, the CLDATA can be stored and post processed for the particular MCU when it is to be downloaded. This conserves the storage space of the DNC computer. This also provides for greater flexibility for shop floor scheduling.
- It is also possible to provide syntax checking (to check the correct usage of the word addresses in the part program) and graphic proving of the part programs on the DNC computer.
- It is also possible to provide a calculation facility for estimating the machining times of the various part programs.
- It can be used as a part program preparation system as well. For this purpose the system can provide a certain number of help facilities that can be used during the part program preparation process.
- Geometric help
- Calculator help
- Controller (G-codes and M-codes) and machine tool function help
- It can manage the tool offsets. Complete tool library can be maintained by the DNC computer along with the offsets, so that they can be assembled and then down loaded into the MCU as and when required.

- It can function as a machinability database for calculating the feeds and speeds for the various tools used by the part programs.
- Complete tool management functions can also be embedded into the DNC program. It is possible to carry out the tool life management function as described earlier in a DNC program.

Ancillary Functions

- Integrating with the CAD/CAM system or the part programming system for the purpose of direct transfer of the part programs into the disk storage of the DNC computer.
- Possibility of integrating with the shop floor control system for proper integration purpose, which is discussed in greater detail later in the form of enhanced DNC.
- Getting the data from the machine control unit, the information about the health of the machine in the form of sensor signals or diagnostic messages, which can be used for predictive/preventive maintenance purpose (Fig. 20.6).

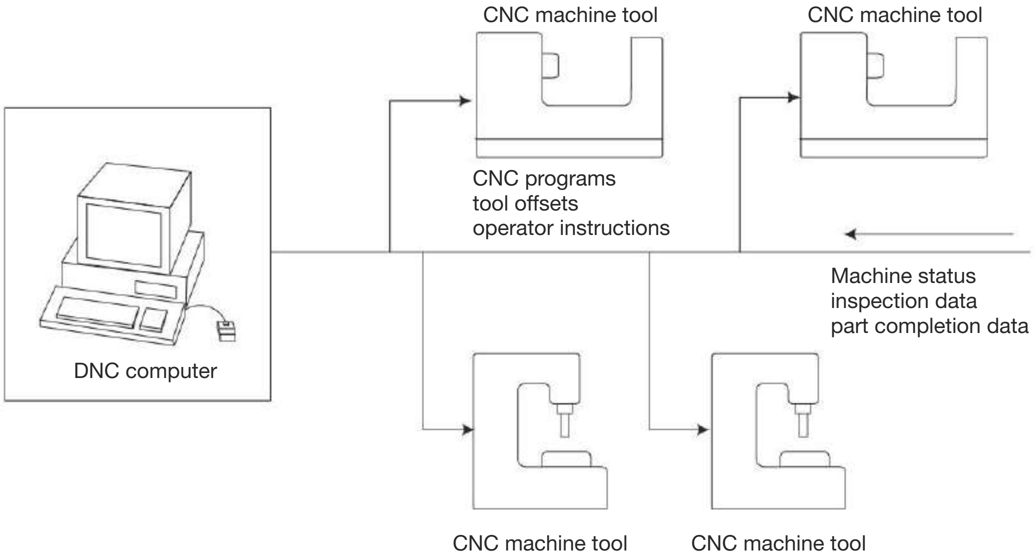


Fig. 20.6 *Typical direct numerical control operation with enhanced functions*

- Getting the data from the machine control unit, information about the parts that are completed in terms of part numbers, quantity, etc.
- If probes are used in a given program, then the metrological data in terms of the dimensional acceptance can also be uploaded from the MCU.
- Also getting the data from the machine control unit, the information about the tools and the lives that were used up for proper tool management system.

20.2.1 Typical Installations

As discussed earlier, the serial communication port available in the personal computer is generally used for linking the CNC machine tool for DNC function. All the modern machine tools are provided with a serial port or RS 232C port so that it can be directly connected as shown in Fig. 20.7.

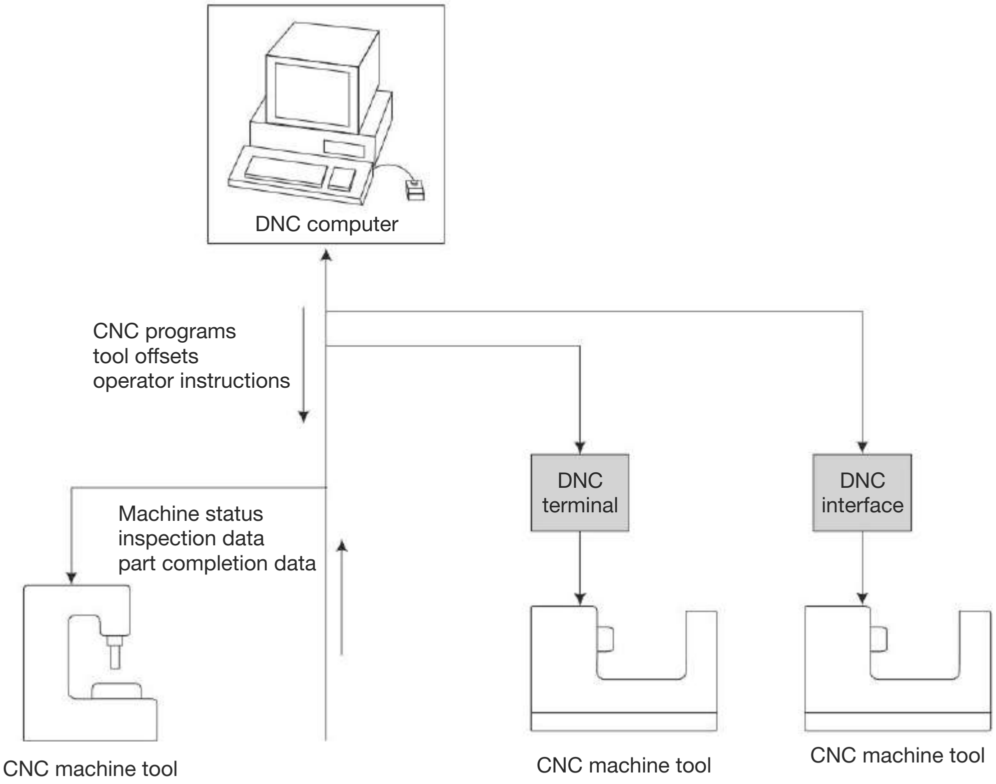


Fig. 20.7 *Typical direct numerical control operation*

Older NC and CNC controllers that have little or no internal memory or communications capability are difficult to be integrated into a DNC system. Such controllers then have to be provided with a special interface unit, which normally can be plugged into the tape reader port available in such controllers. Such systems sometimes are called the Behind the tape reader or BTR systems. Once the system is installed even these older NC controls can get integrated into a network as shown in Fig. 20.7. They can then run long part programs that would have required miles of paper tape. All other part program management, editing and other functions found only on newer CNC controls can be done either on the interface terminal or on the DNC terminal.

When a normal PC under Windows operating system is used, the number of serial ports available for direct connection to the CNC machine tool are limited to four (COM1 to COM4). When more than four machine tools are to be connected then a multiplexer may have to be utilised to connect more than one machine tool to a single serial port. A multiplexer is a mechanical or electronic switch through which a single serial port can be shared by a number of CNC machine tools. This allows for the sharing of the computer and is commonly practiced for downloading part programs. However, when drip feeding is to be done it is not possible to multiplex, since the time slicing may not be possible without starving the machine tool. In such cases special intelligent cards are available with large number of serial ports that can sit directly into the free slots available on the motherboard of the PC.

Sometimes a dedicated shop terminal can be located for each of the machine tool which can then be linked through a network with the DNC computer as shown in Fig. 20.7. In such case the number of ports is not a limitation. The terminal is generally a very simple computer, which runs a component of the DNC program for direct linking with CNC machine tool. It can be a low end computer with a small LCD display and memory.

20.2.2 Requirements

In order to use DNC there has to be a certain discipline enforced in the preparation of the part programs to help with the part program management. Following points are highlighted which need to be enforced for a successful implementation of DNC in a shop.

- It is a general practice to use the part program number to classify the program. For this purpose a standard procedure needs to be followed to classify the program in terms of the machine tool on which it is to be used, the tools required, etc.
- Each of the part programs should be provided with a header which can provide all the necessary information in terms of the part number, drawings, cutting tools, fixtures, machine tools, revision data, etc.
- The operator instructions should be provided in the part program where necessary. The use of comments (using parentheses) can be made use for this purpose.
- The basic instructions for communication between the MCU through the operator and the DNC computer should be simple enough so that they can be handled easily by the shop personnel who may not be very well versed with computers.
- At the time of planning the DNC installation care has to be taken to see that the requirements of the individual machine tools for better utilisation of the resources are met.
- Similarly the cabling when planned should be such that in case of future expansion it should be possible to add more machine tools or change the DNC requirements.

The use of DNC involves additional cost in terms of the network, computer, terminals and the software. Therefore, it is necessary to see when DNC will be useful. The use of DNC can be justified under the following conditions.

- When the interconnected CNC machines are large in number. With smaller number of machine tools it is easier to handle the machines directly unless the drip feeding becomes necessary such as in tool rooms.
- When the part program sizes are very large and cannot be held in the part program memory of the MCU.
- Part program variety is large and batch sizes are small. In such cases the machine tools need to be fed by a large number of part programs in a day which can easily justify the use of DNC.
- Very frequent changes in program designs require that programmers spend more time at the part programming terminal which is easier than the MCU.

It is generally expected that there will be a 2 to 5 per cent increase in the operational efficiency of the CNC machine tools with the use of DNC. If the typical cost of a DNC installation is about Rs. 5 00 000 for a 10 machine configuration, then it is possible to recover the cost of the installation in less than a year's time assuming a machine's hourly rate of Rs. 300 with a two-shift operation.

20.2.3 Enhanced DNC

As mentioned earlier, DNC provides for better utilisation of the CNC machine tools by providing the part programs as and when required. However, the running of the CNC machine tool does not depend upon the part programs alone. It also requires the necessary tools to run the particular part program as well as the blank work pieces in position for running the said programs. The choice therefore falls on the ability of the DNC computer to maintain the interaction with other facilities within the shop such as the work preparation area as well as the tool crib.

It is therefore possible to visualise a better DNC model, which we may call as the *Enhanced DNC* or *Area Manager* and is schematically shown in Fig. 20.8.

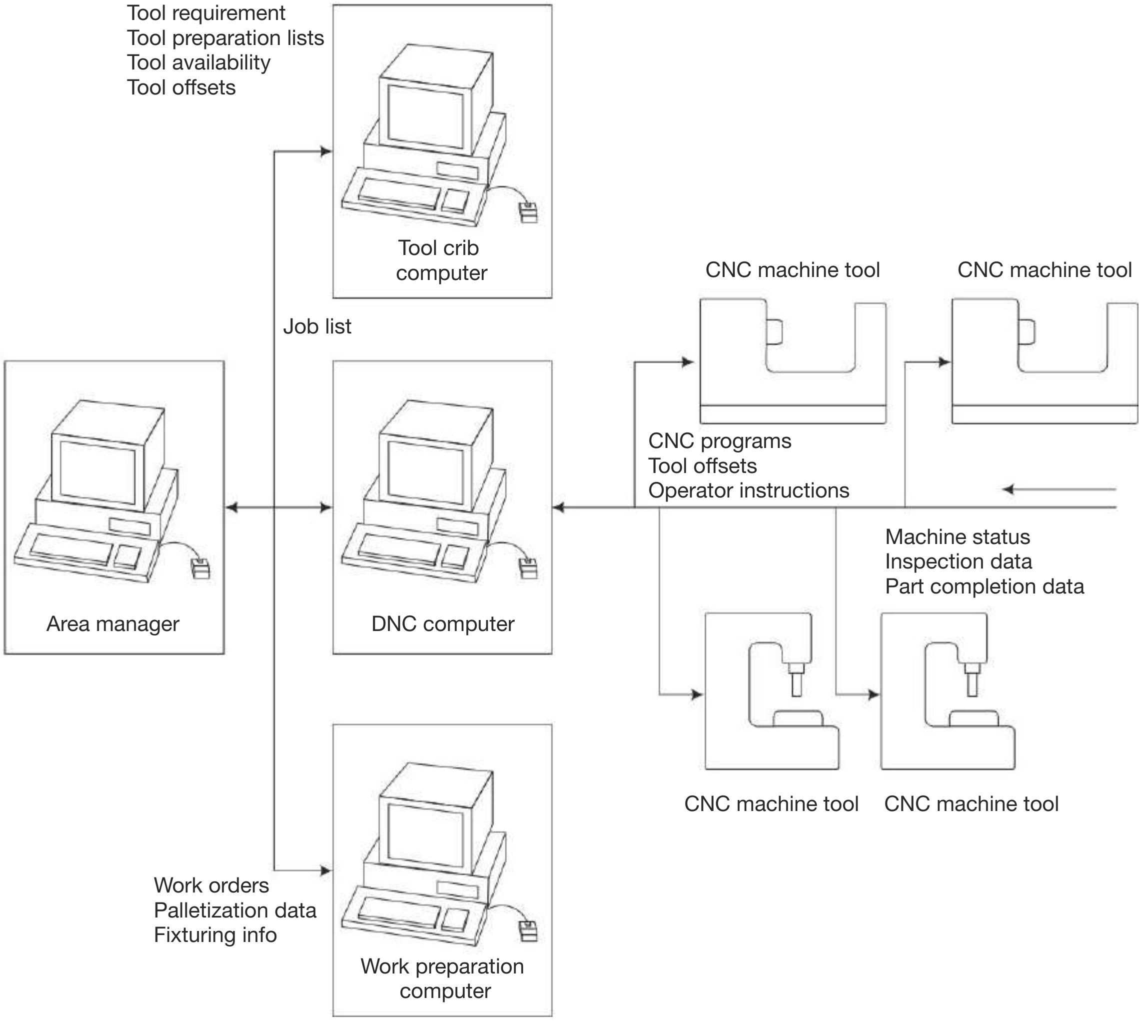


Fig. 20.8 *Enhanced direct numerical control operation linking the tool and work preparation areas with CNC machine tools*

In this scenario, it is possible to visualise a separate terminal at each of the important areas, viz., the work preparation area and the tool preparation area shown as tool crib in Fig. 20.8. At each of these terminals the operators will be able to input the data necessary for integrating into the system.

The area manager will send the following information to the work preparation area.

- Work orders, the part numbers of the planned jobs with quantity.
- Fixturing information that gives the details of fixtures used as well as instructions for fixturing.
- Palletization data in terms of the pallet numbers to be used for the individual parts along with the pallet offsets if any.

Work preparation area completes the preparation of the blank workpieces and clamps the workpieces on the pallets or work holding fixtures and then transports them to the respective destination with information back to the area manager.

Similarly the area manager will send the following information to the tool crib.

- Tool requirements in terms of the actual tools that are required for the jobs that are scheduled for the day.
- The tool setting up instructions in terms of the type of adopter, tool holding method to be used, etc.

The tool crib may come back to the area manager with information on the availability of the tools, the tools that are assembled, the tool offsets and the tool lives which could be used for the tool life management.

Thus the area manager will be coordinating the activities of all the three components of the total manufacturing system concerned thereby, improving the productivity of the total system. At the upper end, it is possible to have the area manager linked to the factory production control system to provide the necessary feedback.

20.2.4 Advantages of DNC

1. DNC helps in the elimination of the local input device such as the tape reader. This helps in the entry of error free part programs. For very long part programs paper tape becomes very inconvenient to use.
2. The large storage capacity of the DNC computer makes it possible to store several part programs.
3. If the same job is to be made on different machine tools then separate part program entry into the individual controllers is eliminated reducing the duplicity of work. Instead of storing the different versions of the same part program, the CLDATA can be stored and post processed for the particular MCU when it is to be downloaded.
4. DNC software helps in managing the part programs in a better and easier way.
5. Because of management function software, the shop floor running of machines, inventories of tooling, etc. can be handled much better. Further, real time rescheduling of machines provides lot of flexibility to ensure lower idle times and therefore greater productivity. This is greatly facilitated by the fact that in DNC the part programs are generally stored in the form of cutter location data (CLFILE) rather than post processed programs for individual machine tools. This storage of part programs in general format provides easier real time rescheduling of several NC/CNC machine tools. The management software also makes it possible to prepare regular manufacturing reports quickly for “higher level” decision making.
6. DNC provides the starting point for factory integration. With the experience gained, it is possible to extend the integration into the various aspects of manufacturing ultimately leading to CIM as shown in Fig. 20.9.

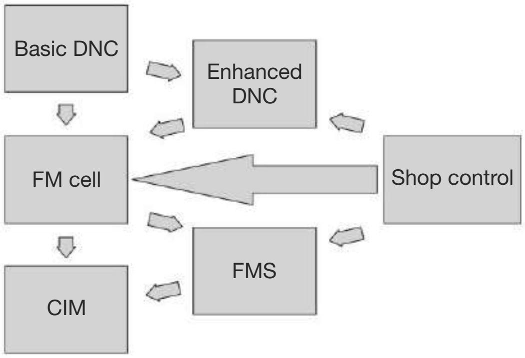


Fig. 20.9 *Various paths possible for CIM starting from direct numerical control*

20.3 COMMUNICATION STANDARDS

It is a fact of life that in a manufacturing organisation a large variety of machine tools are used which are made by different manufacturers having different control systems. Same thing happens with the other devices being used. In view of the heterogeneity of the various systems used it becomes extremely difficult to integrate all of them unless a proper understanding of all the various interface details. To tackle such problems it is necessary to standardise the way the interfacing could be done which paves the way for easier interfacing of the heterogeneous equipment used in the manufacturing organisations.

Some of the major components in the manufacturing systems that need communication are the following.

- CAD/CAM work stations from different vendors
- Personal computers
- CNC controllers
- Controllers of flexible manufacturing cells
- Material handling devices like robots and AGVs
- Programmable logic controllers (PLCs)

In 1980 General Motors (GM) experienced difficulties in getting the vendors to agree on specific standards. Also the standards were so broad that it had become very complex and hard to develop hardware and software for them, thus driving up the costs. In the GM plants there were thousands of computer controlled equipments which have become a nightmare to integrate. The fact is that no single vendor could meet all the needs and multiple vendors caused communication incompatibilities.

Realising the problems GM decided to begin the development of a networking protocol for the high data rates expected, while improving noise immunity for such computer controlled manufacturing environment. This scheme was expected to provide a common standard for all equipment to simplify integration. The aim of the task force is to develop an open heterogeneous factory communication network based on OSI model for all the processes involved in the manufacturing as detailed above.

The reference model for communication within open systems was established by International Standards Organisation (ISO) in 1983 in the form of Open System Interconnection (OSI) model as shown in Table 20.3. This is the basis for the architecture of many proprietary communication networks.

Table 20.3 *Reference model for Open System Interconnection (OSI) model ISO DIS 7498-1983*

Layer No. and Name	*Communication functions (or protocol provisions)*
7: Application layer	To provide protocols for exchange of information between application processes. Provides all services directly comprehensible to application programs.
6: Presentation layer	To provide for representation of data referred by application processes or communicated between them. Restructures data to/from standardised format used within the network. (e.g. ASCII, EBCDIC, ...) - data formats, data coding, ...
5: Session layer	To provide for the organisation and synchronisation of interactions between application processes and management of their data.
4: Transport layer	To provide for transparent data transfer between end systems. Provides reliable data transfer from end node to end node.
3: Network layer	To provide routing and relaying of data through intermediate devices if any. Useful for connecting WANs for interconnecting a number of LANs.
2: Data link layer	To provide for direct transfer of data between directly connected devices and check errors at the lower level. Improves error rate for frames moved between adjacent nodes.
1: Physical layer	To provide for transparent transmission of bit streams between systems or devices (mechanical/ Electrical linkages and data protocols). Encodes and physically transfers bits between adjacent nodes. Cables, modems, etc.

In 1988 GM released the network architecture called Manufacturing Automation Protocol (MAP) 3.0 to be complied by its suppliers. The implementation of OSI for MAP 3.0 is shown in Table 20.4.

Table 20.4 *MAP application of the OSI reference model-MAP 3.0 released in 1988*

Layer	*MAP communication protocols*
7: Application	ISO-MMS (Manufacturing Message Specification) ISO 9506. MMS provides the following important functions. • Variable access • Message passing • Resource sharing (synchronisation) • Program management • Event management MMS follows the client-server principle. MAP provides user interfaces which make the application programs independent of the implementation details of the network services. File transfer protocol ISO FTAM DIS 8571 Real time Manufacturing Messaging Format Standard (MMFS) ISO CASE KERNEL - ISO 8649/1-2, DIS 8650/1-2 MAP – Directory services MAP – Network management
6: Presentation	ISO Presentation DIS 8823.
5: Session	ISO Basic Combined Subset and Session Kernel: ISO Session ISO 8327 full duplex.
4: Transport	ISO Transport ISO 8037 Class 4.
3: Network	ISO connectionless Internet: ISO Internet DIS 8473.
2: Datalink	ISO Logical Link Control uses the connectionless service ISO 8802/2 (IEEE 802.2). ISO-Token Bus ISO 8802/4 for medium access method.

Contd..

Contd..

1: Physical	IEEE 802.4: 10 Mbps broadband ISO 8802/4 Token passing bus Medium Access Control. This uses the coaxial cables which are widely used in the US factories including GM. Since MAP uses only three channels with a bandwidth of 12 MHz, other signals or networks can make use of the same cables. This can cover relatively large distances up to 10 km. IEEE 802.4: 5 Mbps carrier band communication for low level real time data as a low cost option with up to 32 stations and a maximum distance of 700 m.

MAP allows for direct linking of two MAP compliant (following the standards as above) devices to be integrated seamlessly. However there are still many loose ends to be tied up to get to that stage and efforts are going to remove as many problems as possible. In Fig. 20.10 is shown a possible example of a MAP installation.

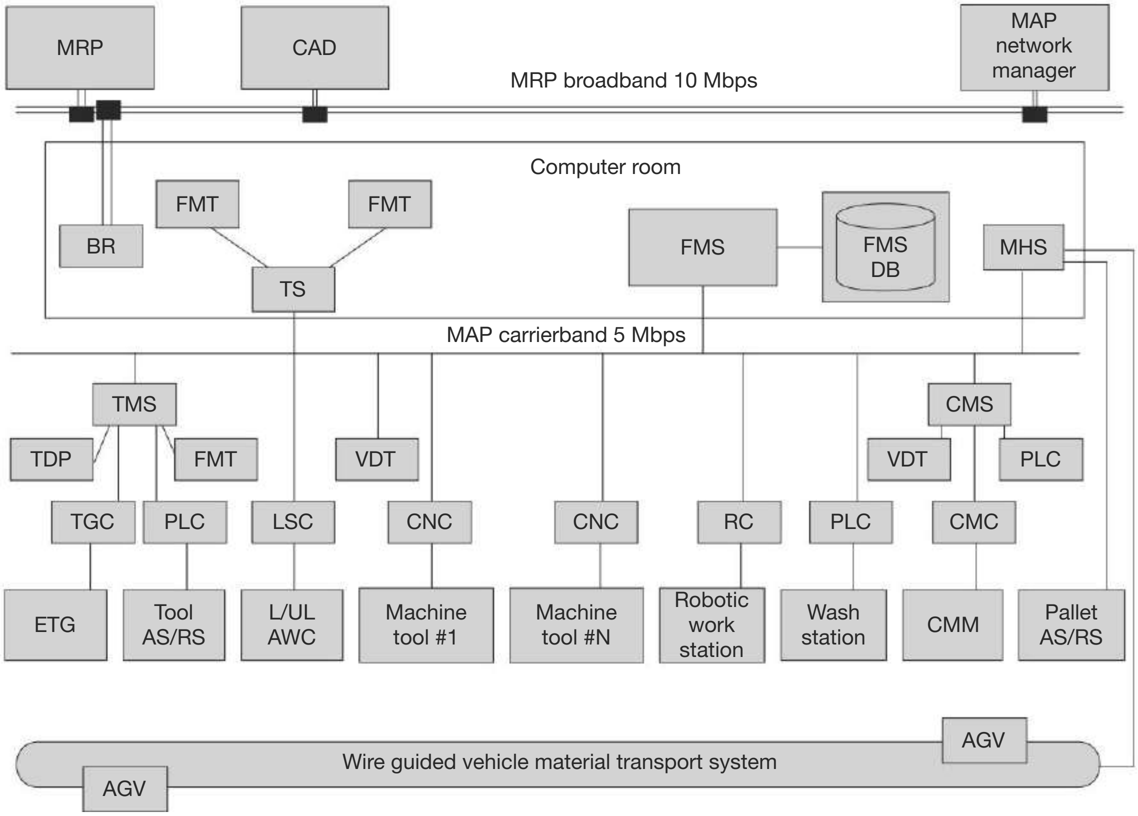


Fig. 20.10 *Typical MAP installation linking the FMS with that of the other Information Systems*

The abbreviations used in Fig. 20.10 are explained below.

MRP = Material Requirements Planning
CAD = Computer Aided Design
BR = Bridge
FMT = FMS Terminal
TS = Terminal Server
FMS DB = FMS Database
MHS = Material Handling System
TMS = Tool Management System
CMS = Coordinate Measuring System
TOP = Tool Data Panel
TGC = Tool Gage Control
PLC = Programmable Logic Control
LSC = Load Station Controller
VDT = Video Display Terminal
CNC = Computer Numerical Controller
RC = Robot Controller
CMC = Coordinate Measuring Control
ETG = Electronic Tool Gage
AS/RS = Automatic Storage/Retrieval System
L/UL = Load/Unload
AWC = Automatic Work Changer
CMM = Coordinate Measuring

Summary

- Communication of digital signals between CNC machine tools and other equipment used in the manufacturing systems is very important for running the operations smoothly.
- Serial and parallel communications are the two most common methods employed in computing equipment.
- Serial communication is used by CNC machine tools because of its ability to communicate over long distances and linking with dissimilar equipment.
- Direct numerical control (DNC) is a method employed to link CNC machine tools with a computer to exercise a lot of control on the way the machine tool is operated which is not possible with machine control unit alone.
- There are certain requirements to be considered before adopting DNC for a particular case.
- Enhanced DNC is a first step which can lead subsequently towards the computer integrated manufacturing
- There are a large number of benefits that can be achieved by the adoption of DNC, particularly when the installation has more than one CNC machine tool.

Questions

1. Which communication modes are available for CNC machine tools to talk to the outside world?
2. What are the differences between serial and parallel transmission?
3. Briefly explain the serial transmission method used in CNC machine tools.
4. Explain why serial transmission is generally preferred in CNC machine tools.
5. What are the functions served by DNC?
6. Briefly explain the procedure to be followed for remote program buffer usage.
7. Briefly explain the type of installations used in DNC on shop floor.
8. What are the requirements for adopting DNC?
9. What are the circumstances in which DNC becomes useful?
10. List the advantages of using DNC.
11. What do you understand by enhanced DNC?
12. Give a brief description of MAP.

21

MATERIAL-HANDLING SYSTEMS

Objectives

Generally, a material spends more time in a shop moving than being machined, which means that there is more time wasted in adding to the cost of the product. Thus, material-handling methods used are important in improving the profitability of a manufacturing organisation. The major automated material-handling systems that are generally used in advanced manufacturing are

- Automated Guided Vehicles (AGV)
- Robots
- Automated Storage and Retrieval Systems (AS/RS)

After completing the study of this chapter, the reader should be able to

- Understand the principles of material-handling system requirements
- Understand the principles on which AGVs operate
- Learn about the guidance principles used in AGV
- Identify different types of AGV systems that are used in different operations in manufacturing systems
- Design the AGV system for a given application
- Learn the various types of robots that can be used as material-handling units
- Understand the different parts of a robot and their functions
- Learn about the various types of sensors used in robots
- Select and design different types of grippers
- Understand different types of robot-programming methods used
- Make approximate cost justification for implementing robots
- Appreciate the different applications of robots
- Make preliminary estimations for robot cell designs
- Understand AS/RS as a storage system
- Learn various advantages of using AS/RS
- Design AS/RS for a given application

21.1 INTRODUCTION

The material-handling system is vital to the efficient operation of a manufacturing operation. Material handling is defined by the Material Handling Industry of America as 'the movement, storage, protection and control of materials throughout the manufacturing and distribution process including their consumption and disposal'. In a manufacturing plant, there are a number of items that need to be moved, such as raw materials, tools, consumables, work-in-process parts, finished goods, waste materials, etc. The function of a material-handling system is to make sure that the right material in the right quantity reaches the right place at the right time. Thus, it is a very important function in a manufacturing operation, but often overlooked. Though transport of material does not add value to the product, it is essential and as such it is important for the manufacturing engineer to properly account for its contribution to the overall cost of the product.

21.1.1 Principles of Material Handling

The principles of material handling as identified by the College-Industry Council on Material Handling Education (CICMHE) are as follows:

Planning Principle All material handling should be the result of a deliberate plan where the needs, performance objectives, and functional specification of the proposed methods are completely defined at the outset.

Standardisation Principle Material handling methods, equipment, controls, and software should be standardised within the limits of achieving overall performance objectives and without sacrificing needed flexibility, modularity, and throughput.

Work Principle Material handling work should be minimised without sacrificing productivity or the level of service required of the operation.

Ergonomic Principle Human capabilities and limitations must be recognised and respected in the design of material-handling tasks and equipment to ensure safe and effective operations.

Unit Load Principle Unit loads should be appropriately sized and configured in a way which achieves the material flow and inventory objectives at each stage in the supply chain.

Space Utilisation Principle Effective and efficient use must be made of all available space.

System Principle Material movement and storage activities should be fully integrated to form a coordinated, operational system that spans receiving, inspection, storage, production, assembly, packaging, unitising, order selection, shipping, transportation, and the handling of returns.

Automation Principle Material-handling operations should be mechanised and/or automated where feasible to improve operational efficiency, increase responsiveness, improve consistency and predictability, decrease operating costs, and eliminate repetitive or potentially unsafe manual labour.

Environmental Principle Environmental impact and energy consumption should be considered as criteria when designing or selecting alternative equipment and material-handling systems.

Life-Cycle Cost Principle A thorough economic analysis should account for the entire life cycle of all material handling equipment and resulting systems.'

21.1.2 Types of Material-Handling Devices

A large variety of material-handling equipment is in use in the manufacturing and service industries. The varieties of equipment that is used for transporting material within the factory or a warehouse is different, depending upon the type of service that needs to be provided.

Industrial Trucks These are a large variety of powered and non-powered equipment used for transporting material and equipment on the shop floor. This class of equipment includes fork-lift trucks, hand carts, and tractor-trailer rigs. Trucks are normally driven by a human operator and are useful for intermittent moves over varying paths but require adequate aisles.

Automated Guided Vehicles (AGVs) These are battery-powered, driverless vehicles for automatic transport of parts and tooling on the shop floor. These move on fixed paths laid underneath the factory floor, and transport material from the workstations to storage locations, load stations, etc. Though the path of travel is laid underneath the factory floor, it is made of segments which allow the AGV to have a flexible path. These are one of the first choices for automating the material movement. A class of AGVs called rail-guided vehicles have fixed rails on which they move. This is far more restricted in terms of the path the RGV can take and service only a few workstations.

Conveyors These can be used for moving materials over a fixed path to specified locations. It transports materials of relatively uniform size and weight with moderate to high frequency. Conveyors can be used to move material, as well as position fixture for workers along a production line. There are many types of conveyors available, characterised by the type of surface used, such as belts, chutes, wheels, or rollers. Of these types, only belts circulate, keeping the part in the same position relative to the belt. Chutes, wheels, and rollers stay in place (or rotate about their longitudinal axis), while parts move along the conveyor. Conveyors are generally powered, while non-powered conveyors when used will have the worker move the part over the conveyor. Another type of conveyor is the *trolley conveyor*, which is normally located overhead, allowing parts to be transported at equal space increments. Parts are suspended on hooks or placed on carriers connected to the conveyor.

Cranes and Hoists These are overhead lifting devices that are often used for intermittent movement of large and heavier equipment within a fixed space. Hoists lift material vertically while suspended from a hook. Cranes move horizontally over guide rails with the product being suspended from a hoist when the crane moves. A number of types of cranes are used such as bridge crane, gantry crane, monorail crane and stacker crane.

21.2 AUTOMATIC GUIDED VEHICLES (AGV)

The Automated Guided Vehicle (AGV) is a programmable mobile vehicle without the need of human intervention. The Material Handling Institute defines it as *'An AGV is a vehicle equipped with automatic guidance equipment, either electromagnetic or optical. Such a vehicle is capable of following prescribed guide paths and may be equipped for vehicle programming and stop selection, blocking and any other special functions required by the system.'*

A typical AGV with a pallet and workpiece mounted on it is shown in Fig. 21.1. These are basically driverless vehicles and work generally in fixed routes that are laid on the factory floor. AGVs are used for workpiece distribution and transferring them from stores to shop/assembly line. They are sometimes also called *robocarts*. The main components of an AGV based material handling system are given below.

- The vehicle, which is used to support and move the material from one point to the other without the help of a driver or operator. The main parts of an AGV are
  - Structure
  - Drive system
  - Steering mechanism
  - Power source, battery
  - Onboard computer for control

- The guide path, the actual path through which the vehicle moves
- Traffic management, that manages the maximum load movement through the system avoiding other vehicles and collisions
- Load transfer is the pickup and delivery method used for interfacing with other parts of the system such as conveyors, or CNC machine tools.

21.2.1 AGVS Types

A number of AGV types are available to cater to the variety of functions. They are

- AGVS towing vehicles
- AGVS unit load vehicles
- AGVS pallet trucks
- AGVS fork trucks
- Light load vehicles
- AGVS assembly line vehicles

The schematic representation of these vehicles is shown in Fig. 21.2. Over the years, the developments in AGVS have made them very versatile in view of the very large applications for which these are used.

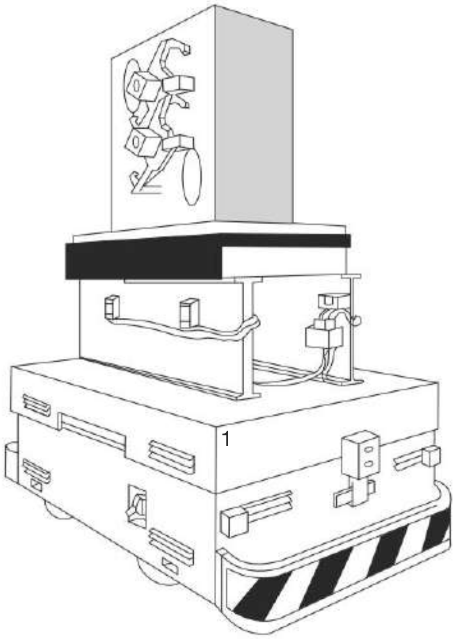
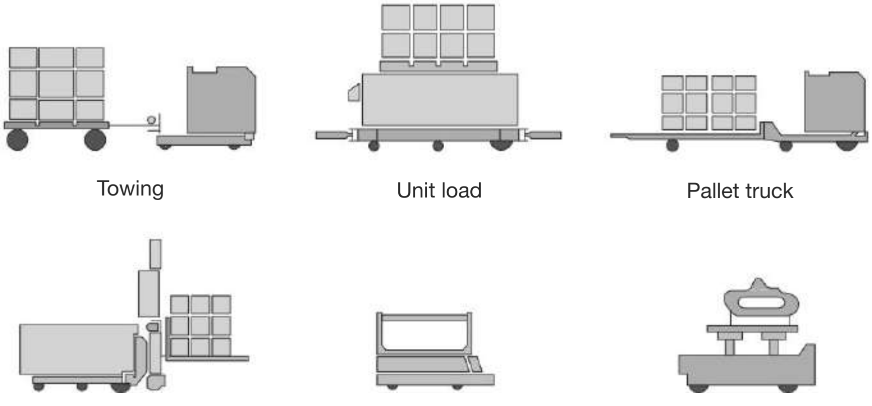


Fig. 21.1 *The Automated Guided Vehicle System (AGVS) with a workpiece pallet*


Towing

Unit load

Pallet truck

Fork truck

Light load

Assembly line

Fig. 21.2 *Various types of Automated Guided Vehicle Systems (AGVS)*

AGVS Towing Vehicles These are the first types introduced and are still very popular. These are used for very large load applications. The towing vehicle can have a variety of trailers. These are generally used for bulk transport applications.

AGVS Unit Load Vehicles These have a deck that permits unit-load transport operation and are suitable for automatic transfer of load. The decks can be either lift and lower type, which is most common, powered or non-powered roller, chain, or custom design. The unit load carrier is used for moving high volumes over moderate distances, and can easily integrate other subsystems such as conveyors and storage systems. Typical speeds are 50 metres per minute. They are used in warehousing and distribution systems. The typical path of a unit load vehicle is shown in Fig. 21.3. These systems have bi-directional mobility, and operate independently. As a result, they allow for good system versatility for product movement.

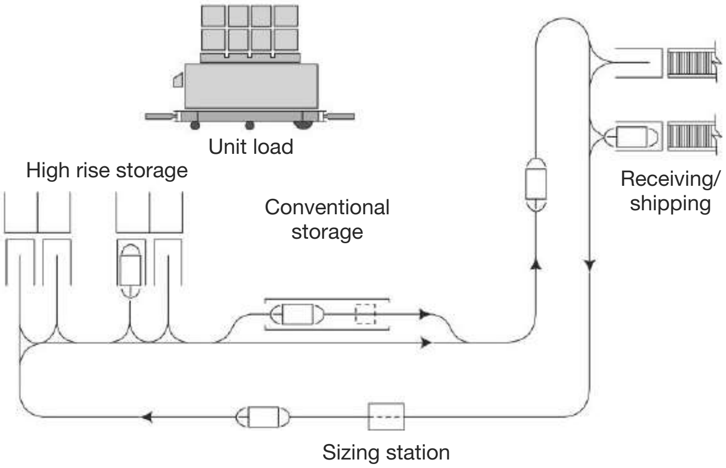


Fig. 21.3 *Typical application path of a unit-load type AGVS*

AGVS Pallet Trucks Pallet trucks are used to transport palletised loads from floor level and eliminate the need for fixed load stands. These are used in distribution functions.

AGVS Fork Trucks Fork trucks have the ability to service palletised loads, both at the floor level as well as on stands. They may also be able to stack the loads when required. These are generally used where the heights of load transfer vary. The vehicle has the capability of positioning to any height so that conveyors or all load stands of varying height in a given system can be serviced. These are some of the most expensive AGVS and can only be justified where total automation is required.

Light Load Vehicles Light load AGVs are vehicles with small capacities of the order of 200 kg and therefore are used to transport small parts through a light manufacturing environment.

AGVS Assembly Line Vehicles These are the adoption of light load vehicles for serial assembly processes. For light assembly applications, these vehicles carry subassemblies such as motors or transmissions, to which parts are added in a serial assembly operation. Prior to the assembly operation, the vehicle reaches the parts staging area where the necessary parts are placed on a tray onboard on the vehicle. The vehicle then moves to the assembly area where the assembler completes the assembly operation taking the parts onboard. When the assembly is completed, the vehicle is released and proceeds to the next parts-staging area. The same process may be repeated a number of times, before the completion of the assembly. The typical layout is shown in Fig. 21.4.

AGVS assembly systems allow flexibility in the assembling operation by providing for parallel operation. It is possible to track individual parts and measure work rates.

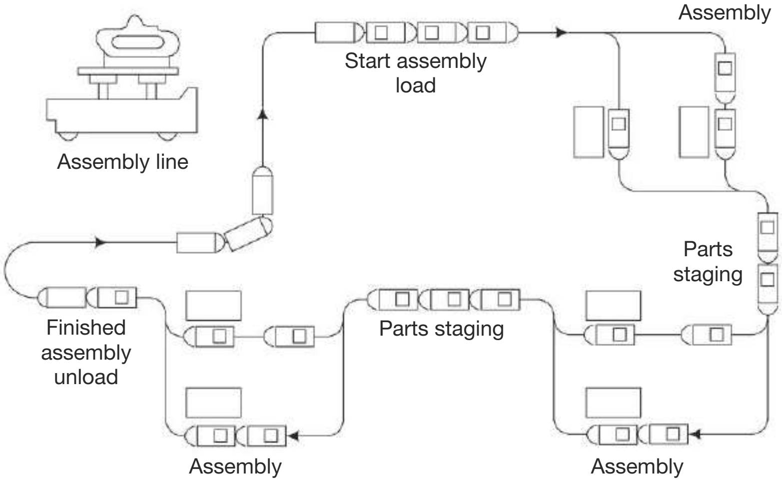


Fig. 21.4 *Typical application path of an assembly-line AGVS*

21.2.2 Guidance

An AGV is operated with onboard batteries and it moves generally in a fixed path. One of the important elements in an AGV is the guidance control. The various guiding principles used in AGV control are given in Table 21.1. The actual use of a particular guidance method is mainly dependent upon application, environment and need. The wire guidance is the most commonly used method in the manufacturing shops.

Table 21.1 *Guidance principles used for guiding AGVs*

Guidance type	*Description*
Wire guided	Vehicle's antenna senses and follows an energised wire embedded in the floor
Infrared	Infrared light is transmitted and reflected from reflectors in the roof of a facility; radarlike detectors relay signals to a computer and calculations and measurements are taken to determine position and direction of travel
Laser	Wall-mounted laser scans barcoded reflectors; through known distances and measurement of the distance, the vehicle's front wheel has traversed, the AGV can be accurately manoeuvred and located

The principle of wire guidance is given in Fig. 21.5. The control wire is embedded in the factory floor along which the AGV is to traverse. For this purpose, a rectangular slot is cut into the concrete floor and the wire is placed in position with the rest of the slot being filled with epoxy as shown in Fig. 21.5. The wire is actually in segments depending upon the actual path to be taken as shown in Fig. 21.6. The transfer of AGV from one loop to the other is done with the help of the circular transfer elements present in the path.

Each of the travel is identified by a particular frequency, and the wire that forms the part is energised to that frequency. The onboard controller of the AGV is adjusted for this frequency. The sensor coils present in the AGV sense the presence of the magnetic field and accordingly steer the AGV along the path. The two coils placed at equi-distance on either side of the coil help in maintaining the movement of the vehicle along the wire. If the AGV has to follow a different path then its frequency needs to be adjusted for that frequency.

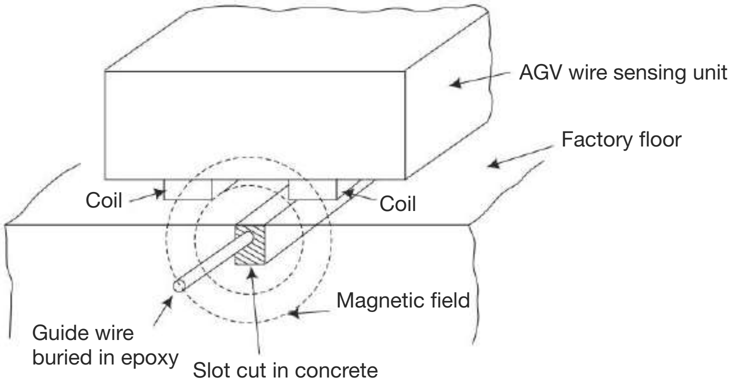


Fig. 21.5 *The principle of wire guidance used in AGV*

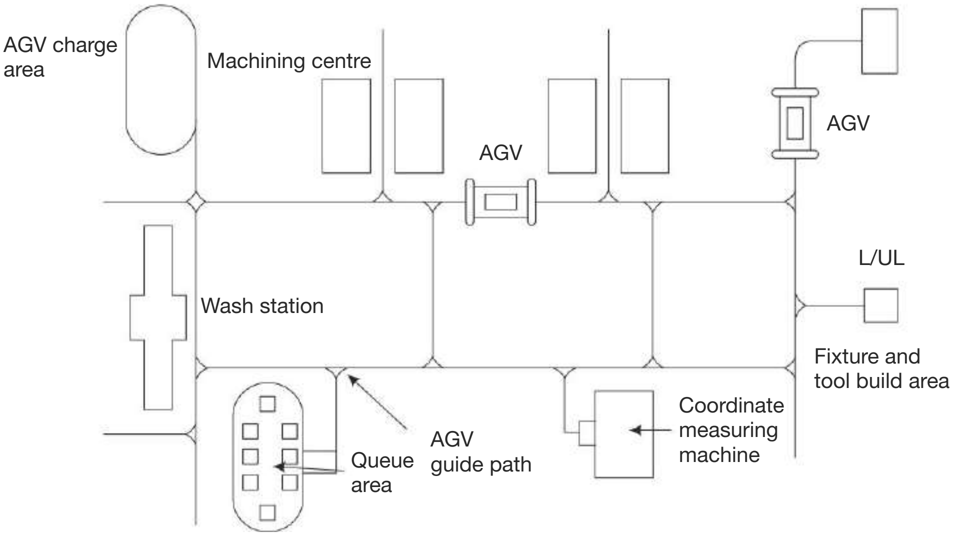


Fig. 21.6 *Typical path of a wire-guided AGV*

There are situations when the floor-wired system might not be feasible for guiding the AGV movement. Some of the situations are when

- the floor is uneven and not suitable for embedding wire,
- there are frequent changes in the path, and
- there are a number of metal encumbrances in the floor.

In such cases, free-ranging AGVS with no fixed path using laser ranging are also available but are less used in manufacturing plants. The AGVS locates its position by reading the barcode targets and by sensing the steer-wheel angle and rotation as shown in Fig. 21.7. The onboard computer communicates the information processed through a radio link to a stationary control computer.

Another form of the AGV is a rail-guided vehicle or RGV, which travels on fixed rails laid out as shown in Fig. 21.8. This type of vehicle is used for short travel distances and heavy workpieces. These are not

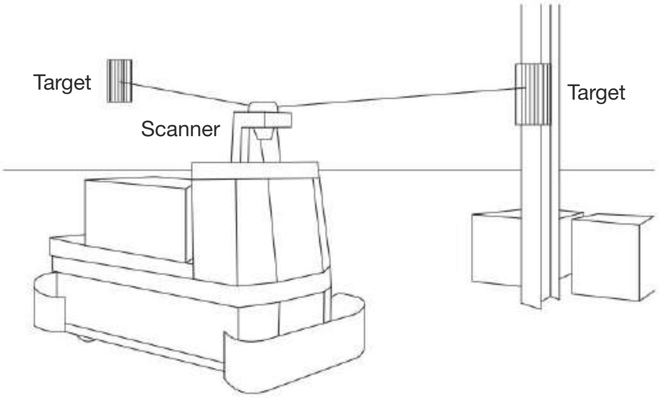


Fig. 21.7 *Wireless guidance of AGVS using laser guidance*

as flexible as the wire-guided ones and, therefore, are used exclusively in flexible manufacturing systems involving smaller number of machine tools; whereas the wire-guided AGVs are used in almost all types of applications including assembly and storage.

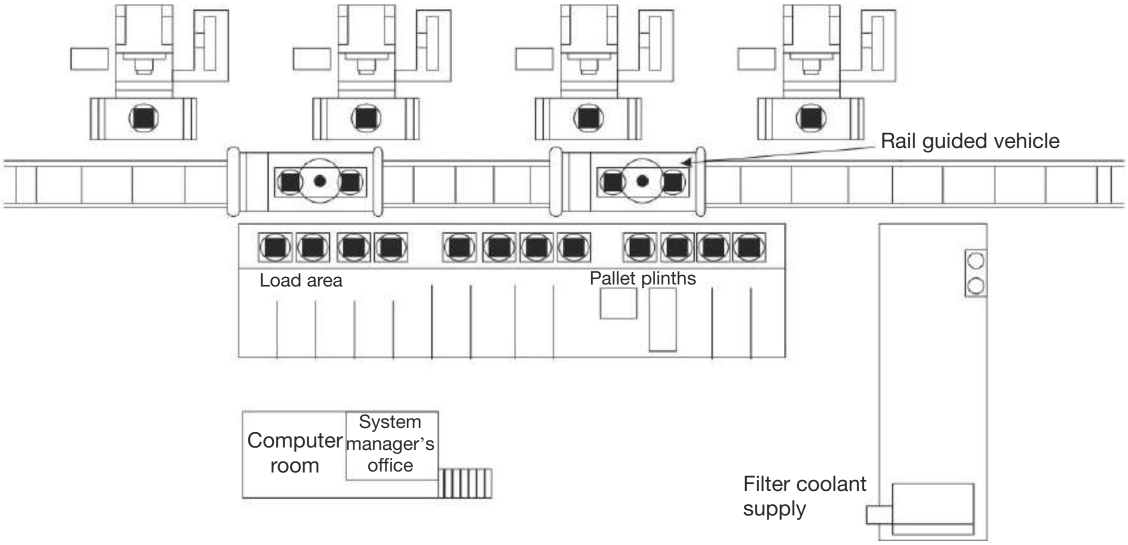


Fig. 21.8 *Typical path of a rail-guided AGV*

Since the AGVs are used for many applications, the type of work-handling system to be provided depends upon the application. Figure 21.1 shows a typical example where the pallet is directly mounted on the AGV and is the most common form used in machining systems. Other designs involve the provision of lift platform, telescopic loading fork, etc.

21.2.3 System Management

As shown in Fig. 21.6, there can be more than one AGV servicing the facilities. Accordingly, the traffic control system will have to take care of all the AGV movements. Since the vehicle is run by batteries, there will be a battery-charging station provided which is used for recharging the AGV batteries when the AGV is idle.

The AGV traffic control system has to take care of

- Selection of AGV and administration of idle vehicles
- Control of the despatching sequence
- Tracking the AGVs
- Controlling the actual traffic

21.2.4 AGV System Design

AGVS is a complex system and a number of parameters need to be considered. They are

- Track layout
- Number of AGVs required
- Operational and transportation control

Track Layout The track layout defines the possible vehicle movement path. Links and nodes that represent the action points such as pick-up and drop-off points, maintenance areas or intersections represent the path. The guide path can be divided into four types:

- Uni-directional single lane
- Bi-directional single lane
- Multiple lanes
- Mixed

The bi-directional single lane is the most cost-effective and widely used layout.

Number of AGVs Required It is important to estimate the optimum number of AGVs required for a system. Too many AGVs will congest the traffic, while too few may mean larger idle time for the workstations in the system. The number of AGVs required is the sum of the total loaded and empty travel time and waiting time of the AGVs in a busy time period, divided by the time an AGV is available during that period.

Total time per delivery, T_{dv} is given by

$$T_{dv} = \frac{D_b}{V} + T_1 + T_{ul} + \frac{D_e}{V}$$

where

D_b = total average loaded travel time
V = vehicle travel speed
T_1 = loading time
T_{ul} = unloading time
D_e = total average empty travel time

The number of deliveries per vehicle per hour, N_d is given by

$$N_d = \frac{60\ T_f}{T_{dv}}$$

where T_f is a traffic factor to account for the waiting period of the vehicle at the line and intersections due to blocking. Theoretically, without any traffic congestion, its value should be 1. For example, if there is only one AGV in the system, there may not be any blocking. For a general situation, it can be taken as any value between 0.85 and 1.00.

Operational and Transportation Control Typical interfaces of various models in an AGVS are shown in Fig. 21.9. The operation and transportation consists of the following modules:

- Vehicle despatching
- Vehicle routing
- Traffic control

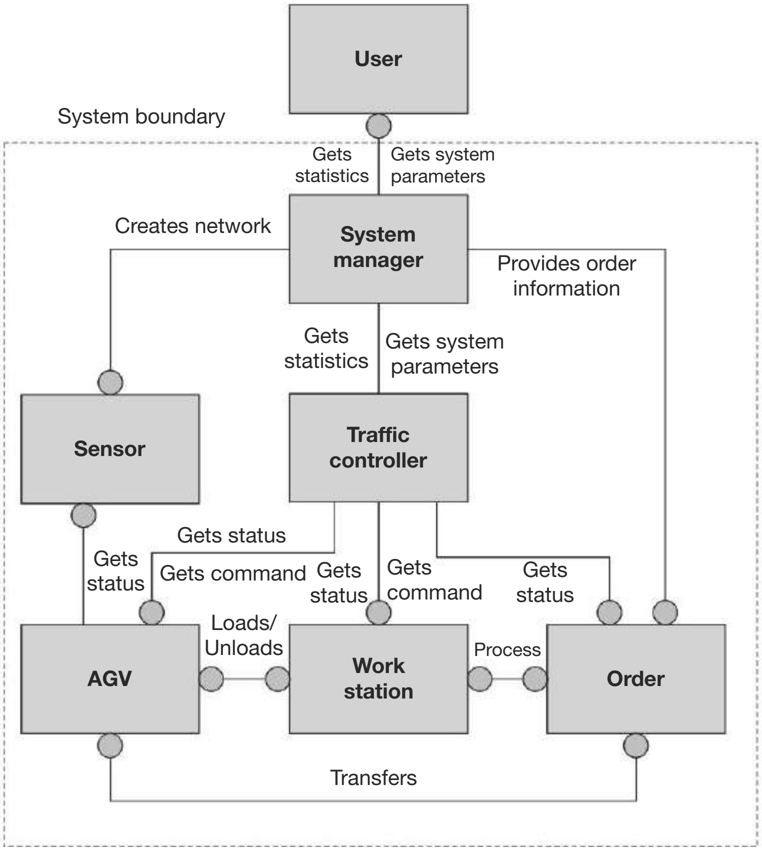


Fig. 21.9 *Typical information model of AGVS*

(a) Vehicle despatching Once the on-hand task creates a demand for an AGV, a choice needs to be made regarding the vehicle to be despatched among the pool of vehicles available. Alternatively, when several workstations need servicing, a choice is to be made as to which workstation is to be serviced. Selection criteria can be applied for assigning the vehicle or workstation. Some of the selection criteria are

- A random vehicle
- Longest idle vehicle

- Nearest vehicle
- Farthest vehicle
- Least utilized vehicle
- Random workstation
- Nearest workstation
- Farthest workstation
- Maximum queue size
- Minimum remaining queue size
- First come first served
- Unit load arrival time, due date, or priority

Generally, availability of multiple AGVs is rare, since the optimum number of AGVs would have been selected at the design stage, so most important assignment rules to be considered are those related to the workstation.

(b) Vehicle routing In order to dispatch an AGV to any workstation, it is necessary to find the shortest feasible path from the existing position. While selecting the shortest path, it is necessary to consider only those paths that are free and not occupied by vehicles. It may also be necessary to consider the future positions of the vehicles in the route in addition to their current occupied positions.

(c) Traffic control In identifying the traffic control systems for AGVS movement, the approaches that can be used are

- Forward-sensing control
- Zone-sensing control
- Combinatorial control

In the forward-sensing control, the AGV is equipped with obstruction detecting sensors that can identify another AGV in front of it, and slow down or stop. This helps in improving the AGV utilisation due to closer allowable distance between vehicles. However, this approach may not be able to detect the obstacles at intersections and around corners. This is generally useful for long and straight paths. The zone-sensing control is more global since the central computer keeps track of the entire guide path, which is divided into zones. Once an AGV enters a zone, that zone becomes blocked for other AGVs. Though this is a more safe approach, it introduces some inefficiency. In combinatorial control, both of the above approaches are selectively used to obtain the benefits of both strategies.

21.2.5 Advantages

The main advantages derived from the use of AGVs in manufacturing environment are given here.

1. Dispatching, tracking and monitoring under real-time computer control. This helps in planned delivery, online interface to production and inventory control systems, and management information on vehicle and workstation production.
2. Better resource utilisation. Most AGVs can be justified economically in three years or less.
3. Increased control over material flow and movement.
4. Reduced product damage and less material-movement noise.
5. Routing consistency with flexibility.
6. Operational reliability in hazardous and special environment.

7. Ability to interface with various peripheral systems such as machine tools, robots and conveyor systems.
8. Increased throughput because of dependable on-time delivery.
9. High location and positional accuracy.
10. Improved cost savings through reduction in floor space, WIP and direct labour.

21.3 ROBOTS

A robot is an automatically controlled material-handling unit that is widely used in the manufacturing industry. It is generally used for high-volume production and better quality. Implementation of robot technology with integration of an automatic system can contribute to increase in productivity of a company and enhances the profitability of the company.

The word 'robot' first appeared in 1921 in the Czech playwright Karel Capek's play 'Rossum's Universal Robots'. The word is linked to Czech words *robota* (meaning work) and *robotnik* (meaning slave). *Computer Aided Manufactures International* of USA describes the meaning of robot as *a device that performs functions ordinarily ascribed to human beings, or operates with what appears to be almost human intelligence*. Another definition from the Robot Institute of America is ...*a programmable multifunction manipulator designed to move and manipulate material, parts, tools or specialised devices through variable programmed motions for the performance of a variety of specified tasks.*

ISO defines a robot as *a robot is an automatically controlled, reprogrammable, multipurpose, manipulative machine with several reprogrammable axes, which are either fixed in place or mobile for use in industrial automation application.*

The Webster dictionary defines a robot as *an automatic apparatus or device that performs functions ordinarily ascribed to humans or operates with what appears to be almost human intelligence.*

There are a number of successful examples of robot applications such as

- Robots perform more than 98% of the spot welding on Ford's Taurus and Sable cars in USA
- A robot drills 550 holes in the vertical tail fins of a F-16 fighter in 3 hours at General Dynamics compared to 24 man-hours when the job was done manually
- Robots insert disk drives into personal computers and snap keys onto electronic typewriter keyboards

21.3.1 Robot Applications

True to the above definitions of robot as an automatic machine, industrial robots are observed to perform the following tasks (shown in the ascending order of technological complexity) in manufacturing.

(a) **Parts Handling** This may involve tasks like
  - Recognising, sorting/separating the parts
  - Picking and placing the parts at desired locations
  - Palletising and depalletising
  - Loading and unloading the parts on required machines

(b) **Parts Processing** This may involve operations like
  - Routing
  - Drilling
  - Riveting
  - Arc welding

- Grinding
- Flame cutting
- Deburring
- Spray painting
- Coating
- Sandblasting
- Dip coating
- Gluing
- Polishing
- Heat treatment

(c) **Product Building** This may involve assembly of typical products like

- Electrical motors
- Car bodies
- Solenoids
- Circuit boards and operations like
    - Bolting
    - Riveting
    - Spot welding
    - Seam welding
    - Inserting
    - Nailing
    - Fitting
    - Adhesive bonding
    - Inspection

The automation of the above tasks greatly facilitates computer controlled manufacturing systems. Further, robots have often been used in undesirable and hazardous environment like that of excessive heat, dust, noise, fumes, etc., and for dirty, dangerous dull and difficult tasks. Accordingly, the industrial robot has become an essential component of all flexible manufacturing systems, subsystems, cells and modules. The robot application in US is given in Table 1.

Table 21.2 *US robot sales by application*

	1985	*1990*	*1995*
Machine tending	16%	15%	15%
Material transfer (machine tending)	16	15	15
Spot welding	26	15	10
Arc welding	10	10	9
Spray painting/coating	10	10	7
Processing (drilling, grinding, etc.)	5	7	7
Electronics assembly	6	12	14
Other assembly	5	8	12
Inspection	5	7	10
Other	1	1	1
Total	100	100	100

Robots are being applied in industries because of the following:

- ***Hazardous or Uncomfortable Working Conditions*** In situations where there are potential dangerous or health hazards (like heat, radiation, toxicity, etc.) robots may be used. Some of the examples are hot forging, die casting, spray-painting, etc.
- ***Difficult Handling*** If the workpiece or tool involved in the operation is awkward in shape or heavy, it is possible that a robot can do this job much better.
- ***Multishift Operation*** For increasing production and reducing the costs, multishift operations may be desirable in which robots can work continuously.
- ***Repetitive Tasks*** If the work cycle consists of a sequence of elements, which do not vary from cycle to cycle, it is possible that a robot can be programmed to do the job.
- ***Higher Accuracy*** In situations where the accuracy of operation required is very high.

21.3.2 Robot Types

Robots can be classified by the type of motions provided. But a few definitions need to be understood in relation to robots.

- **Tool Centre Point (TCP)** It is the point of interest, which is used to carry out the robot function.
- **Work envelope** Extreme positions of Tool Centre Point (TCP) determine limits of robot motion. Space (Volume) of all possible TCP.
- **Degrees of freedom** Number of independent ways the robot can move. Similar to the axes in CNC machine tool.
- **End effector** It is that part of the robot that is used to carry out the necessary function. It is generally a gripper or end-of-arm tool that is placed in the gripper.

Cartesian Coordinates Positioning may be done by linear motion along three principal axes—left and right, in and out, and up and down. These axes are known, respectively, as the Cartesian axes X, Y and Z. Figure 21.10 shows a typical manipulator arm for a Cartesian coordinates robot. The work area or work envelope serviced by the Cartesian-coordinates robot's arm is a big box-shaped area. Programming motion for a Cartesian-coordinates robot consists of specifying to the controller the X, Y and Z values of a desired point to be reached. The robot then moves along each axis to the desired point. This is one of the simplest types of robots.

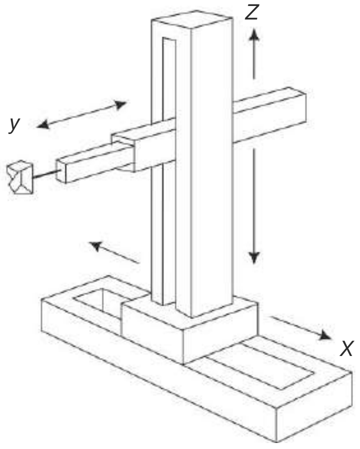


Fig. 21.10 *Typical motions of a Cartesian or rectilinear robot*

Spherical or Polar Coordinates This type of robot uses mostly rotational axes. The axes for the spherical coordinates are θ, the rotational axis; R, the reach axis; and β, the bend-up-and-down axis. The work area serviced by a polar-coordinates robot is the space between two concentric hemispheres. The reach of the arm defines the inner hemisphere when it is fully retracted along the R axis. The reach of the arm defines the outer hemisphere when it is fully straightened along the R axis. Figure 21.11 shows the typical robot.

Cylindrical Coordinates In this type of robot, there is a rotary motion at the base followed by the two linear motions. The axes for the cylindrical coordinates are θ, the base rotational axis; R (reach) the in-and-out

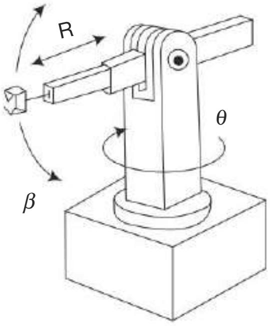


Fig. 21.11 *Typical motions of a spherical robot*

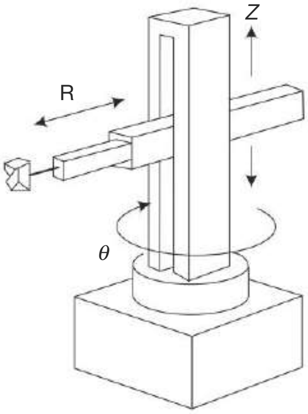


Fig. 21.12 *Typical motions of a cylindrical robot*

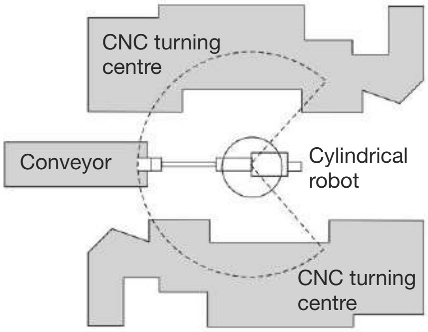


Fig. 21.13 *A cylindrical robot serving two CNC turning centres*

axis; and *Z*, the up-and-down axis. The work area is the space between two concentric cylinders of the same height. The inner cylinder represents the reach of the arm with the arm fully retracted, and the outer cylinder represents the reach of the arm with the arm fully extended. Figure 21.12 shows the typical cylindrical robot. An example of a cylindrical robot linking the two CNC turning centres and a conveyor for workpiece loading and unloading is shown in Fig. 21.13.

Jointed Coordinates If the arm can rotate about all three axes, then it is called a revolute coordinates, articulate or jointed-arm robot. Figure 21.14 shows a typical manipulator arm for the articulated robot.

SCARA Robot The Selective Compliance Assembly Robot Arm (SCARA) is a type of robot that is commonly used for assembly application (Fig. 21.15). The arm picks up a piece-part vertically from a horizontal table, and moves it in a horizontal plane to a point just above another place on the table. Then it lowers the part to the table at the proper point to accomplish the assembly, perhaps including a rotation operation to insert the part into the assembly.

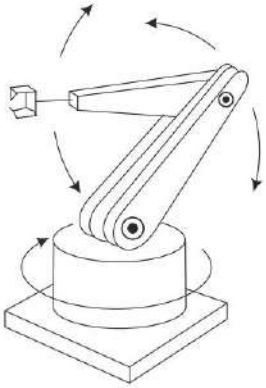


Fig. 21.14 *Typical motions of an articulated robot*

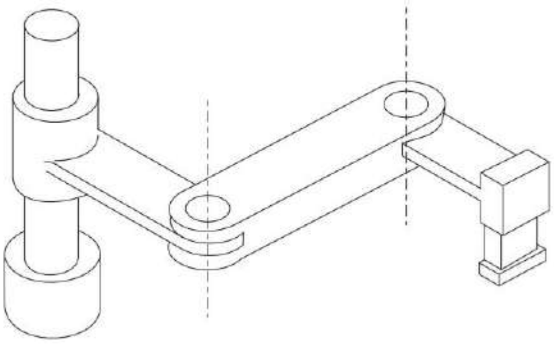


Fig. 21.15 *A SCARA robot*

Workspace of a Robot The workspace of a robot, which is also called *work envelope* or *robot reach*, is the maximum volume that a robot can reach mechanically. The mounting point of the robot where the end effector is attached is considered for the purpose of calculating the workspace. The additional space that can be reached by the end effector is not considered in the robot reach. This is utilised for the selection of robot and calculation of the robot placement in a work cell in relation to the workstations. Some examples are given later in the chapter. The work envelope of a typical articulated coordinate is shown in Fig. 21.16.

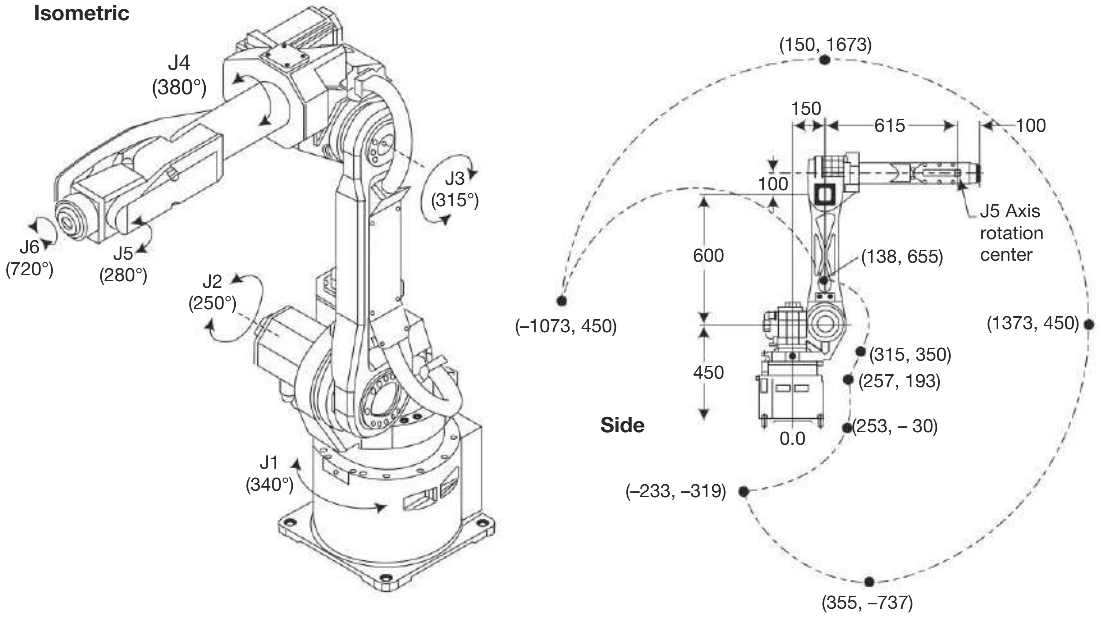


Fig. 21.16 *The work envelope of an articulated robot. (Courtesy, Fanuc Robotics North American Inc.)*

21.3.3 Basic Components of a Robot

The robot systems used in manufacturing systems generally have the four basic components—manipulator, controller, power source and the end effector. The base of the manipulator is usually fixed to the floor of the work area. Sometimes, the base may be moveable by attaching it to either a rail or track, allowing the manipulator to be moved from one location to another. Sometimes, it is also possible to have a gantry system from which the robot will be hanging, thus conserving the floor space.

The *manipulator*, which does the physical work of the robotic system consists of a number of links which can be either a straight or a moveable arm of the robot as explained earlier. The movement of the manipulator is controlled by the actuators. The actuator, allows the various axes to move within the work cell. The drive systems can use electric, hydraulic or pneumatic power.

Hydraulic power is the most powerful and is also the most expensive. It consists of a pump of sufficient capacity, and a reservoir for hydraulic fluid. It is safe from fire hazards. It is generally preferred for spray-painting applications. The pneumatic power is the least expensive, but also provides low power. It has limited capability and usually operates with mechanical, fixed endpoints for each axis. Thus, it limits the programmability to only the two extreme positions.

Electric power provides the most versatile applications. As a result, it is more popular for precision jobs. It can be closely controlled and so is able to follow complicated paths of motion. The type of drives used are stepper motors and servomotors. The details of these drives are given in Chapter 10.

The *controller* in the robotic system (Fig. 21.17) is the heart of the total operation. The controller stores preprogrammed information for later recall, controls peripheral devices, and communicates with computers. The controller is used to control the robot manipulator's movements as well as to control peripheral components within the work cell. The controller stores all program data for the robotic system. It can store several different programs, and any of these programs can be edited.

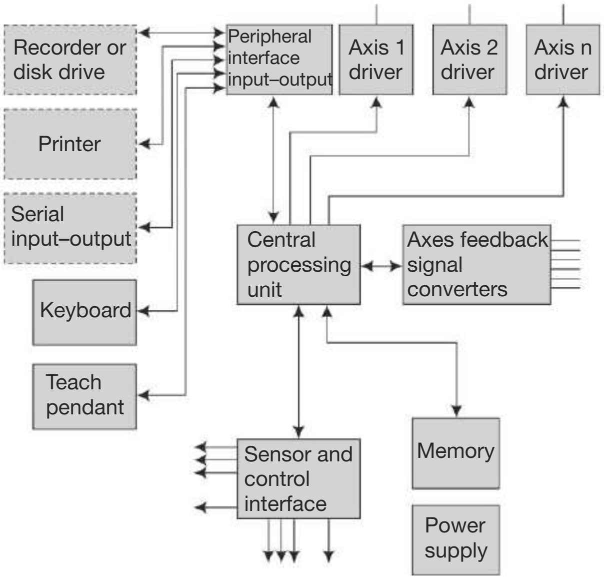


Fig. 21.17 *Typical functions of a robot controller*

The controller is also required to communicate with peripheral equipment within the work cell. Such peripherals that are commonly connected to the controller are teach pendant, hard drive, keyboard, memory and floppy disk. The two-way communication between the robot manipulator and the controller maintains a constant update of the location and the operation of the system. The controller also controls any tooling placed on the end of the robot's arm.

21.3.4 Types of Motion Control

Though the type of drive used signifies to a great extent the type of motions that are possible, it is necessary to understand the various types of motion control that are possible to get an idea of the actual application for which a robot can be used. The four types of motion controls that are used in robots are

- Axis limit
- Point-to-point
- Contouring
- Line tracking

Axis Limit The simplest type of motion control is in the case used with pneumatic drives. In this case, the only possible positions are the two extreme positions for each of the axis. These positions (minimum and maximum) are generally controlled by limit switches or adjustable stops. This type of control is low cost and the least sophisticated. Generally, no speed control is possible, and the speed depends on the type of drive used.

Point-to-Point Most of the controls in the market are of this type. This control is used when the system is supposed to reach only specified points in the workspace. This is better than axis-limit systems. However, the path and speed between points are not controllable. As such, the actual path between points is not predictable especially for non-cylindrical coordinate robots. It is generally good for component insertion, hole drilling, spot welding and some crude assembly operations. It is generally used for such operations as injection moulding machine handling, die-casting machine handling, and part handling at machine tools.

Contouring The continuous path control is more advanced compared to point-to-point and is similar to NC systems. The entire path of the gripper is continuously controlled and generally utilises a feedback loop. The motion system has controls for position as well as velocity. These systems are permitted to traverse any path required. It is generally good for spray painting, finishing, gluing, and arc-welding applications.

Line Tracking This is the most complex motion system for a robot. It is a robot that is performing an operation alongside a continuously moving conveyor. It is superior to contouring control, and is another capability added to contouring. It is more difficult since the base of the robot is fixed to the ground while the part is moving on a contour. Robots moving on a linear guide (rails) have a certain advantage in such cases compared to the fixed-base robots. The application for which this is most suited is spray painting, but it can also be used when multiple operations are to be performed on a part.

The power supply is the unit that supplies power to the controller and the manipulator. Two types of power are delivered to the robotic system. One type of power is the electric power for operation of the controller. The other type of power is used for driving the various axes of the manipulator. This power can be developed either from hydraulic, pneumatic or an electric power source.

The sensors present in the robot at various locations communicate to the robot controller about the status of the manipulator. For proper control of the manipulator, it is necessary to know the position, velocity and acceleration of each of the joints. In addition to the sensors to track the motion, other sensors are used to provide further feedback about the workpiece handling.

The purpose of the robot manipulator is to perform work. An end effector attached to the robot's arm must accomplish the work to be done by the robot. The end effector can be a gripper for work handling or end-of-arm tooling if a particular job is to be done such as welding or riveting.

The manipulator moves the end effector to the programmed locations. These moves of the end effector are controlled by a robot's program stored in the controller memory. The type of end effector depends upon the type of work holding to be done. These can be operated by mechanical means such as using a pneumatic or hydraulic cylinder, or by using vacuum to lift and transfer the part, or by using an electromagnet to lift and move the part. The robot's end effector may have sensors such as proximity switches, light sensors, pressure switches, magnetic-field sensors, vibration detectors or speed-of-motion sensors depending upon the application.

21.3.5 Sensors

The manipulator portion of the robot consists of a number of elements that are connected at a number of joints and the final destination reached by the tool-centre point depends upon the accuracy of each of these joints. Therefore, it is necessary to provide necessary feedback about the current position and the dynamics of the component. The type of sensors used for this purpose are similar to the ones that are used in CNC machine tools as discussed in Chapter 10.

In addition to this type of sensors, the end of the arm tooling of the robot needs to have some type of sensors present to understand its environment, so that it will be able to perform its duties properly. The feedback from these sensors ensure that the robot controller operates the various movements efficiently. Also, the presence of sensors provides greater flexibility to the operation of a robot. The type of sensors used in robots are

- Proximity sensors
- Range sensors
- Other sensors for specific applications

Proximity Sensors These indicate the presence of an object within a specified distance in space without any physical contact with the object. These operate based on either inductive or capacitance principle. The sensor generates a magnetic field as shown in Fig. 21.18. The object, upon entering the magnetic field, generates a loss in the eddy current, which is utilised to sense the distance from the sensor surface.

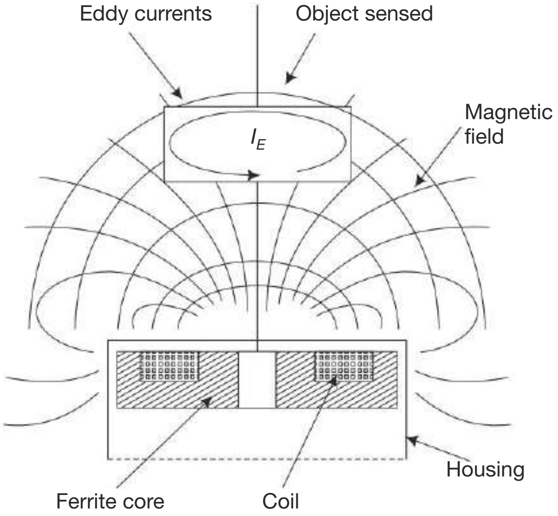


Fig. 21.18 *Typical electromagnetic field generated by an inductive sensor*

The choice of proximity sensors requires a clear understanding of physical and electrical properties of the sensors as well as the objects. A number of types of sensors are available in this category such as those that could be used for all materials, only for ferrous materials, or based on capacitance principle.

Range Sensors These measure distance from a reference point to other points of reference using sonar, infrared and television cameras.

Other Sensors for Specific Applications In addition to the above, a large number of other sensors are also used in robotic applications as part of the end effector to sense and identify the presence and type of holding of the component in the gripper. The type of measurements that need to be done are the following:

- **Presence of a part in the specified location** by means of optical sensors. In this case the object will interrupt a light beam (for example from a light emitting diode) which otherwise would have fallen on a light-sensing system. The breakage in the output from the light sensor indicates the presence of an object at that location. For example, this type of sensor can be used for closing the gripper to pick up an object, or if installed in the gate in the enclosure of the robot, will stop the operation of the robot identifying the intrusion of an operator into the robot-working space.

- **Pressure/force sensing** The elementary form of sensors used with grippers are some form of force or pressure experienced by the gripper fingers while closing on a part. This indicates the amount of force being applied on the part, which should depend upon the type of component structure and the likely motion of the gripper arm as it is moved in space (the centrifugal forces).
- **Tactile sensing** These are used to have a feel of the object similar to the feel experienced by human skin. It consists of a number of pressure sensors arranged on the surface of the gripper that is likely to come into contact with the object.

21.3.6 Grippers

A gripper is like the arm of an operator that establishes the connection between the workpiece and robot. In addition to this, grippers have to provide a number of advanced functions depending upon the application. For example, they must centre and orient the workpiece to facilitate the operation being carried out. The end-of-arm tooling used in a robot work cell should have the following characteristics [Rehg 2003]:

1. The tooling must be capable of gripping, lifting, and releasing the part or family of parts required by the manufacturing process.
2. The tooling must sense the presence of a part in the gripper, using sensors located either on the tooling or at a fixed position in the work cell.
3. The tooling weight must be kept to a minimum because it is added to the part weight to determine the maximum payload.
4. Containment of the part in the gripper must be ensured under conditions of maximum acceleration at the tool plate and loss of gripper power.
5. The simplest gripper that meets the first four criteria should be the one implemented.

Grippers generally consist of a number of fingers, which are kinematically linked and provided with motion to perform the gripping and opening actions. The type of drives that can be provided are electric, pneumatic or hydraulic. The type of motions that can be provided by these power sources are shown in Table 21.3.

Table 21.3 *Type of drive systems used in a gripper*

Gripper drive	*Drive movement*
Electrical drive	
Stepper motor	Rotational
DC servo motor	Rotational
Pneumatic drive	
Pneumatic cylinder	Linear
Compressed air motor (high speed)	Rotational
Swivel cylinder (low speed)	Rotational
Hydraulic drive	
Hydraulic cylinder	Linear
Hydraulic motor (unlimited rotation)	Rotational
Swivel cylinder (limited rotation)	Rotational

The typical arrangement of a two-finger gripper operated by a pneumatic cylinder is shown in Fig. 21.19. When the piston moves forward, the fingers move, opening the gripper for releasing the part. The reverse action provides for the gripping of the part. The fingers can be changed for different types of parts. It is possible to calculate the force acting on the part depending upon the dimensions of the fingers and the pneumatic power.

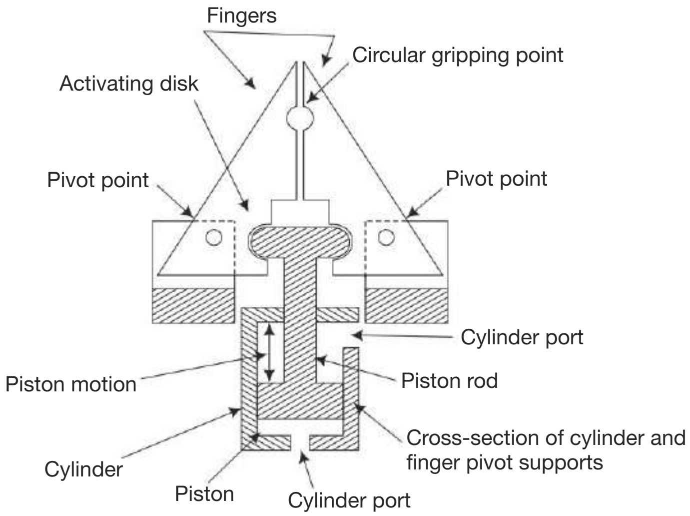


Fig. 21.19 *Typical two-finger gripper with a pneumatic cylinder*

Grippers need to be designed specifically for a given application utilising the concept that will be optimum for the purpose. However, it is possible to categorise the parts into some common shapes, such as that general-purpose grippers can be made available. For example, a few of the types of a general gripper available is shown in Fig. 21.20. This gripper is used for cylindrical workpieces.

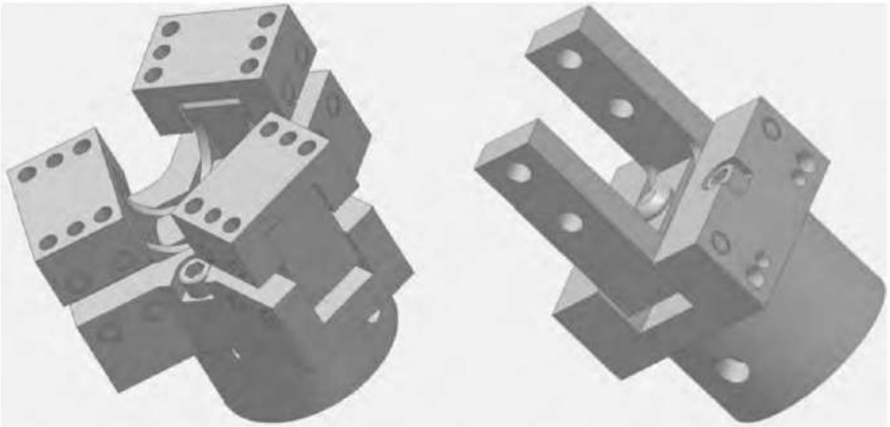


Fig. 21.20 *General-purpose standard grippers*

The clamping mechanism used in a gripper can be broadly classified as

- Mechanical clamping
- Magnetic clamping
- Vacuum clamping

(a) Mechanical Clamping This is the most common form of grippers which are operated by means of a pneumatic or hydraulic power source, with the pressure being applied directly on the component surface. The different styles used in industrial applications are parallel jaws, finger grippers, and expansion/contraction grippers. Parallel jaws have one or two moving jaws with either flat or V surfaces. An example is shown in Fig. 21.20. Finger grippers either encase the component or hold it at the very tip of the jaws. Expansion/contraction grippers have a flexible diaphragm or other device, which expands or contracts upon activation. This applies a friction force on the component and is generally used for delicate components.

(b) Magnetic Clamping It uses electromagnetism for holding and is used only for magnetic materials. It is independent of the component geometry to a certain degree.

(c) Vacuum Clamping By using vacuum, the parts adhere to the gripper because of the negative pressure. The common form is the use of suction cups arranged in a pattern to suit the component.

Double Grippers In applications of machine-tool loading and unloading, robots need to store the part and also orient it. In such cases, to reduce the machine downtime, it is advantageous to use a dual gripper instead of the normal single gripper as shown in Fig. 21.21. If a single gripper is used then the robot will have to remove the part after machining from the machine tool, and transfer it to the output tray, pick up the raw part from the input tray and then load the machine tool. For all this period, the machine tool will be idle. With a dual gripper, the robot can approach the machine tool with the raw part as shown in Fig. 21.21, and after picking the machined part from the machine tool, can simply index to put the raw part into the machine tool, thereby saving considerable idle time for the machine tool. The disadvantage with dual grippers is that they require more space for manoeuvring and have to be of higher capacity because they carry two workpieces at a time.

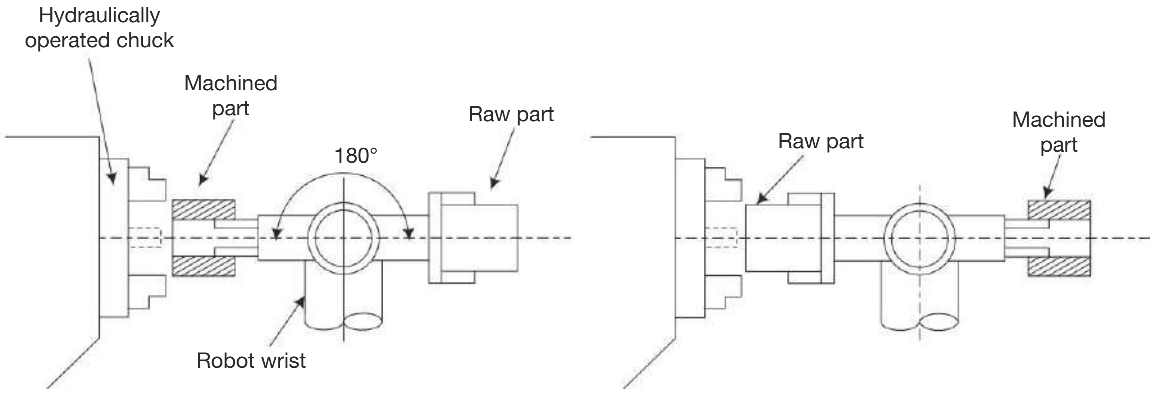


Fig. 21.21 *Dual gripper in operation with a machine-tool loading*

Grippers with Sensors Grippers can be provided with sensors such that they provide tactical information to the control before an action is taken. The use of sensors, therefore, makes a gripper intelligent. These are sometimes called '*sensory grippers*'. The sensors also enable a robot to function in a dynamic fashion by checking its sensors continuously to determine what is happening to the environment around the gripper as it is functioning. A sensory controlled gripper designed by Karlsruhe University, Germany, is shown in Fig. 21.22 with a variety of sensors. Normally, most of the common grippers are provided with one or two sensors only.

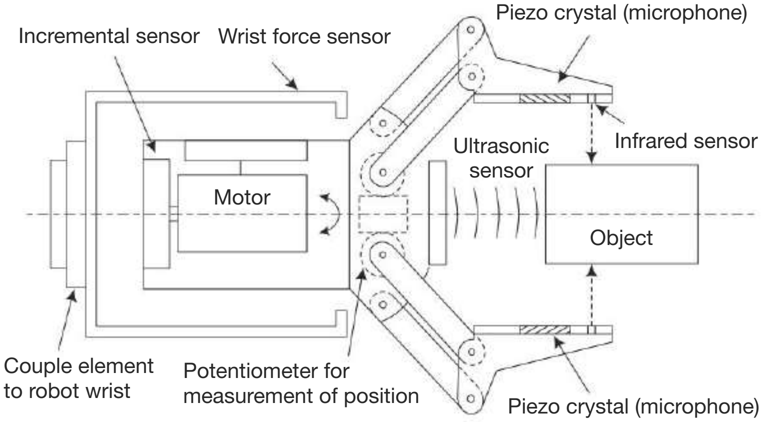
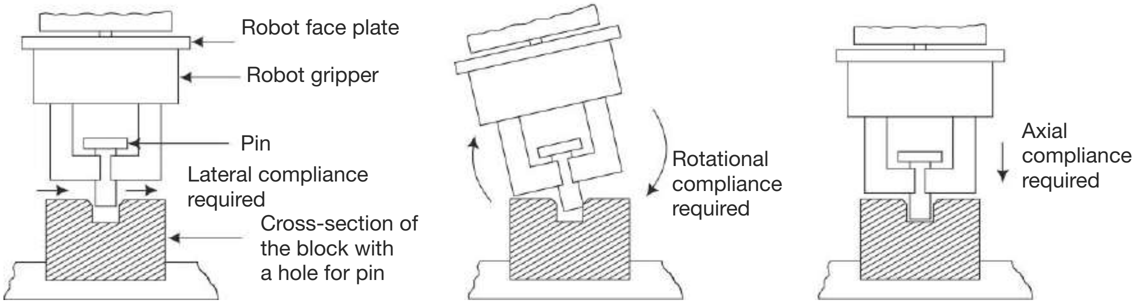


Fig. 21.22 *An example of a gripper with a number of sensors*

Compliance During the assembly operation, when two items need to be fitted, there are likely to be a number of problems since we live a real world and not an ideal one. For example, misalignment between the two parts can cause the assembly to be either not completed or jammed. To reduce this problem, passive devices that react to the forces and torques generated when a device is held on a flexible mount, while attempting to assemble into another item, are employed. Depending upon the type of misalignment, the type of compliance required is decided. Some examples of compliances required are shown in Fig. 21.23. The type of compliances required could be lateral, rotational or axial.



(a) Lateral compliance (b) Rotational compliance (c) Axial compliance

Fig. 21.23 *Compliance of a robot gripper*

Commercially available passive or Remote Centre Compliance (RCC) devices are shown in Fig. 21.24. The RCC devices use combinations of different elastomer pads, shims and geometry to meet the particular application.

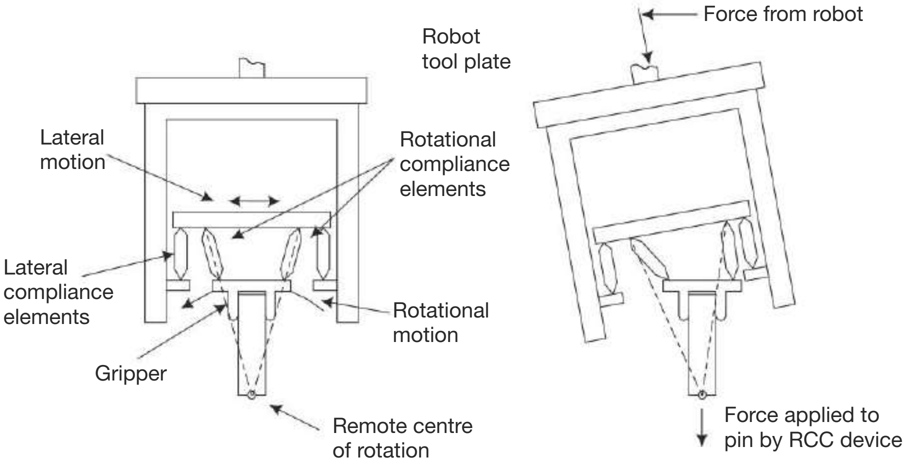


Fig. 21.24 *RCC with lateral and rotational compliance*

21.3.7 Robot Programming

The methods used for development of robot programs, or more generally called teaching a robot, is done as follows:

- Lead by nose
- Teach pendant
- Offline programming

Lead by Nose An experienced operator holds the robot hand and completes the operation manually by moving the robot hand to the various positions. The controller records the actual motions, which can then be used to replay later for actual work. This is suitable for spray-painting applications, but not for material handling.

Teach Pendant In this case, a teach pendant which has all the necessary functions to move the robot is used by the operator to do the job. The operator moves the hand to the various positions and records in the memory the various locations and paths taken to complete the program. Later on, the same program can be used for regular operation. This is the most commonly used method for material-handling application.

Offline Programming These are generally used with the simulation systems where more sophisticated operations, which involve a number of elements within a manufacturing cell, are simulated. The simulation programs have the facilities for defining the machine tools, workpiece geometries, material handling equipment such as robots, conveyors, etc. After the work-cell arrangement is defined, the movement of the robot for work handling can be defined using the simulation language. An actual simulation of the operation can be seen on the workstation screen in wire-frame modelling or realistic shaded image to check the validity of the operation. Once approved, the robot program can be postprocessed for the particular robot to be used.

21.3.8 Robot Programming Languages

A large number of programming languages for writing robot motion programs have been developed. Some examples are RAPT, SIGMA, MAL, ROBEX, MCL, VAL, and AML. These are somewhat similar to typical computer programming languages with the vocabulary words related to robot functionality.

VAL II used by Unimation Robots

VAL stands for Victor's Assembly Language. This is the programming language used by Unimation Robots, which were the first major commercial robots developed. The programming language is similar to BASIC. It has a complete set of vocabulary words for writing and editing robot programs. Generally, the robot programs have three basic modes of operation. They are the following:

(a) Monitor mode In this mode, the user defines the various positions to be used in the program. These positions are taught using the teach pendant and stored into the memory to be used in the program.

(b) Editor mode This mode is used by the user to write new programs and edit the existing ones. In this mode, the syntax checking is done.

(c) Run mode This is the mode in which the robot actually executes the program. The robot actually performs the sequence of motions in the run mode.

A complete program forms a set of statements to direct the sequence of motions of the PUMA robot. One statement usually corresponds to one movement of the robot's arm or wrist. For example, Move to a point; Move to a point in a straight-line motion; Open gripper; Close gripper.

The various commands available in VAL II can be broadly classified into the following categories:

- Constants and variables
- Motion commands
- End effector and sensor commands
- Computation and operations
- Program control
- Monitor-mode commands

Constants and Variables Constants and variables are given in a program to carry out the necessary computational procedures, or to assign certain types of data such as positional information to be used in the program at various locations. The rules to be followed are similar to any computer language such as BASIC. Some examples are shown below to explain the syntax used.

```
HERE P1
```

In this example, the current location of the robot is assigned to the variable P1.

```
POINT PB = P2
```

In this example, the point location stored in P2 is assigned to the variable PB.

```
SET A = 12
```

In this example, a real value of 12 is assigned to the variable named A.

```
SETI NBR = 1
```

In this example, an integer value of 1 is assigned to the variable named NBR.

Motion Commands These are the most important part of a robot command set. These help in the movement and control of the manipulator arm.

MOVE Moves the robot to the location and orientation specified by the variable symbol A.

```
MOVE A
```

MOVES Moves the robot along a straight-line trajectory to the location and orientation specified by the variable symbol A.

```
MOVES A
```

APPRO Moves the end effector or tool to the position defined by the variable symbol A, but offsets it along the tool *Z*-axis by the distance given in millimetres.

```
APPRO A, 50
```

APPROS Similar to APPRO except that the move to the vicinity of the desired point is made along a straight-line trajectory

DEPART Moves the tool back from its current position to the distance given, along the current *Z*-axis of the tool.

```
DEPART 50
```

DEPARTS As DEPART but along a straight-line path.

ALIGN Allows the tool *Z* coordinate to the nearest robot world coordinate system axis. This is used before teaching a series of locations.

SPEED or SP Specifies the speed of all subsequent robot motions under program control and precedes the execute (EX) command.

```
SPEED 50
SP 50            ; 50% of the standard speed
SPEED 25, IPS    ; 25 inch per second speed
```

End Effector and Sensor Commands These commands operate the end effector or any of the sensors that are part of the robot.

OPEN and CLOSE Opens and closes the gripper to occur during the execution of the next motion.

OPENI and CLOSEI Opens and closes the gripper immediately without waiting for the start of the next motion.

WAIT Causes the program to wait for the signal from a specified input channel to reach a particular option. For example,

```
WAIT 1           ; wait from input channel 1
WAIT -3          ; wait until input channel 3 is low
```

SIGNAL Used to output a signal to the specified value. Examples are

```
SIGNAL 1         ; turn on the output signal 1
SIGNAL –1        ; turn the output signal off
SIGNAL 6,-3,5    ; Signal to say robot turn on output 6 to high, turn off 3 and turn on channel 5
```

Computation and Operations It is also possible to carry out simple arithmetic operations as part of the program as shown below for incrementing the variable NBR.

```
SETI NBR = NBR + 1
```

SHIFT Used to update the positional values

```
SHIFT A BY –200, 30        ; -200 mm along X-axis and 30 mm along Y-axis
```

Program Control

DO Causes the robot to execute a specified programming instruction.

```
DO MOVE P1
```

GOTO Transfers the control of the program to the label specified.

```
GOTO 95          ; program control moves to statement 95
```

Arithmetic IF allows for comparing the variables with some constant values and accordingly branches the program.

```
IF NBR EQ 0 THEN 60
```

SUBROUTINE   Defining a subprogram

GOSUB is used to call and execute a subprogram

GOSUB TRAY   Go to subprogram TRAY

RETURN is the instruction in a subprogram which when executed returns the control to the next executable statement in the program.

EXIT or E   Exits from the program and control is returned to the monitor mode

DELAY is used to specify the time required for a new tray to be kept in place.

```
DELAY 10        - Robot idles for a period of 10 seconds
```

Monitor-Mode Commands Program instructions are entered into the memory to form programs by using the Monitor Command EDIT.

EDIT or ED   This statement is used to create or modify the program named in the statement.

```
EDIT TEST
ED TRAY
```

EXECUTE or EX   Commands the robot to execute the program names. The user can specify how many times the program is to be executed by giving the number of executions following the program name.

```
EX TEST
EX TEST, 5
```

HERE   Used to define the current position of the robot by a symbol specified by the user.

```
HERE A
HERE PT1
```

The user moves the robot to the desired position under manual control by means of the teach pendant. When the robot is in the desired location, the user defines that position with the HERE statement. The CRT displays the PUMA's six axis positions. The operator then depresses the carriage return to proceed.

A simple VAL II program to move a part from position *A* to *B* is shown below.

```
A.TO.B              Program name
OPEN                open the robot gripper during next instruction
MOVE POSA           move to position A
CLOSEI              close gripper immediately
MOVE POSB           move to position B
```

Though the above program is a working one, it is not a good one. The care that needs to be taken while making the robot move in practical situations is not taken into account while writing the above program. The main point to be considered is the path taken by the robot from position A to position B, which may not be a straight line. This actually depends on the robot control system. As a result, there is the likelihood of the gripper hitting the part rather than picking it up because of the rapid motion of the robot arm. Hence, the program is modified by adding the APPRO statement, whereby the robot approaches the part to a position within a specified distance, first and then moves vertically over the part so that it is able to pick up the part without any problem. Such a program as modified is shown below.

```
GOOD.A.TO.B          program name
OPEN                 open the robot gripper
APPRO POSA, 25       approach position A with in 25 mm
MOVE POSA            move to position A
CLOSEI               close gripper immediately
DEPART 25            back away from position A by 25 mm
APPRO POSB, 25       approach position B with in 25 mm
MOVE POSB            move to position B
OPENI                open the robot gripper immediately
DEPART 25            back away 25 mm
```

Let us now improve the above program further by looking at the actual input and output places for the part. The robot will be picking up the parts from a chute and placing them in successive boxes. For such an activity, the possible sequence of operations are

1. Move to a location above the part in the chute
2. Move to the part
3. Close the gripper jaws
4. Remove the part from the chute
5. Carry the part to a location above the box
6. Put the part into the box
7. Open the gripper jaws
8. Withdraw from the box

The corresponding VAL II program that is literally the translation of the above individual operations is shown below.

```
BOXFILL              program name
APPRO PART, 50       move to a location above the part in the chute
MOVES PART           move along a straight line to the part
CLOSEI               close the gripper jaws
DEPARTS 150          withdraw the part from the chute along a straight-line path
APPROS BOX, 200      move along a straight line to a location above the box
MOVE BOX             put the part into the box
OPENI                open the gripper jaws
DEPART 75            withdraw from the box
```

Let us now try a more ambitious program, which is typical of the applications for which the robots are generally used with machine-tool applications. In such situations, parts are picked up and then arranged on a pallet in a specified sequence, which is called *palletising*. For this purpose, the parts need to be arranged on a pallet in rows and columns as shown in Fig. 21.25. There are a total of 5 rows and 6 columns with a total of 30 parts per tray. The distance between the rows is 30 mm and that between the columns is 40 mm. The robot starts with the lower left corner of the tray, and then proceeds to place individual parts that are offset by the column distance along the *X*-axis. Once all the parts in one row are placed, the robot then positions to place the part at the beginning of the next row by offsetting the path by an amount equal to the total distance moved along the row, with the distance to be moved along the *Y*-axis by the row distance.

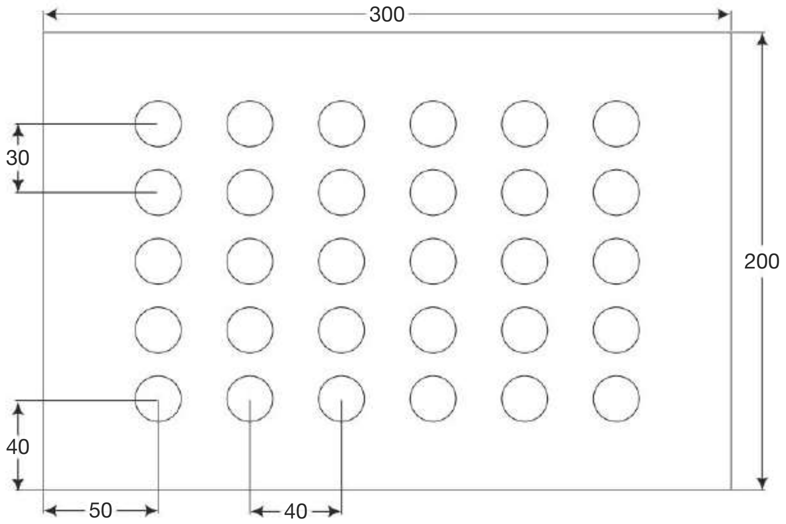


Fig. 21.25 *Palletising of parts by a robot using indexed loops*

The VAL program is shown below for the above palletising application. The program is divided into three parts. The first part is the main program where the total action is shown. The other two are subprograms, which are used to carry out the repetitive actions required. The PUT subprogram has all the necessary statements that are required for moving the part from the input to the output tray. The TRAY subprogram coordinates the calculation of the actual positions of the individual parts on the pallet, which are arranged row-wise first. The programs are provided with the meaning for each statement to explain the procedure.

```
      PALLET
95    SET POSA = AA             start with corner position
      SETI NBR = 0              initialise the part counter NBR = 0
85    GOSUB TRAY                go to subprogram TRAY
      GOSUB PUT                 go to subprogram PUT
      SETI NBR = NBR + 1        update the counter
      IF NBR LT 30 THEN 85      if the NBR is less than 30, move another
      TYPE NEW TRAY, PLEASE     type a message for new tray
      DELAY 10                  delay 10 seconds
      GOTO 95
      SUBROUTINE PUT            program name
      OPEN                      open the robot gripper
      APPRO POSA, 25            approach position A with in 25 mm
      MOVE POSA                 move to position A
      CLOSEI                    close gripper immediately
      DEPART 25                 back away from position A by 25 mm
      APPRO POSB, 25            approach position B with in 25 mm
      MOVE POSB                 move to position B
```

```
        OPENI                       open the robot gripper immediately
        DEPART 25                   back away 25 mm
        RETURN
        SUBROUTINE TRAY             tray subprogram
        IF NBR EQ 0 THEN 60         if it is the first position in the row
        IF COL EQ 6 THEN 50         if the end of the row is reached?
        SHIFT A BY 40               shift 40 mm along X-axis to pick the next part
        SETI COL = COL + 1          change the column number
        RETURN                      return to the main program
   50   SHIFT A BY –200, 30         – 200 mm along X-axis and 30 mm along Y-axis
   60   SETI COL = 1                initialise the column counter
        RETURN                      return to the main program
```

In the above program, it is assumed that the tray will be replaced within a timespan of 10 seconds. If that does not happen, the robot will still go through the required motions of palletising at an empty space. To take care of such a situation, a more appropriate program would be to incorporate sensors to tell the robot about the presence of the new tray which is indicated by the sensor output 1 as shown below.

```
        IF NBR LT 30 THEN 85
        SIGNAL 1                    Signal to say robot is ready for a new tray
        WAIT 1                      wait from input channel 1
        SIGNAL –1                   turn the output signal off
        GO TO 95
```

21.3.9 Cost of Acquisition of a Robot

The cost of acquisition of a robot is one of the most important considerations. The general feeling is that it is too expensive. However, one has to make a really careful study of all the options before finalising an application involving robots. Figure 21.26 shows a typical curve indicating the effectiveness of the various methods of production that are practised in manufacturing industries.

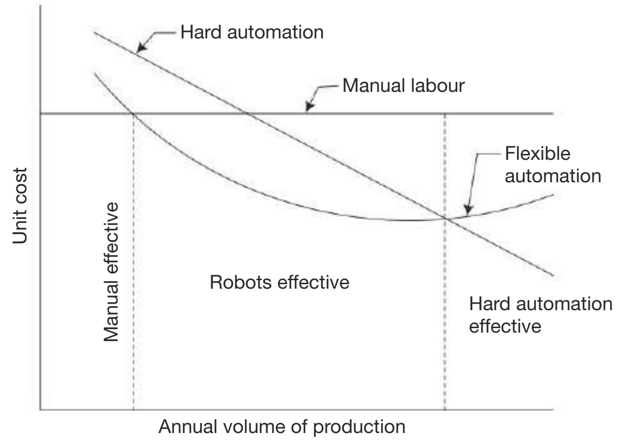


Fig. 21.26 *Trend of robotised and other manufacturing methods*

In order to analyse the economy of introducing robots into operation, it is important to consider all the costs involved. Robots are becoming increasingly attractive at present because of the increase in the direct labour costs. With the economies improving, naturally the wages of the workers will increase which reflects in the increase in direct labour cost as shown in Fig. 21.27.

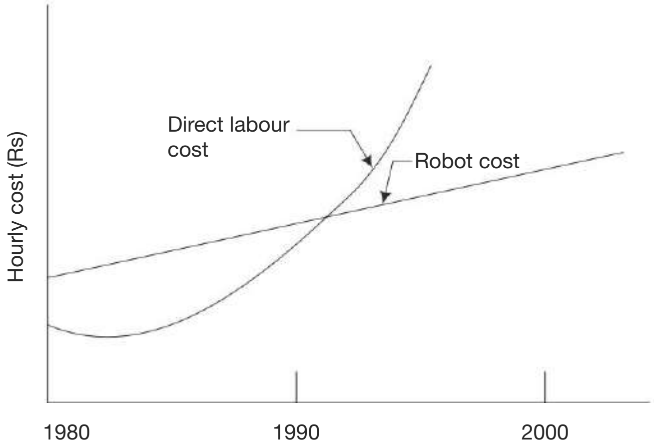


Fig. 21.27 *Expected changes in labour costs*

Robot Investment Costs

- Cost of robot (manipulator and controller)
- Engineering cost (for planning and other initial expenses)
- Programming cost
- Installation cost (including safety provisions)
- Tooling cost (cost of grippers, part positioners and other special jigs and fixtures)
- Miscellaneous costs including those in the cell

Robot Operating Costs

- Direct labour cost (operator)
- Supervisory costs of the personnel engaged excluding maintenance personnel
- Maintenance costs (including labour, spare parts, insurance other services)
- Costs of utilities and servicing facilities (including energy, safety equipment and other consumables)
- Training costs (frequent and regular training and refresher courses are necessary for the staff)

In addition to these direct costs, which can be identifiable, a number of indirect benefits can be realised that are difficult to quantify. They are

- Inventory carrying cost (both in process and finished goods)
- Materials utilisation and savings
- Reduced scrap and rework
- Labour absenteeism and turnover
- Space utilisation

- Improved management control
- Improved worker participation

Hence, when one has to consider the economic aspects, all the above costs in some form or other would have to be accounted for before finalising the acquisition of a robot for an application.

In the following, a simple example has been worked out to show the economic justification for the introduction of a robot. Let the following be the costs associated with a robotisation programme.

Cost of a robot and accessories, C_r = Rs 2 000 000

Total operating cost per year = Rs 300 000

Robot life period = 5 years

To calculate the operating cost of the robot, we first assume that the depreciation is calculated based on the straight-line method for simplicity. Hence, the costs are

Yearly depreciation, $D = \dfrac{2\ 000\ 000}{5} = 400\ 000$

Annual maintenance costs, M = Rs 200 000

Annual value of increased output, O = Rs 900 000

Assume that one operator working on a job is replaced by the robot. The operator would be working for 2000 hours in a year considering all the holidays and leaves. Assuming a salary of Rs 10,000 per month and other benefits and overheads, the operator cost (L) may work out to be about Rs 100 per hour.

For One-shift Operation

Total operating cost, C_o = Depreciation + Operating cost + Maintenance cost
= 400 000 + 300 000 + 200 000
= Rs 900 000

Total savings, C_s = Operator salary + Increased output
= 100 × 2000 + 900,000
= Rs 1100 000

Net savings = $C_s - C_o$ = 1100 000 – 900 000 = Rs 200 000

$$\text{Annual rate of return} = \frac{200\ 000}{2000\ 000} = 10\%$$

For Two-shift Operation

Total operating cost, C_o = Depreciation + Operating cost + Maintenance cost
= 400 000 + 2 × 300 000 + 2 × 200 000
= Rs 1400 000

Total savings, C_s = Operator salary + Increased output
= 2 × 100 × 2000 + 2 × 900 000
= Rs 2200 000

Net savings = $C_s - C_o$ = 2 200 000 – 1 400 000 = Rs 800 000

$$\text{Annual rate of return} = \frac{800\ 000}{2\ 000\ 000} = 40\%$$

As seen from the above, the robotisation programmes would become very attractive if they are conceived for longer working hours than in the case of single-shift operations like in conventional methods. It is even possible to conceive a round-the-clock operation to make the payback period as short as possible. However, care has to be taken to see that the associated facilities are properly developed to ensure that the use of the robot is optimised with the best possible programmes and tooling.

21.3.10 Robot Applications

Robots are being increasingly used in industrial applications in a variety of situations. Industrial applications of robots can be broadly classified into three categories:

- Material handling
- Processing operations
- Assembly operations

Before choosing of a robot for any given application, we should carefully consider a number of factors related to the particular operation in question. Some considerations are

- Part-positioning requirements including the orientation
- Gripper design considerations
- Actual distances involved in the cell for the various components
- The specifications of the robot in terms the workspace, payload capacity and the accuracy and repeatability
- Robot configuration in terms of degrees of freedom and control required
- Utilisation of the various components of the work cell

Material-Handling Material handling operations in manufacturing require a variety of tasks that need to be completed for moving material from one point to another depending upon the processing requirements. These tasks can be categorised as follows:

- **Pick-and-place operations**, which involve the robot to move the part from one location to the other, i.e., such as from one conveyor to the other for changing the part-movement direction. A major class of these operations are involved in palletising and depalletising operations.
- **Machine loading and unloading**, which involves the loading of machines with unmachined parts, and unloading of the part after the machining process is completed. A number of processing machines are used such as these.
    - Die casting In die casting, the part needs to be ejected when the die is opened. The temperature of a typical casting when ejected may be of the order of 250°C to 400°C which is difficult to handle in view of their short cycle times of the order of 1 minute. This is a very unpleasant job for a human operator in view of the high temperature, short cycle time, dirty atmosphere and heavy castings. Thus, the use of a casting-extracting robot is an ideal application, and many die-casting machines have a dedicated robot to do this job because of the short cycle times.
    - Injection moulding This operation is similar to die casting, except that the temperature of the part is low compared to it. Also, it is necessary to trim the gating and runners from the moulding when it is taken from the mould. This can be accomplished by a trimming set-up placed closer to the injection-moulding machine, such that the robot can place the moulding directly into the trimming and set-up and then transfer to its bin after trimming, thus completing the production cycle. Also, if the cycle time is short then the robot can be dedicated to one moulding machine. However, if the cycle times are large, more than one moulding machine can be attended to by a single robot.

- **Forging and other hot working operations** This is an operation that would require a tolerance of very high temperatures, typically of the order of 1100°C for hot forging of steels. This is not a very conveniently amenable operation like die casting and injection moulding. The operations for which robots could be used are drop forging and upset forging, where the part needs to be given multiple blows to get the necessary deformation. The human operator uses his dexterity and visual sense to identify the likely problems and to manipulate the part inside the die. This part is difficult for a robot. As the temperature of the part changes during the forging process, the size of the part changes, which need to be accommodated by the gripper. Also, the design of special grippers for the generally smaller-volume cycles in forging have to be addressed properly by the designers.
- **Machining operations** Machining is a very important operation and practically all the industrial parts have some amount of machining operation being carried out at some stage. The machining processes that are most common are drilling, milling, turning and grinding. In addition to these processes, specialised processes such as gear cutting are also carried out. A robot is generally used to load a raw casting or forging on to the machine to be properly located and machined. After the completion of the machining operation, the part is unloaded from the machine. Since this being a major application of robotics, it is separately discussed.
- **Sheet-metal press operations** Sheet-metal operations, in particular blanking and piercing operations, require a very high-speed feeding operation. These require some special characteristics and being a major application of robotics, is separately discussed.

Machine Loading A number of factors need to be considered for the successful application of a machine loading and unloading operation.

- **Type of gripper**, in particular the use of a double gripper. This greatly facilitates the operation by increasing the metal cutting time of the machine tool by reducing the idle time. An example for this is worked out a little later to demonstrate the advantage.
- **Large number of degrees of motion**, to properly orient the part in the machine tool fixture.
- **Good repeatability**. In order to position the part properly at the machining position, a very high accuracy is required for the robot to achieve the required precision.
- **Palletising capability**. For mid volume production, parts need to be palletised for transfer to and from the machining cell. The robot controller should have sufficient programming capability to achieve this.

The type of grippers to be considered for machine loading is their ability to handle a variety of parts and materials. To that extent, some types of grippers to be considered are

- Opposing dual V for self-centring parts during pick-up.
- Offset opposing dual V for long shafts
- Vacuum for non-magnetic parts (aluminium, plastic, wood, etc.) or parts that have a precision finish
- Magnetic grippers for picking up parts that cannot be picked up by the external or internal details
- Three jaw for round cylindrical stock
- Simple two jaw for flat stock

The choice of robot to be used for a given material-handling application depends on the requirements in terms of flexibility as well as the space available.

- An articulated robot is good to cater to a smaller number of machine tools (maximum of four) in view of its ability to reach all of them (Fig. 21.28). However, the articulated arm if it is floor mounted is likely to provide obstruction to the other elements in the cell such as conveyors, manual operations, etc.

- Another approach is to keep an articulating arm on a pedestal set on a rail track sled that is positioned between two rows of machine tools (Fig. 21.29). This approach requires the placement of heavy metal plates across the rail-track pit for manual operations with the cart traffic. The pit is a potential collection place of trash and debris and it requires a seventh axis of motion.

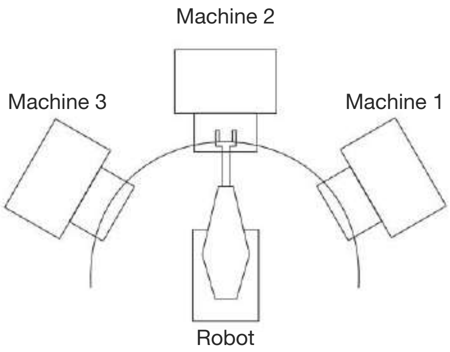


Fig. 21.28 *Articulated robot at the centre of a cell*

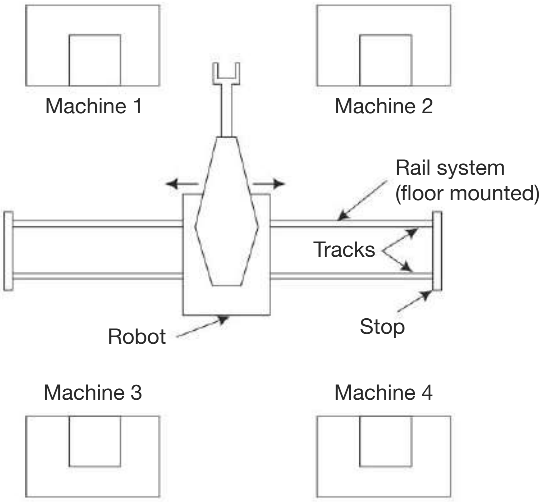


Fig. 21.29 *Articulated robot on a mobile pedestal*

- To get more working space around machine tools, an approach that is possible is an inverted articulating arm pedestal set on a monorail track between two rows of machine tools (Fig. 21.30). This is an expensive configuration and its ability to repeat its position when lifting heavy parts may be a concern. A typical articulated arm attached to a monorail is shown in Fig. 21.31.

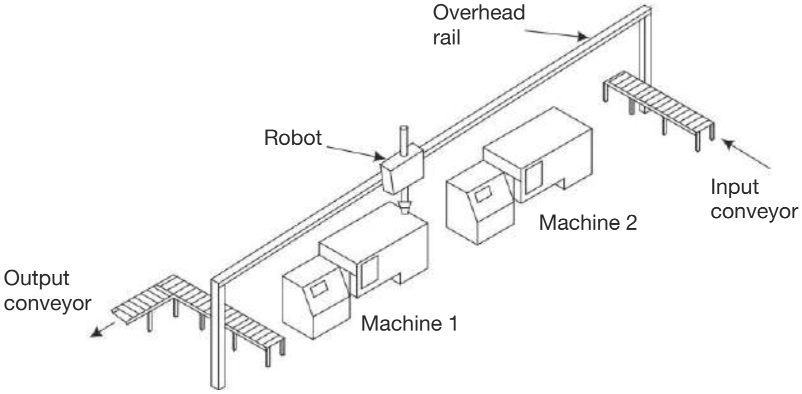


Fig. 21.30 *Articulated robot at the centre of a cell*

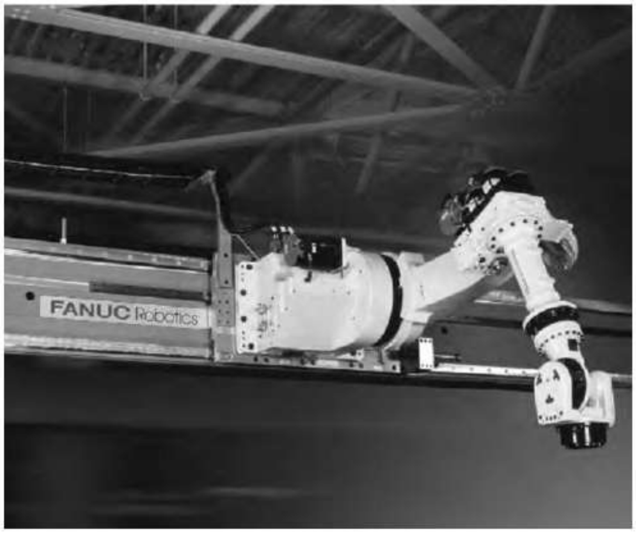


Fig. 21.31 *Articulated arm attached and moving on a monorail (Courtesy, Fanuc)*

- Another approach that often helps for attending to a large number of machines is a gantry system where the robot is attached to a single axis or a two-axis positioning overhead system. The gantry is supported by a rigid overhead frame, which can be parked in a non-interference position when not in use or undergoing maintenance. The gantry approach also gives us the freedom to move machines into or out of the cell without causing any major production disruptions due to the X-Y axes always being located overhead. A typical example of a cell for machining heavy castings is shown in Fig. 21.32.

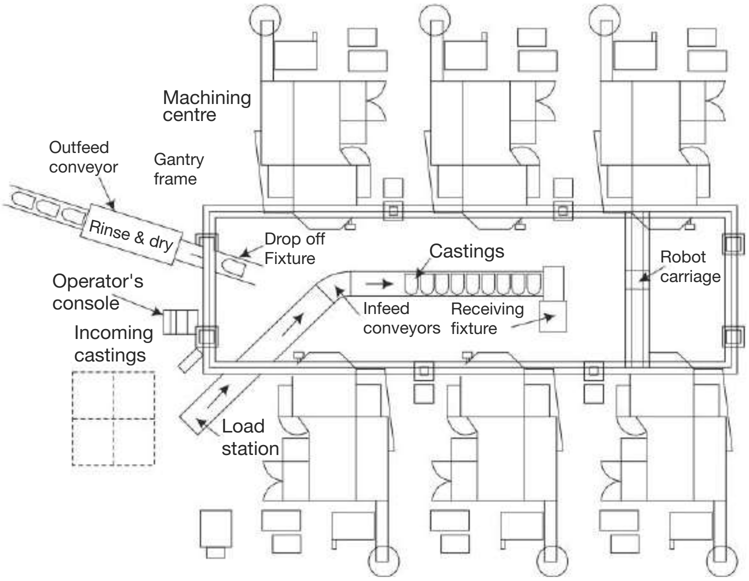


Fig. 21.32 *A typical machining cell with a number of milling machines with a gantry robot*

Robot Machine Loading System Productivity

Example 21.1 An example of a robot machine loading/unloading system is shown in Fig. 21.33, which consists of four identical machining centres, each being attended by a robot. The parts are transferred between the indexable conveyor and the machining centre by the robot. The cycle time of each machining operation is 12 seconds. Calculate the total production rate of the set-up, if (i) all the machines are (i) working parallel (same operation being carried at all the machines and the part is completely machined in a single machine tool), and (ii) in series. Calculate the daily production rate for one, two and three-shift operation.

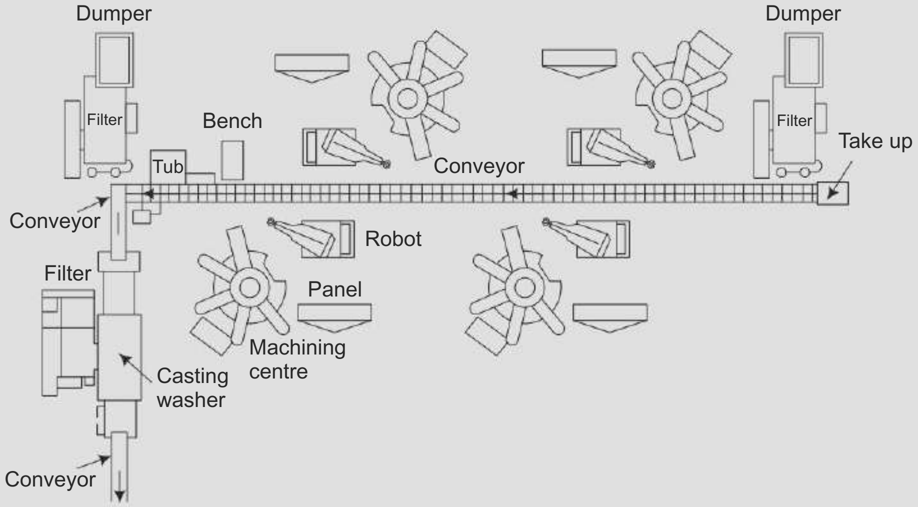


Fig. 21.33 *A typical machining cell with a number of milling machines with robots and a conveyor*

Solution Assuming no stoppage, all 8 hours of operation for shift are available.

Total time available for production = 8 × 3600 = 28 800 seconds

Parallel Operation In parallel operation, one part is completely processed in a single machine, so that for every 8 seconds, four parts come out of the system.

$$\text{Production rate} = \frac{4 \times 28\,800}{12} = 9600 \text{ parts per shift}$$

$$= 19\,200 \text{ for two shifts}$$

$$= 28\,800 \text{ for three shifts}$$

Series Operation In series operation, one part is completed only after it passes through all the four machines. Hence, the cycle time for the part is 48 seconds. However, since all are working simultaneously, one part comes out of the system for every 12 seconds.

$$\text{Production rate} = \frac{28\,800}{12} = 2400 \text{ parts per shift}$$

$$= 4800 \text{ for two shifts}$$

$$= 7200 \text{ for three shifts}$$

Example 21.2 A robot machining cell is shown in Fig. 21.34 consisting of a number of workstations (turning machines and grinding machines). The robot loads and unloads the workstations from the central conveyor. The average robot operation times are

Pick up part from conveyor including waiting time	2.6 s
Move robot from conveyor to machine	1.7 s
Deposit finished part on conveyor	0.3 s
Load part on to the machine	1.1 s
Unload part from machine	0.8 s

Suppose each of the workstations has an operation cycle of 24 seconds. Assuming an average of 20 per cent system downtime for maintenance, clearance of malfunctions, and other causes,

(a) What is the daily eight-hour shift production assuming a one-handed robot gripper?

(b) What will be the improvement in production rate if a two-handed robot gripper is employed?

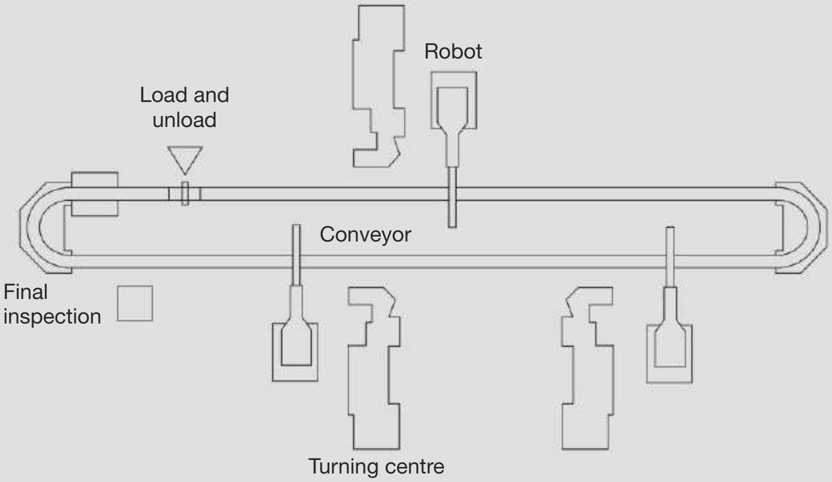


Fig. 21.34 *A machining cell consisting of a number of workstations with robots and a central conveyor*

Solution

For One-Handed Gripper Assume that the robot has no other duties and waits at the machine in order to unload it as soon as the machine cycle is complete. Arbitrarily selecting the beginning of the machine operation cycle as the beginning of the system cycle, the typical operation sequence is

Machine operation cycle	24.0 s
Unload machine	0.8 s
Move to conveyor	1.7 s
Deposit finished part on conveyor	0.3 s
Pick up new part	2.6 s
Move to machine	1.7 s
Load into machine	1.1 s
Total	32.2 s

Machine availability = 80% (taking 20% for maintenance)

Assuming no stoppage, all 8 hours of operation for shift are available.

Total time available for production = 8 × 3600 = 28 800 seconds

$$\text{Production rate} = \frac{28\,800 \times 0.8}{32.2} = 715.52 \text{ parts per shift}$$

For Two-Handed Gripper In the case of a two-handed gripper, the robot operations of move to the conveyor, deposit part, pick up the new part, and move to the machine could all be performed by the robot *during* the machine operation cycle and, therefore, can be omitted from the timed sequence. The appropriate operation sequence would be

Machine operation cycle	24.0 s
Unload machine	0.8 s
Load into machine	1.1 s
Total	25.9 s

$$\text{Production rate} = \frac{28\,800 \times 0.8}{25.9} = 889.58 \text{ parts per shift}$$

The increase in production by employing the double-handed gripper is

$$\frac{889.58 - 715.52}{715.52} = 24.3\%$$

This increase has been achieved by having a double gripper, which is relatively less costly compared to an additional robot or other processing equipment. Another point that can be noted is that in this case, such a large amount of improvement was possible because the cycle time of the workstation is much larger compared to the handling times of the robot. To explain this, a second example is worked out below with a short processing time.

Example 21.3 For Example 21.2, suppose each of the workstations has an operation cycle of 1 second. With all other data remaining the same, calculate the production rates.

Solution

For One-Handed Gripper Assume that the robot has no other duties and waits at the machine in order to unload it as soon as the machine cycle is complete. Arbitrarily selecting the beginning of the machine operation cycle as the beginning of the system cycle, the typical operation sequence is

Machine operation cycle	1.0 s
Unload machine	0.8 s
Move to conveyor	1.7 s
Deposit finished part on conveyor	0.3 s
Pick up new part	2.6 s
Move to machine	1.7 s
Load into machine	1.1 s
Total	9.2 s

Machine availability = 80% (taking 20% for maintenance)

Assuming no stoppage, all 8 hours of operation for shift are available.

Total time available for production = 8 × 3600 = 28 800 seconds

$$\text{Production rate} = \frac{28\,800 \times 0.8}{9.2} = 2504.35 \text{ parts per shift}$$

For Two-Handed Gripper In this case, the operation times of the robot are much higher compared to the processing time. As a result, the machine processing is done during the robot operation time. The appropriate operation sequence is

Unload machine	0.8 s
Move to conveyor	1.7 s
Deposit finished part on conveyor	0.3 s
Pick up new part	2.6 s
Move to machine	1.7 s
Load into machine	1.1 s
Total	8.2 s

$$\text{Production rate} = \frac{28\,800 \times 0.8}{8.2} = 2809.76 \text{ parts per shift}$$

The increase in production by employing the double-handed gripper in this case is

$$\frac{2809.76 - 2504.35}{2504.35} = 12.2\%$$

It can be noticed that compared to the previous exercise, the improvement is only 12.2% compared to 24.3% by employing the double gripper. Still that is an improvement without much economic outlay and, therefore, should be pursued while planning for the adoption of robots in material handling application linking with workstations.

Sequential Machine Loading The analysis of robot machine-loading applications in the case of dedicated robots is relatively easy as seen in the previous exercise. However, it becomes a bit more complicated when a single robot has to attend to the task of feeding several workstations in an organised sequence. In such cases, the automation engineer has to time and plan all operations carefully such that the robot tasks can be appropriately planned. While preparing the robot programming, the automation engineer has to anticipate cycle completions at appropriate stations and move to these stations in advance to shorten machine idle time while waiting for the robot. Things become relatively complex if a single robot has to attend to more than two machines. We will study an example with a robot attending two workstations to see how the process can be facilitated.

Example 21.4 A single robot serves two machines sequentially as shown in Fig. 21.35. The robot times for each station are 0.1 minute for pick-up and 0.1 minutes for release. The move times are given in the Table 21.4. Machine-processing times are 6 minutes for machine B and 9 minutes for machine C. Determine the cycle time for this system. Calculate the system's cycle time and production rate.

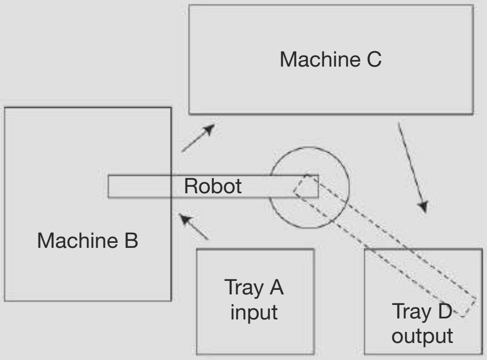


Fig. 21.35 *A machining cell consisting of two workstations attended by a robot with input and output conveyors*

Table 21.4 *Move times in minutes*

From	*To*			
	Tray A	*Machine B*	*Machine C*	*Tray D*
Tray A	—	0.3	0.3	0.3
Machine B	0.3	—	0.3	0.3
Machine C	0.3	0.3	—	0.3
Tray D	0.3	0.3	0.3	—

Solution In order to analyse the case, it is necessary to construct a table where the actual timings for the various operations that need to be carried out sequentially can be laid out. From such a table, it is possible to calculate the cycle time when the cycle starts to repeat itself. While filling the table, you have to take care to see that a workstation can be unloaded only when it finishes the machining operation, and not when the robot is available. The robot will have to wait for that operation to be completed. Table 21.5 shows the analysis of the operations in sequential order.

Table 21.5 *Robot cycle analysis*

Status before operation						
Clock time	*Robot position*	*Stations loaded*		*Operation*	*Time required*	*Scheduled completion*
		B	*C*			
0	A			Pickup at A	0.1	0.1
0.1	A			Move to B	0.3	0.4
0.4	B			Load machine B	0.1	0.5
0.5	B	#1		Processing at B	6.0	6.5
6.5	B	#1		Unload B	0.1	6.6
6.6	B			Move to C	0.3	6.9
6.9	C		#1	Load C	0.1	7.0
7.0	C		#1	Processing at C	9.0	16.0
7.0	C		#1	Move to A	0.3	7.3
7.3	A		#1	Pickup at A	0.1	7.4
7.4	A		#1	Move to B	0.3	7.7
7.7	A	#2	#1	Load machine B	0.1	7.8
7.8	A	#2	#1	Processing at B	6.0	13.8
7.8	A	#2		Move to C	0.3	8.1
16.1	C	#2		Unload C	0.1	16.2
16.2	D	#2		Move to D	0.3	16.5
16.5	D	#2		Unload at D	0.1	16.6*
16.6	D	#2		Move to B	0.3	16.9
16.9	B	#2		Unload B	0.1	17.0
17.0	B			Move to C	0.3	17.3
17.3	C		#2	Load C	0.1	17.4
17.4	C		#2	Processing at C	9.0	26.4

Contd...

Contd...

17.4	C		#2	Move to A	0.3	17.7
17.7	A		#2	Pickup at A	0.1	17.8
17.8	A		#2	Move to B	0.3	18.1
18.1	B		#2	Load machine B	0.1	18.2
18.2	B	#3	#2	Processing at B	6.0	24.2
18.2	B	#3	#2	Move to C	0.3	18.5
26.4	C	#3		Unload C	0.1	26.5
26.5	C	#3		Move to D	0.3	26.8
26.8	D	#3		Unload at D	0.1	26.9*
26.9	D	#3		Move to B	0.3	27.2
27.2	B	#3		Unload B	0.1	27.3
27.3	B			Move to C	0.3	27.6
27.6	C		#3	Load C	0.1	27.7
27.7	C		#3	Processing at C	9.0	36.7
27.2	C		#3	Move to A	0.3	28.0
28.0	A		#3	Pickup at A	0.1	28.1
28.1	A		#3	Move to B	0.3	28.4
28.4	B	#4	#3	Load machine B	0.1	28.5
28.5	B	#4	#3	Processing at B	6.0	34.5

The lines labelled by * in Table 21.5 are the points in the cycle in which the station D is unloaded, which represents the arbitrary endpoints in our cycle. After the system becomes fully loaded, the difference in clock times between successive 'unload D' operations represents the system cycle time.

$$\text{Cycle time} = 26.9 - 16.6 = 10.3 \text{ minutes}$$

Though we have taken 'Unload D' as an arbitrary point for calculating the cycle time, we could have obtained the cycle time from any two identical operations after the system reached its normal operating procedure.

Assuming no stoppage, all 8 hours of operation for shift are available.

Total time available for production = 8 × 60 = 480 minutes

$$\text{Production} = \frac{480}{10.3} = 46.4 \text{ units/shift}$$

It can be noticed from the cycle analysis that the robot has to wait a substantial amount of time because of the large processing times, compared to the material handling and moving times. Also, the procedure that we have followed in Table 21.5 is intuitive but tedious. It will become more complex when there are more than two machines to be attended to sequentially. In such cases, it is possible to follow another simpler procedure to calculate the cycle time.

In this procedure, we will divide the total robot cycle times into two major segments:

- Unload a given machine till same machine is reloaded—Cycle segment α
- All other robot operations—Cycle segment β

Determine the maximum machine time M among all machine stations.

(a) If $M \le \beta$, total system cycle time $T = \alpha + \beta$

(b) If $M \ge \beta$, total system cycle time $T = \alpha + M$

Table 21.6 *Robot cycle time analysis*

Operation	*Time required*	
Unload C	0.1	
Move to D	0.3	
Unload at D	0.1	
Move to B	0.3	Cycle segment α = 1.3 minutes
Unload B	0.1	
Move to C	0.3	
Load C	0.1	
Move to A	0.3	
Pickup at A	0.1	
Move to B	0.3	Cycle segment β = 1.1 minutes
Load machine B	0.1	
Move to C	0.3	

Maximum machining time, $M = 9$ minutes

Since $M \geq \beta$, total system cycle time $T = \alpha + M = 1.3 + 9 = 10.3$ minutes

In Table 21.6, we arbitrarily selected the machine station C to compute cycle segments α and β. However, it is possible to use any other machine and the result will be the same.

As can be seen from the above example, in cases where the robot portion of the cycle is constant from machine to machine, the recognition and computation of robot cycle segments and greatly simplify the computations for total cycle time and system production rates for sequential machine loading applications of robots.

Sheet-Metal Press Feeding Operation Sheet metal press shops are one of the most capital-intensive areas in manufacturing. Profitability of these facilities depend on the availability, productivity and manufacturing quality. The need for maximum utilisation of cost-intensive machinery and the rising personnel cost compel manufacturers to adopt the use of robots for material handling. This is also helped by the growing number of workpiece variations, smaller batch sizes, higher quality requirements and the mounting pressure to cut costs. The demands placed on these manufacturing systems have previously been met to only a limited degree in the case of production lines with individual presses. Industrial robots are well suited for fulfilling the high requirements involved in interlinking (loading and unloading) individual presses.

Safety is another criterion that prompts the use of robots in press shops. It is also important that the press dies as well as the working people should be protected, since there is often only one single die available for each type of part. Prolonged downtime as a result of a damaged press die can lead to critical delays in the subsequent manufacturing operations.

The problems posed during the handling of the workpieces, is the high acceleration forces acting on the panels during transfer. Complex press dies with the position of the workpiece in relation to the press centre and the angular orientation varying from press to press are common. Complicated loading motions are often necessary, with the workpiece having to be turned by up to 90 degrees against the direction of transfer. Handling the workpiece within the press can be further impeded by lateral dies on the lower press tool and by guide elements. The panels are often heavily oiled, which makes it difficult for them to be handled with suction mechanisms.

A typical automobile sheet-metal press shop may have an output of 3,400 – 3,650 parts/shift (press stroke 15 – 21 strokes/min) at 100% efficiency of the whole line depending on the type of part.

Processing Operations

Welding Welding is one of the first applications where robots were used extensively. Robots are used for spot welding as well as arc welding.

Spot welding is the most widely used application, particularly in the automobile body assembly lines. It is used to weld the frames and body panels together. Traditionally, human operators use the portable spot welders to make the spots at a number of locations. The operator needs to manipulate the position of the welding unit with all its associated cables, which are suspended from an overhead hoist system. In view of the high speed of operation required, the quality and consistency of the weld is often a problem. It is also a hazardous operation. The heavy spot-welding gun is attached to the end effector and the robot is programmed to perform a sequence of welds on the body as it is presented to the robot. Line-tracking control is generally used for the robots used in assembly lines.

Because of the significant hazard posed by the arc, *arc welding* is the process well suited for robotisation (Fig. 21.36). However, there are a large number of hurdles to be overcome before a robotic arc-welding operation can be designed. Arc welding is generally performed in confined areas, such as the inside of the tanks and ship hulls, which are difficult to access for robots. The other problem is the variability of the components to be welded, which is often difficult to control. The variability is in terms of dimensions as well as the type of edges and surfaces presented to welding. This causes a variation in gap between the parts, which have to be compensated by the welder. As a result, an arc-welding robot will have to be provided with sensors to monitor these variations and the controller will have to compensate for that.

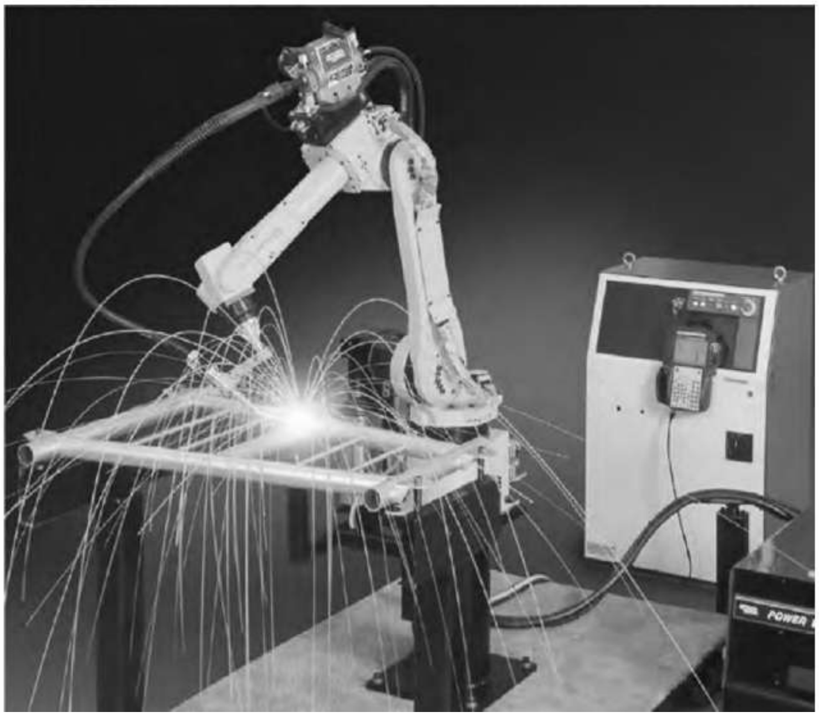


Fig. 21.36 *Arc-welding robot in operation (Courtesy, Fanuc)*

Arc-seam sensing is probably the most widely used seam-sensing technique today. It is reliable, relatively simple, and is totally integrated in the electric circuitry of the controller. It causes no obstruction and is not damaged by the spatter, fumes and the heat of the welding process. The only drawback is its limited range of application. Welds that do not have clearly defined surfaces such as square butt welds, thin sheet lap welds, etc., cannot be tracked with this means.

Contact sensing with the gas nozzle or the welding wire, is a reliable alternative for certain weld profiles. Often this can be applied in conjunction with arc seam tracking. Sensing is done prior to welding. When sensing with the welding wire, it is necessary that the wire is always cut, before sensing.

Inductive sensors are also low-cost solutions for many weldments. These sensors cannot sense while the robot is welding, i.e., sensing time has to be added to the weld cycle. The sensor, when installed near the torch, can limit access of the torch to welds in obstructed and narrow environments. Distortions that may occur during welding are not noticed.

Laser sensing is the most versatile sensing method available today. It monitors virtually any weld profile in real time so that the robot can immediately react to any changes that may occur. Even changes in weld preparation and air gaps can be noticed and corrective action can be taken automatically as welding takes place. Laser sensors are still quite bulky and may limit accessibility of the welding torch at times. The sensing optics are sensitive and prone to suffer in the rather hostile environment near the torch.

The justification of an arc-welding operation is generally based on the arc on time, which is the time the actual arc is taking place. Typical values for manual operations are about 20 to 30%. This can be increased to about 80% for a robot arc-welding operation.

Spray Painting Spray painting utilises atomised paint particles to be directed against the surface that needs to be painted. Human operators direct the spray gun at the object so as to cover the desired surface. A large amount of paint gets dispersed into the air, which is a health hazard. This makes it a fit case for robotisation. Robotised spray painting provides a consistent paint quality and is not affected by the noise from the nozzle, and fumes and mist in the air. Some of the applications are painting car bodies and other parts, appliances, bathroom fixtures, etc.

The main advantages of a robotic spray-painting system are consistency in the painted surface produced, reduced paint material used, greater productivity, lower consumption and fewer rejects. The main benefit achieved is removing a human operator from the paint booth, which makes the ventilation requirements a little less stringent than when human operators are involved.

The robots that are used for spray-painting applications are the articulated type that have continuous path control with manual lead through programming. Generally, an expert painter will teach the robot with lead through programming which is then used for continuous painting of the parts. The spray gun is the robot's end effector and also has a number of variables to be controlled. Some of the variables that need to be controlled to get a consistent paint quality are paint-flow rate, air pressure, and atomisation. It is also necessary to clean the gun periodically. It is possible to include a cleaning cycle in between the painting operations without losing much of productive time.

Automated Spray-Painting Line Economics

Example 21.5 A firm has a manual spray line employing 14 skilled human spray painters. The manual system produces 90 units per hour with an operation of 6 hours per shift. The two hours of lost production takes care of breaks, line servicing, maintenance, and clean-up. The total cost of each human spray painter, including salary, benefits, and overhead, is Rs 300 per hour. Additional maintenance costs of the manual system amount to Rs 24 000 per month.

A replacement robotic spray line costs Rs 45 million consisting of 10 robots with a production rate of 150 units per hour and works a total of 8 hours per shift. The robotic spray line costs the firm a total of Rs 15 million towards the depreciation, amortising the capital investment and salvage values. The robots cost Rs 1500 per hour of system operation which include service, repair and supervision.

Compare the two systems for their costs.

Solution Cost analysis—Manual System

Assuming a single-shift operation,

$$\text{Labour cost of 14 painters} = \frac{14 \times 300 \times 8}{6} = \text{Rs } 5600\text{ /hour}$$

To calculate the maintenance cost per hour, assume that there are 50 working weeks in a year, and 5 working days per week.

$$\text{Maintenance cost} = \frac{24\,000 \times 12}{50 \times 5 \times 6} = \text{Rs } 192\text{/hour}$$

$$\text{Hourly production cost} = \text{Rs } 5600 + \text{Rs } 192 = \text{Rs } 5792\text{/hour}$$

The manual spray system produces 90 units /hour.

$$\text{Cost per unit} = \frac{5792}{90} = \text{Rs } 64.36 \text{ per unit}$$

Cost analysis—Robotic System

Assuming a total of 2000 working hours per year,

$$\text{Robot cost} = \frac{15\,000\,000}{2000} = \text{Rs } 7\,500\text{/hour}$$

$$\text{Maintenance cost} = \text{Rs } 1500\text{/hour}$$

$$\text{Hourly production cost} = \text{Rs } 7\,500 + \text{Rs } 1500 = \text{Rs } 9000\text{/hour}$$

The robotic spray system produces 150 units/hour

$$\text{Cost per unit} = \frac{9000}{150} = \text{Rs } 60 \text{ per unit}$$

The economic benefits of going for a robotic spray system are marginal, and it may not be possible to justify such a huge investment based on such a small improvement. However, robots can work for two or three shifts per day, thereby decreasing the cost per unit without incurring any additional cost.

Example 21.6 Recalculate the cost of the robot spray line assuming that they operate two shifts per day.

Solution Assuming a total of 4000 working hours per year

$$\text{Robot cost} = \frac{15\,000\,000}{4000} = \text{Rs } 3750\text{/hour}$$

$$\text{Maintenance cost} = \text{Rs } 1500\text{/hour}$$

$$\text{Hourly production cost} = \text{Rs } 3750 + \text{Rs } 1500 = \text{Rs } 5250\text{/hour}$$

The robotic spray system produces 150 units /hour.

$$\text{Cost per unit} = \frac{5250}{150} = \text{Rs } 35 \text{ per unit}$$

This is a very favourable cost for adopting the robotic spray line. However, one point to be noted is that by working two shifts a day, the production rate is doubled, and the firm will have to examine whether there is a demand to satisfy the same. If a robotics or other automation application is not economically attractive on a one-shift basis, it will likely be more attractive on a two-shift basis. Three shifts will likely be better than two, and three shifts seven days per week will likely be the best alternative, provided the demand exists for the type of production volume such an operation would represent. It may be possible to think of having a system with fewer robots, thereby decreasing the investment cost without losing the productivity benefits.

Annual capacity of the manual system = 90 × 6 × 5 × 50 = 135 000 units/year

Annual capacity of robotic system = 150 × 2000 = 300 000 units/year

Paint Spray-Line Analysis Considering Potential Robot Failure

The above examples have considered that the system will not fail. However, in the real world the systems, are likely to fail, and therefore the economic calculations will have to take those failures into account.

Example 21.7 Let us assume that in the above example, the robots are likely to fail with a Mean Time Between Failure (MTBF) for each of these robots to be approximately 2000 hours. Whenever a failure occurs, there will be a loss of production of four hours and other expenses will be Rs 150 000 per occurrence. Recalculate the production cost per unit, allowing for the failures.

Solution Since there are ten robots with each having an MTBF of 2000 hours, the system MTBF would be one-tenth that amount or 2000 h/10 = 200 hours.

When any robot fails, there are two costs to be considered:

1. loss of system production time, and
2. additional material and damage cost.

The loss of the production time due to failure = 4 hours

$$\text{Availability of the system} = \frac{200}{200 + 4} = 0.98 \text{ or } 98\%$$

Actual production time of a robot = 0.98 × 2000 = 1960 hours

$$\text{Robot cost with failure} = \frac{15\,000\,000}{1960} = \text{Rs } 7653.06/\text{hour}$$

$$\text{Maintenance cost} = \text{Rs } 1500\ /\text{hour}$$

$$\text{Hourly production cost} = \text{Rs } 7653.06 + \text{Rs } 1500 = \text{Rs } 9153.06/\text{hour}$$

To calculate the actual number of failures in a given period, we have to consider the MTBF of the system and the actual production period.

$$\text{System failures/year} = \frac{1960}{200} = 9.8 \text{ failures/year}$$

$$\text{Failure cost} = \frac{150\,000}{200} = \text{Rs } 750/\text{hour}$$

$$\text{Hourly production cost} = \text{Rs } 7653.06 + \text{Rs } 1500 + \text{Rs } 750 = \text{Rs } 9903.06\ /\text{hour}$$

The robotic spray system produces 150 units/hour.

$$\text{Cost per unit} = \frac{9903.06}{150} = \text{Rs } 66.02 \text{ per unit}$$

The effect of failure is to increase the cost of production.

Other Processing Operations

A number of processing operations have been tried using robots. They are

- Drilling and other machining operations
- Grinding, polishing and deburring operations
- Water jet cutting
- Laser drilling and cutting

Requirements for each of these operations vary depending upon the actual application. An industrial robot with a portable drill as its end effector has the dexterity of a human operator with a handheld drill. It can position its work platform in an infinite number of planes, thus catering to more complex situations. This positioning capability is essential in such tasks as drilling rivet holes in aircraft skins, the surface of which may be curved and, therefore, does not have consistent plane.

The human operator with a handheld drill or an industrial robot can adjust the plane of the tool platform such that the direction of the feed is normal to the surface of the aircraft skin. In such an operation, the robot can be positioned more precisely than a human-held drill, and thus quality is enhanced even when using a template. Safety, quality, and economics are the benefits to be gained from robotic drilling operations.

As an example, one robot drills 550 holes in the vertical tail fins of F-16 fighters. This application not only saves worker hours but also eliminates a hazard by preventing worker exposure to composite dusts created by the drilling operation.

Assembly Operations The most challenging arena for robot application is robotic assembly. Assembly requires precision, repeatability, variety of motion, sophisticated gripper devices, and sometimes compound gripper mechanisms in which the gripper plays an active and primary role in one of the assembly steps, besides simply holding the piece parts.

The robotic assembly operations should be fast and efficient. For the most part, assembly operations are simply not dangerous, dirty, or hot. Therefore, the assembly robot must be fast and efficient to compete with human assembly operators without consideration for the robot's special capabilities for working in inhospitable environments.

21.4 AUTOMATED STORAGE AND RETRIEVAL SYSTEM

21.4.1 Introduction

In large manufacturing establishments, the volume of items to be stored and retrieved becomes so large that the manual means of doing so becomes extremely unreliable and time consuming. The large volumes also call for proper information management procedures in order to reduce the duplication and reduction of inventory costs.

Automated Storage and Retrieval System (AS/RS) plays a central role in the automated factory. The AS/RS not only controls inventories, but also keep track of parts and materials in process or transit. In other words, the AS/RS has the ability to know where everything is, on a real-time basis, even as the material moves within the factory. The importance of it for management to make manufacturing decisions based on accurate real-time information, can be understood from the fact that in-process materials rarely spend more than 5% of their time being worked on, and that the remaining 95% is spent in transit or storage. Thus, it becomes easier to visualise the role of materials handling and storage system within an automated factory.

An AS/RS, sometimes also referred to as *automated warehouse*, is a combination of equipment and controls, which handles the stores and retrieves material faster and with greater safety and efficiency than conventional material handling and storage systems. It contains several rows of storage racks and storage and retrieval devices (stacker cranes). The system can be linked to other external devices such as conveyors or Automated Guided Vehicles (AGV) for transferring material (in trays or pallets) to the shop floor or palletising stations. A typical AS/RS is shown in Fig. 21.37.

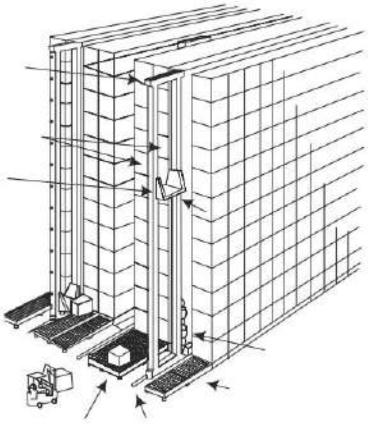


Fig. 21.37 *Typical AS/RS with its component parts*

The incoming items are first sorted and assigned to pallets. The pallet loads are then routed through weighing and sizing stations to ensure that they are within the load and size limits. The accepted ones are transported to the Pick-up and Delivery (P and D) stations, with the details of the pallet contents communicated to the central computer. This computer assigns the pallet a storage location in the rack and stores the location in its memory. The pallet is moved from P and D station to the storage by Storage and Retrieval machines (S/R machines), or stacker crane. Upon receipt of a request for an item, the computer searches its memory for the storage location and directs the stacker crane to retrieve the pallet. The supporting transportation transports the pallets from the P and D station to its final destination.

The major components of an AS/RS are

(i) Storage and retrieval machine (shuttle crane or stacker crane),
(ii) Storage structures,
(iii) Transport devices (AGV, Conveyor, etc.), and
(iv) Controls

(i) Storage and Retrieval Machine (S/R machine)

The S/R machine is characterised by its ability to operate accurately and safely at high speeds, reach heights of 30 m or beyond, and operate in aisles only a few centimetres wider than the load it carries (Fig. 21.38). The S/R machine can be fully automated and therefore easily controlled by a computer.

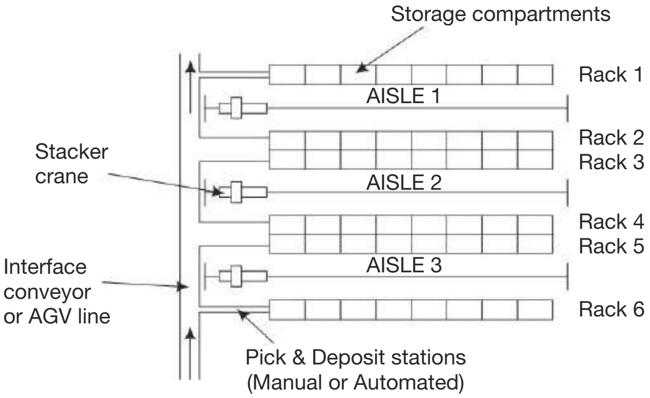


Fig. 21.38 *Physical arrangement of the aisle and storage structures in an automatic storage and retrieval system*

The modern S/R machine runs on a floor-mounted rail and is guided at the top. It comes in a wide variety of sizes and configurations, because its design is a function of the load it carries and the task it performs. Speed is generally determined by the system's throughput requirements (i.e., how many loads need to be stored and retrieved in an hour). Travelling speeds sometimes exceed 150 m/min while hoisting speeds can be as high as 50 m/min.

(ii) Storage Structure The storage structure (racking) is a critical part of the AS/RS. It differs considerably from conventional pallet racking in that AS/RS storage racks are normally much higher and interface directly with the S/R machines, thus making manufacturing and installation tolerances more critical. AS/RS rack design must provide for integration with S/R machine guide rails.

The most common storage structures are free-standing and installed inside a building. Specifications differ, depending on the type of load to be stored and system configuration. Today an increasing number of systems are rack-supported, that is, the rack storage structure supports the building itself. This type of system can be over 30 metres tall and is popular because it reduces construction time and cost.

(iii) Transport Devices The transport (system) device moves the loads beyond the limit of the S/R machine. Some systems need only a conveyor, while larger and more complex installations require elaborate transport devices connecting the S/R with other factory operations. There are many types of transport devices which can be used with AS/RS fork lifts, roller or chain conveyors, overhead power and free conveyors, in-floor tow-lines, shuttle trolleys and automated guided vehicles. The choice depends on the throughput requirements, type of load to be handled, and degree of interaction with shipping, receiving, manufacturing, assembly and other plant operations. An example system that links with a conveyor and AGV is shown in Fig. 21.39.

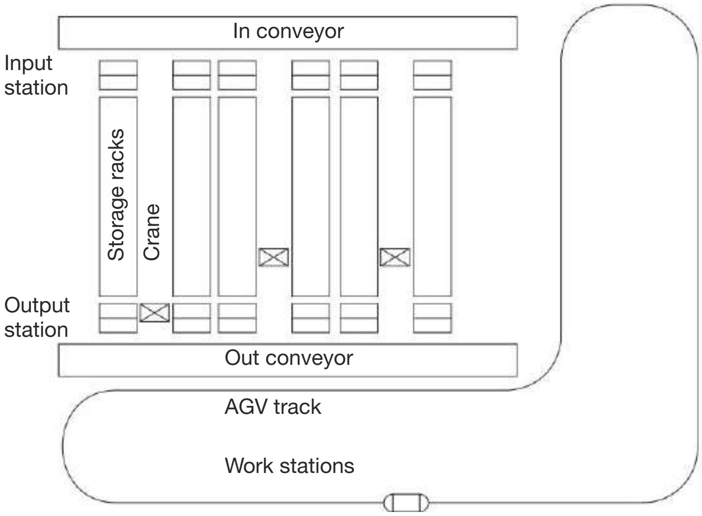


Fig. 21.39 *Physical layout of automatic storage and retrieval system linked with AGV for further movement*

(iv) System Controls The system controls encompass two functions, the control of equipment and the control of data. This computer control system may also perform tasks like inventory control, data automation, networking control and is frequently linked to an even larger corporate-management information system computer.

Advantages of AS/RS Absolute inventory accountability is one of the most important benefits of an AS/RS. By knowing exactly how much of what materials are located at any given moment, inventory reductions are quite common and higher reductions are frequently realised. In addition, these systems discourage pilferage and generate very little product or equipment damage. The systems can also improve customer service and working conditions, and thus result in less indirect cost associated with supervision, administration, facilities and security.

The following are the benefits of using an AS/RS:

- Better space utilisation
- Less direct and indirect labour
- Reduced inventories
- Less energy consumption
- Reduced pilferage
- Less product damage
- Improved working conditions
- Easier housekeeping
- Less equipment damage
- Improved customer service
- Better management control

21.4. 2 AS/RS Design Process

The design of an AS/RS is a well-planned process. The development of the whole AS/RS can be divided into two parts:

1. Physical system design
2. System control policies

The physical system design includes the determination of

- Storage dimensions
- Stacker crane selection
- Number of storage locations

On the other hand, the system control policies part is used for

- Stacker-mode determination
- Storage-location assignment
- Stacker-crane sequencing and scheduling

The maximal benefits of an AS/RS depend on the physical system design and system control policies.

The physical system of an AS/RS is composed of storage locations, stacker cranes and conveyors, as shown in Fig. 21.38. The task of transporting the pallets/trays to and from the production floor is to be carried generally by the AGV. At the pallet loading station, pallets are loaded with the job located in fixture. Loaded pallets from the pallet station are put onto the conveyor. From here, these pallets are then received by respective storage input stations. Once these pallets are transferred to the input station, the S/R machine then arrives and picks the pallet by its shuttle fork platen and stores onto a rack in the storage system. Upon receipt of a request for a pallet, the S/R machine travels to the required location and delivers the pallet to the storage output station for transferring to the production shop. From the output station, the pallet is transferred to the conveyor. This conveyor carries it to the AGV loading station. The AGV then takes this pallet to the respective machine in buffer. The reverse sequence takes place for finished components from the machine.

Some of the design variables that describe the physical configuration of the system and the dynamics, which govern the movement of the pallets and trays, are shown in Table 21.7 with reference to Fig. 21.40.

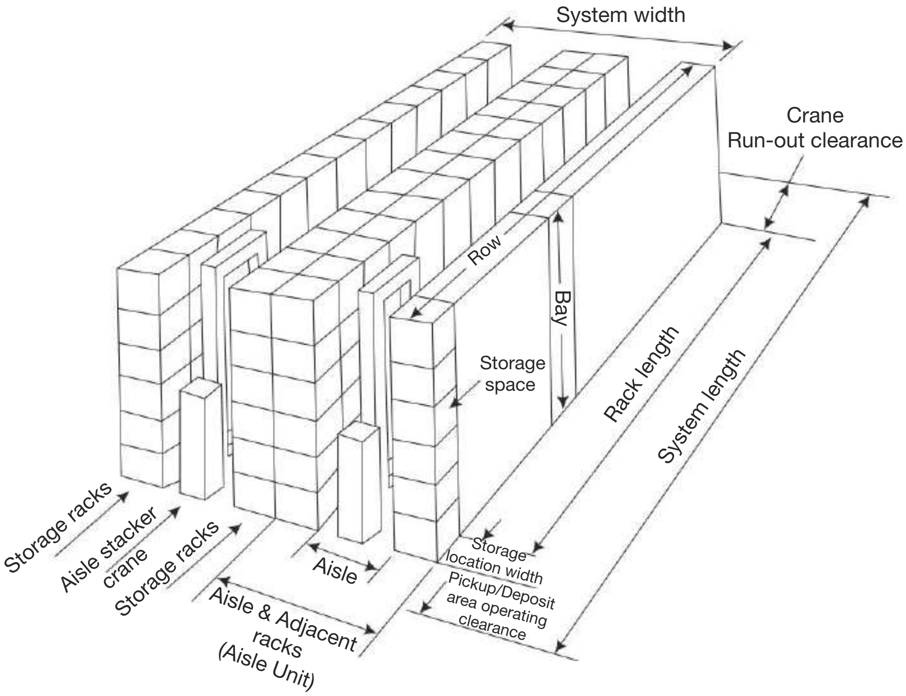


Fig. 21.40 *Definition of various terms related to ASRS*

The actual design of an AS/RS needs to be followed in steps [Maleki. A., 1991] as shown below:

1. Determine load, sizes, configuration and weights.
2. Determine how many storage spaces will be required.
3. Determine system throughput.
4. Determine number of cranes required.
5. Determine number of rows in a system.
6. Determine system height and loads height.
7. Determine the number of bays required per row.
8. Determine rack length, system length, and system width.

Determine Load, Sizes, Configuration and Weights Material to be handled in a given AS/RS comes in a number of sizes and varieties. They may be parts, tools, fixtures, pallets and other supplies. However, it is necessary to standardise the unit loads such that all the parameters of the AS/RS could be designed around that. For this purpose, palletising of the parts needs to be done. Based on the unit load size and weight, the rest of the parameters of the storage spaces are decided. Also, the design of the structure of the rack system has to take into account the loads present in the various racks.

Table 21.7 *System variables*

- *Storage rack dimensions*
  - Length of different pallets
  - Width of different pallets
  - Vertical travel of saddle/tool head
  - Maximum weight on a single rack
- *Storage dimensions*
  - Building length
  - Building height
- *Stacker crane specifications*
  - Horizontal speed
  - Horizontal acceleration
  - Horizontal deceleration
  - Vertical speed
  - Vertical acceleration
  - Vertical deceleration
  - Shuttle speed
  - Shuttle acceleration
  - Shuttle deceleration
- *Number of storage locations and stacker cranes*
- *Input and output stations*
  - Number of pallet on each station
  - Width of each station
  - Height of each station
- *Conveyor variables*
  - Conveyor length
  - Conveyor width
  - Conveyor speed
- *Clearance values*
  - Between rack plane and pallet
  - Between rack top and pallet
  - Between rack side and pallet
  - Between stacker crane and storage face
  - Between storage back to back
- *Storage structure variables*
  - Width of horizontal support
  - Width of vertical support
- *Pallet requirement*
  - Number of pallets required per hour
  - Number of hour for which pallet are required
- *Peak pallet requirement*
  - Minimum time between two request for a pallet/tray from the same storage
- *Pallet separation*
  - Gap between two pallet/tray on conveyor, and input/output stations
- *Pallet stations*
  - Number of AGV loading stations
  - Number of AGV unloading stations
  - Number of pallet loading stations around conveyor

Height of the storage unit = Height of unit load + Height clearance

Length of the storage unit = Length of unit load + Length clearance

Width of the storage unit = (Width of unit load + Width clearance) × u

where u = Number of loads in a single location (Maximum of 3)

Determine How Many Storage Spaces will be Required It is necessary to know the number of storage spaces required in the system. This will have to take into account expected storage volume and the type of storage policies to be adopted. In dedicated storage policy, the number of storages provided will be equal to the sum of the individual maximum inventory levels. Thus, if a particular item is out-of-stock then its corresponding storage-keeping unit will be empty. It would require more storage space. In the randomised storage policy, the number of storage spaces required will be less and individual storage-keeping units will be assigned multiple storage slots.

Determine System Throughput This specifies the number of loads that will be in and out of the storage units. It is not the average, but the maximum units so that all the other necessary facilities can be appropriated properly.

Determine Number of Cranes Required The number of cranes (retrieval machines) required depend directly upon the system throughput and the handling capacity of the crane.

$$\text{Number of cranes} = \frac{\text{System throughput}}{\text{Crane Capacity}}$$

Crane capacity in terms of the number of load cycles per hour depends upon [N. Singh 1996]

- Speed of the crane
- Mix of single and dual-cycle transactions
- Arrangement of the stored items
- AS/RS control system speed
- Speed and efficiency of other material-handling equipment that moves loads to and from AS/RS.

Determine Number of Rows in a System This depends upon the design policy. Normally, one crane is assigned for one aisle in the storage structure. Sometimes it is possible that one crane can attend to two aisles. For one crane serving one aisle, the number of rows can be given as

$$\text{Number of rows} = \text{Number of cranes} \times 2$$

Determine System Height and Loads Height These are the physical parameters of the system and need to be decided. Typically, the system height varies from 10 to 30 metres, with the most efficient systems having heights between 12.5 to 22.5 metres.

$$\text{Number of loads per height} = \frac{\text{Desired System Height}}{\text{load Height}}$$

Determine the Number of Bays Required per Row

$$\text{Number of bays} = \frac{\text{Number of Storage Units}}{\text{Number of Rows} \times \text{Number of loads per Height}}$$

Determine Rack Length, System Length and System Width

$$\begin{aligned}
\text{Bay width} &= \text{Load width} + \text{Required clearance} + \text{Rack support width (Centre to centre)}\\
\text{Rack length} &= \text{Bay width} \times \text{Number of bays}\\
\text{System length} &= \text{Rack length} + \text{Crane run-out clearance} + \text{Pickup/Deposit area}\\
\text{Aisle unit} &= \text{Aisle width} + (2 \times \text{Bay depth})\\
\text{System width} &= \text{Aisle unit} \times \text{Desired number of aisles}
\end{aligned}$$

Example 21.8 Design the AS/RS system with the following requirements. The unit load sizes are 1.2 m (width) × 1.3 m (length) × 1.3 m (height). The height clearance required is 0.25 m, the length clearance is 0.20 m, the width clearance is 0.15 m and the number of unit loads per storage unit (u) is 2. The average cycle time of the operation of the S/T machine is 1 minute. The system has a total number of storage spaces equal to 12 000. The system throughput expected is 480 operations (either storage or retrieval) per hour. Take the desired system height to be less than 24 m. Take centre-to-centre rack support as 0.15 m, bayside support allowance as 0.15 m, clearance for the crane run-out as 3 m, clearance for the pick-up/deposit area as 5 m, and aisle width as 2 m.

Solution Height of the storage unit = 1.30 + 0.25 = 1.55 m

Length of the storage unit = 1.30 + 0.20 = 1.50 m

Width of the storage unit = (1.2 + 0.15) × 2 = 2.7 m

Since cycle time is 1 minute, the number of operations that can be completed by a crane per hour will be 60.

$$\text{Number of cranes} = \frac{480}{60} = 8$$

$$\text{Number of rows in the system} = 2 \times 8 = 16$$

$$\text{Number of loads per height} = \frac{24}{1.55} = 15.48 = 15$$

$$\text{Number of bays in each row} = \frac{12\,000}{16 \times 15} = 50$$

Bay width = 2.7 + 0.15 = 2.85 m

Rack Length = Bay width × Number of bays = 2.85 × 50 = 142.5 m

System length = Rack length + Crane run-out clearance + Pickup/Deposit area

= 142.5 + 3 + 5 = 150.5 m

Bay depth = Width of storage unit + clearance = 2.7 + 0.15 = 2.85 m

Aisle unit = Aisle width + (2 × Bay depth)

= 2 + 2 × 2.85 = 7.7 m

System width = Aisle unit × Desired number of aisles

= 7.7 × 8 = 61.6 m

Summary

- Automated material handling is a very important component of computer aided manufacturing operations. A number of devices are available for this purpose. This chapter discusses about the most important among them—Automated Guided Vehicles (AGV), Robots and Automated Storage and Retrieval Systems (AS/RS).
- AGVs are material-handling systems that are driverless and run on fixed routes most of the time.
- There are different types of AGVs that are used for different applications, viz., AGVS towing vehicles, AGVS unit load vehicles, AGVS pallet trucks, AGVS fork trucks, light load vehicles and AGVS assembly line vehicles.
- Wire guidance is most commonly used for AGVs, though other types of guidance such as infrared and laser are sometimes used.
- System management of AGV is important and will have to be considered at the time of its design or selection.
- AGV system design involves the design of system layout, identifying the number of vehicles required, and deciding on the operation and control policies.
- The main advantages of AGVs are the reliable control over material transportation, better resource utilisation, and increased throughput.
- Robots are being used practically in all types of manufacturing operations.

- Robots are classified based on the type of joints between the various moving members, such as Cartesian, polar, cylindrical or articulated.
- The robots generally have the four basic components—manipulator, controller, power source and the end effector.
- Robot motion can be from simple point-to-point to the more sophisticated line tracking depending upon the requirements.
- Sensors in robots add intelligence to the operation. The types of sensors used in robots are proximity sensors, range sensors, force sensors, etc.
- Grippers need to be designed for specific applications, and can use simple mechanical movement to operate or special drives such as pneumatic, hydraulic and electrical drives for more sophisticated applications.
- Robots can be programmed using lead by nose for simple applications, or by using offline robot programming languages such as VAL II for complex applications.
- It is necessary to consider the various costs involved to justify the introduction of robots for a given application.
- Robots can be used for various material-handling operations, different processing applications such as drilling and grinding, and for assembly operations.
- It is necessary to consider all the operations to evaluate the manufacturing cell using a robot.
- AS/RS employs stacker cranes and high-rise storage structures to systematically store and retrieve materials under computer control.
- AS/RS helps in reducing inventories, better space utilisation and better management control of inventory.
- AS/RS design involves the physical design, which involves the selection of bin and storage structure dimensions, to the system and operation control which decides on the way the items are stored and retrieved.

Questions

1. Discuss the principles of material handling as enunciated by CICMHE.
2. Describe the importance of a material-handling system.
3. What are the different types of material-handling equipment used in manufacturing industries? Give a brief description of each one of them.
4. What is an automated guided vehicle?
5. What are the various types of AGVS that are used in manufacturing automation?
6. Give a brief description of AGVS unit load vehicles in relation to manufacturing industries.
7. Give a brief description of AGVS assembly line vehicles in relation to manufacturing industries.
8. Describe the advantages to be gained by the use of an automated guided vehicle in manufacturing shop.
9. What are the various guidance methods available for an automated guided vehicle?
10. Briefly describe about the wire guidance used in automated-guided-vehicle movement.
11. What are the parameters to be considered for designing the number of AGVs in a system?
12. Describe the traffic control patterns used in AGVS traffic management.
13. Calculate the number of AGVs required with a vehicle speed of 50 m/min with an average loaded travel distance of 180 m. The average empty travel

distance is 120 m. The total time required for loading and unloading is 1 minute. The number of deliveries to be made are 50 per hour. Assume a traffic factor of 0.85. (Ans. 7)

14. Define a robot.
15. Explain the applications for which robots are generally used.
16. What are the reasons for the successful application of robots in manufacturing industries?
17. Briefly explain the classification of robots.
18. Describe briefly the various components in a robot.
19. What are the various types of motion control possible in robots?
20. Describe the most common type of motion control that is available in commercial robots.
21. What are the applications for which line-tracking type of motion control is used in robots?
22. Describe the requirements of sensors in robots.
23. What are the characteristics to be considered for the end-of-arm tooling used in robots?
24. Describe the type of power sources used in the manipulation of grippers for robots.
25. Explain how robots with double grippers increase the utilisation of machine tools.
26. What do you understand by the word 'compliance' in relation to a robot gripper? Give an example of the method used to achieve this.
27. What are the methods used for programming a robot?
28. Write a VAL II program for a robot to palletise a rectangular part that is 250 mm × 150 mm. It has to arrange the part in 3 uniform rows of 4 parts each. The pallet size is 1 m × 0.65 m. The minimum clearance between the parts and edges should be at least 50 mm.
29. An industrial robot loads and unloads two machine tools in sequential order. The robot picks the part from a tray A and loads it into the machine B. After the machining is completed, it unloads the part from the machine B and moves to the machine C where it loads that part. After the machining is completed on the machine C, the robot moves the part to the output tray D. Write a VAL II program for this sequence of operations.
30. In the above example the machine A is idle when the machine C is working and vice versa. What is to be done to improve the situation? A signal needs to be generated to signify the end of operations by using sensors. Develop a more efficient robot program using sensors to improve the machine utilisation.
31. How do you justify the acquisition of a robot for a given application based on economics?
32. Discuss the various material-handling applications for which the robots are used.
33. What are the problems associated with the use of robots for material handling in forging press application?
34. Give the type of layouts that could be used with a robot as the main material-handling unit.
35. Four robots are used for loading and unloading with four machining centres. Each machining centre is doing the same operation in parallel. The individual machine cycle time is 48 seconds that includes the loading and unloading time. What is the ideal production capacity of the system?
36. In the above example, if the machines are producing the part sequentially what will be the ideal production capacity of the system?
37. In the above example, each robot has an MTBF of 1000 hours, and assume that the entire line stops for a period of 12 hours till the problem is rectified. Calculate the production capacity with robot failure and sequential production line.
38. A single robot serves two machine tools as shown in Fig. 21.35. Assume that the processing time on Machine B is 3 minutes while that of the Machine C is 5 minutes. The robot-move times are as in Table 21.4. Calculate the production rate.
39. Write a short note on the application of robots in welding (spot welding and arc welding).
40. Define automated storage and retrieval system.
41. Explain the operation of an automated storage and retrieval system.
42. What are the major components present in an automated storage and retrieval system?
43. Briefly explain the advantages to be gained by the use of an automated storage and retrieval system.
44. Which are the factors to be considered in the design of an automated storage and retrieval system?
45. What are the various types of material handling and/or transportation systems used in FMS?

Describe their individual domains (areas) of applications for which they are used with their relative advantages and disadvantages, if any.

46. Write short notes on the automatic storage and retrieval systems and their application areas in FMS.
47. Design the AS/RS system with the following requirements. The unit load sizes are 0.6 m (width) × 0.5 m (length) × 0.5 m (height). The height clearance required is 0.20 m, the length clearance is 0.15 m, the width clearance is 0.10 m and the number of unit loads per storage unit (*u*) is 2. The average cycle time of the operation of the S/T machine is 1 minute. The system has a total number of storage spaces equal to 10 000. The system throughput expected is 420 operations (either storage or retrieval) per hour. Take the desired system height to be less than 20 m. Take centre-to-centre rack support as 0.10 m, bayside support allowance as 0.10 m, clearance for the crane run-out as 3 m, clearance for the pick-up/deposit area as 5 m, and aisle width as 2 m.

22

FLEXIBLE MANUFACTURING SYSTEMS

Objectives

Current-day manufacturing is facing a number of challenges, which require a different type ‘of solution than can be done with conventional manufacturing methods. Many of the new design of products are characterised by

- High quality (accuracy and finish)
- Large variety
- Small volume
- Less lead time
- Varying designs
- Low cost
- Low product life

This calls for flexibility in the manufacturing operations and elimination waste in all forms. Flexible Manufacturing Systems (FMS) conveniently fit the bill for such difficult manufacturing situations. After completing the study of this chapter the reader should be able to

- Understand the definition and concept of operation of FMS
- Learn about the different types of flexibilities and their relevance to manufacturing
- Understand the type of equipment used in FMS
- Appreciate the importance of tooling and various types of tool management systems that are employed in FMS
- Learn about the types of FMS layouts that are used
- Understand the FMS control requirements
- Develop methods for designing an FMS for any given part spectrum
- Look closely at a case study to understand the process involved in developing an FMS

22.1 INTRODUCTION TO FMS

The use of CNC machine tools provides flexibility in terms of the low job changing time. However the full benefits of automation cannot be achieved simply by the use of the CNC machine tool alone. The complete job making process involves the use of machine tool along with all the associated equipment being made available at the right time. The associated equipment involves the cutting tools, workpiece blank, part program, tool offsets and the like. As a result the effective CNC machine utilisation can be achieved if all these are integrated. Some typical figures for machine utilisation based on general trend in the industries are given in Table 22.1.

Table 22.1 *Automation benefits*

	Automation		*Machine utilisation*
1.	Basic CNC	Manual tool and work loading	40%
2.	Basic CNC with	automatic work holding and workpiece storage, manual loading	60%
3.	Complete machine automation	automatic work and tool handling, Tool monitoring, Workpiece inspection, work and tool storage	75%
4.	Integration of group of machines similar to that shown in 3		80%
5.	Flexible Manufacturing System	Automated workpiece movement between machines	90%

Thus it can be seen that the full utilisation (90% with the rest 10% allocated for maintenance) can be achieved in FMS by properly integrating all the required functions. A large number of definitions have been provided in the literature for the Flexible Manufacturing svstems (FMS) as follows:

> *A series of automatic machine tools or items of fabrication equipment linked together with an automatic material handling system, a common hierarchical digital pre programmed computer control, and provision for random fabrication of parts or assemblies that fall within predetermined families.*
>
> *A FMS is a group of NC machine tools that can randomly process a group of parts, having automated material handling and central computer control to dynamically balance resource utilisation so that the system can adopt automatically to changes in part production, mixes and levels of output.*
>
> *FMS is a randomly loaded automated system based on group technology manufacturing linking integrated computer control and a group of machines to automatically produce and handle (move) parts for continuous serial processing.*
>
> *FMS combines microelectronics and mechanical engineering to bring the economies of scale to batch work. A central online computer controls the machine tools, other workstations, and the transfer of components and tooling. The computer also provides monitoring and information control. This combination of flexibility and overall control makes possible the production of a wide range of products in small numbers.*
>
> *A process under control to produce varieties of components or products within its stated capability and to predetermined schedule.*
>
> *A technology which will help achieve leaner factories with better response times, lower unit costs, and higher quality under an improved level of management and capital control.*

Thus it can be seen that a true FMS can handle a wide variety of dissimilar parts, producing them in small numbers even one at a time, in any order, as needed by making use of all the computer controlled equipment (workstations and material handling) with the help of a central computer control of all the equipment within. Typical costs associated with the various types of manufacturing systems are given in Table 22.2.

Table 22.2 *Typical costs associated with various manufacturing system*

Costs	*Small scale (stand alone machine tools)*	*Medium scale (FMS)*	*Large scale (Transfer lines)*
Direct labour	43.7	24.7	19.7
Overhead	13.5	13.9	23.7
Capital	17.8	33.1	29.8
Other costs	25.0	28.3	26.8
Total	**100.0**	**100.0**	**100.0**

There are a large number of benefits that can be derived from the use of FMS. They are:

1. Flexible Manufacturing Systems are regarded as one of the most efficient methods to employ in reducing or eliminating problems in manufacturing industries.
2. FMS brings flexibility and responsiveness to the manufacturing floor.
3. FMS enables manufacturers to machine a wide variety of workpieces on few machines with low staffing levels, productively, reliably and predictably.
4. A true FMS can handle a wide variety of different parts, producing them one at a time in random order.
5. Machine tools in many manufacturing industries are woefully under utilised due to equipment not being used on second and third shifts, a decreasing availability of skilled personnel, and day-to-day disturbances.
6. FMS shortens the manufacturing process through improved operational control, round-the-clock availability of automated equipment, increased machine utilisation and responsiveness, and reduction of human intervention.
7. Better competitive advantage
8. Lower work in process inventories
9. Reduced throughput time and its variability
10. Improved manufacturing control
11. Improved quality and reduced scrap rate
12. Reduction of floor space used
13. Better status monitor of machines, tools, and material handling devices
14. Improves the short run response time to the problems on the shop floor such as
    - Demand variations,
    - Design and process changes that can be easily adjusted by changing the CNC part program, which is generally developed by a CAD/CAM system as part of the design change,
    - Machine unavailability can be taken care of by the FMS control system which can automatically transfer the part to another machine that is available, and
    - Cutting tool failures can be detected by sensors and stop the machine thereby reducing the catastrophic failures. Then the control system can initiate steps to repair and replace the failed cutting tool.

15. Improve the long term cost-effectiveness of the system by supporting
    - Changing product volumes,
    - Allowing different part mixes, and
    - Allowing new parts to be added

Before we further proceed with the discussion on FMS, it may be necessary to examine the concept of flexibility. It is not a unique definition and it is possible to define a number of flexibilities related to manufacturing, which will be relevant to the manufacturing system. Some of them are

- Machine flexibility
- Production flexibility
- Mix flexibility
- Product flexibility
- Routing flexibility
- Volume flexibility
- Expansion flexibility

Machine Flexibility This defines the capability of machines to a wide range of production of operations and part styles that may require. This may be characterized by having a low setup or change over time, ease of machine reprogramming, sufficiently large tool storage capacity and the skill and versatility of machine operators.

Production Flexibility This aspect refers to the range of part styles that can be produced in the system. This is dependent on the machine flexibility and the range of machine flexibilities.

Mix Flexibility This is the ability with which the product mix in a given system can be changed. This depends to a great extent upon the similarity of parts in the mix, the relative work content times of parts produced and the machine flexibility.

Product Flexibility This is the ease with which changes in product designs can be accommodated in the system. This will depend on how closely the new part design matches the existing part family, the ability for off-line part program preparation and the machine flexibility.

Routing Flexibility This specifies the capacity to produce parts through alternative workstation sequences in response to equipment breakdowns, tool failures, and other interruptions. This is facilitated by the similarity of parts in the mix, similarity of workstations, duplication of workstations, cross-training of manual operators and the availability of common tooling.

Volume Flexibility This is the ability to economically produce parts in high and low total quantities of production depending upon the market demand. This will depend upon the level of manual labour performing production and amount invested in capital equipment.

Expansion Flexibility This is the ease with which the system can be expanded to increase total production quantities, should the need arise. This can be examined by the expense of adding workstations, ease with which layout can be expanded, type of part handling system used and the ease with which properly trained workers can be added.

The development of an FMS for a given application therefore will have to take the required flexibilities into account while designing the various facilities and controlling procedures.

22.2 FMS EQUIPMENT

The various components present in a FMS are shown in Fig. 22.1. Each of these will have further elements depending upon the requirement as shown below:

- Workstations
    - CNC Machine tools
    - Assembly equipment
    - Measuring equipment
    - Washing stations
- Material handling equipment
    - Load unload stations (palletising)
    - Robotics
    - AGV
    - AS/RS
- Tool systems
    - Tool setting stations
    - Tool transport systems
- Control system
    - Monitoring equipment
    - Networks

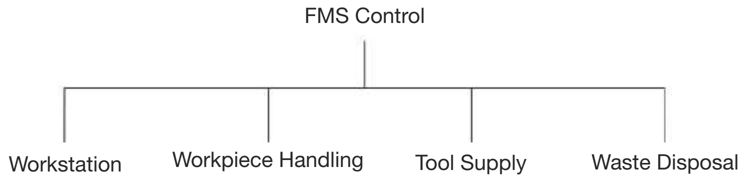


Fig. 22.1 *Typical component blocks of a flexible manufacturing system*

Typical arrangement of an FMS is shown in Fig. 22.2. It can be noticed that the FMS is shown with 4 machining centres, one wash station and one coordinate measuring machine as the workstations. Besides it has the load/unload stations, *AS/RS* for part and raw material storage, and wire guided AGV for transporting the parts between various elements of the FMS. There can be more than one AGV depending upon the transport requirement. It can also be noticed that buffers for parts queuing are also present.

One of the main requirements for a machine tool that can be integrated into FMS is that it has to be fully automatic. That means it has to have ATC as well as APC as shown in Fig. 22.3a. However, in such cases some arrangement has to be provided with pallets to be fed to the APC at regular intervals depending upon the machining cycle of the parts. There can be many arrangements that can be used with pallets. One simple arrangement can be a pallet carousal with a number of pallets on which the part blanks can be setup as shown in Fig. 22.3b. This machine thus will not require any attention till all the workpieces in the pallet carousal are completed. These types are used generally where the machining cycle times are large, such as those for internal combustion engine blocks. This can be used for standalone machines, which can be left unattended in the third shift. Such machines are called Flexible Machining Modules (FMM).

When larger number of pallets are required, then instead of a pallet carousal, a linear carousal which is basically a pallet stand becomes convenient as shown in Fig. 22.4. In the case of linear carousal, it becomes

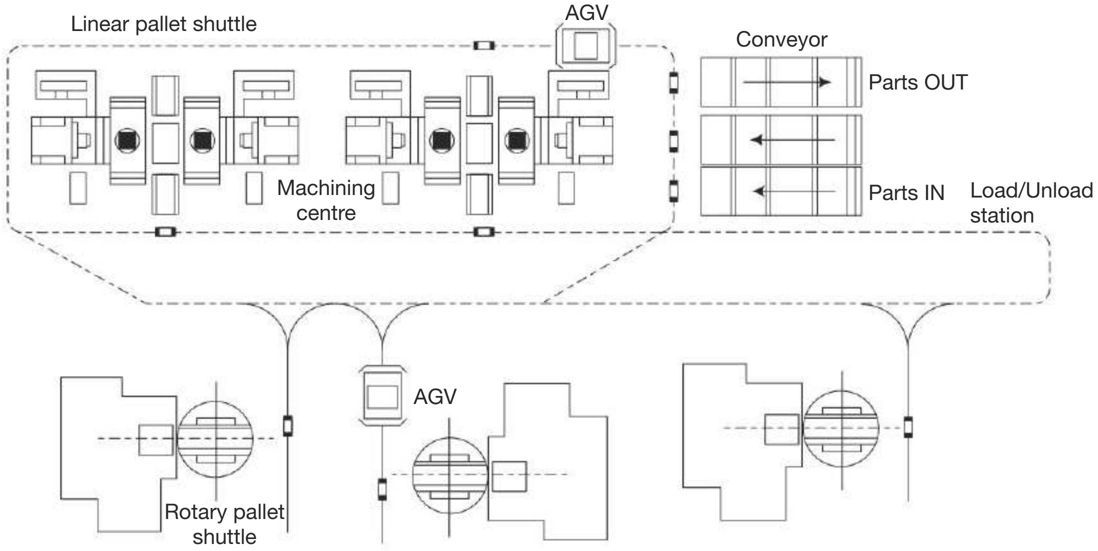
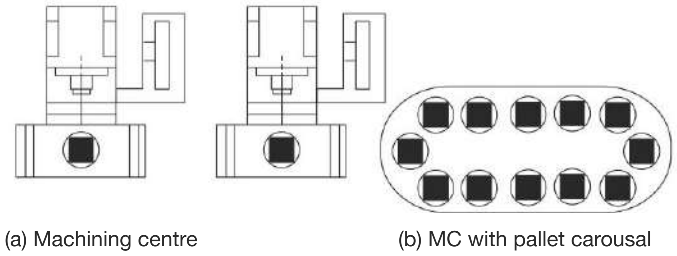


Fig. 22.2 *Typical flexible manufacturing system with all the components*


(a) Machining centre

(b) MC with pallet carousal

Fig. 22.3 *Machining centres for use in FMS*

necessary to provide a pallet transport vehicle to move the pallets from the stand to the machine and vice versa. Another advantage with such an arrangement is that when necessary it will be possible to extend the capacity by simply adding another machining centre as shown in Fig. 22.5. Such an arrangement with more than one machining centre with integrated material handling arrangement is called as flexible machining cell or FMC. In Fig. 22.5 the rail guided vehicle has the integrated automatic pallet changer with which the pallets will be moved between the machining centre and the pallet stand. An actual arrangement of Makino FMC is shown in Fig. 22.6. This being modular, two such modules can be connected together to form a larger machine pool with all associated software integration is the flexible manufacturing system or FMS.

It is possible to have similar arrangement with turning centres as well. The material handling arrangement most convenient with the turning centres is the robot. A typical flexible machining cell with two CNC turning centres is shown in Fig. 22.7.

When the robot has to service a larger number of machining units. It is necessary to keep the robot base on a mobile unit. In Fig. 22.8 the robot is servicing four turning centres, In this case the ROBOT is arranged on rails so that it can cater to all the machining units, Sometimes it is also possible to have the robots hanging from overhead frame to save the floor space as discussed in Chapter 21.

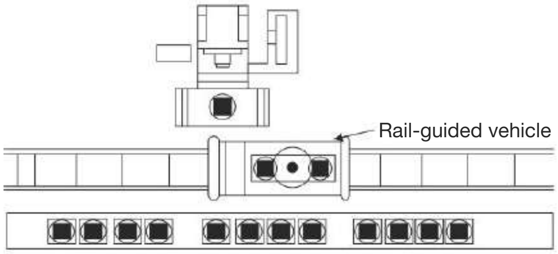


Fig. 22.4 *Machining centres for use in FMS with a pallet pool*

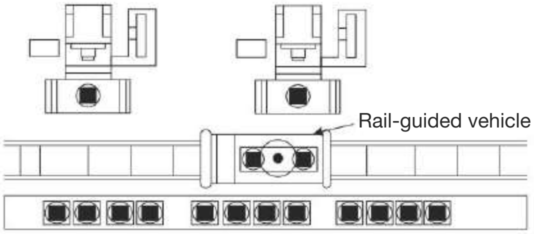


Fig. 22.5 *Flexible machining cell with two machining centres*

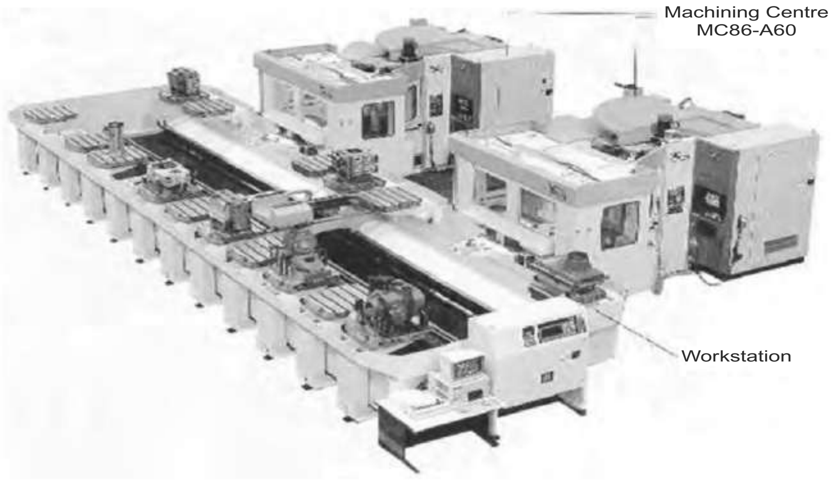


Fig. 22.6 *Actual flexible machining cell with two machining centers (Courtesy, Makino milling machines Co. Ltd., Tokyo, Japan)*

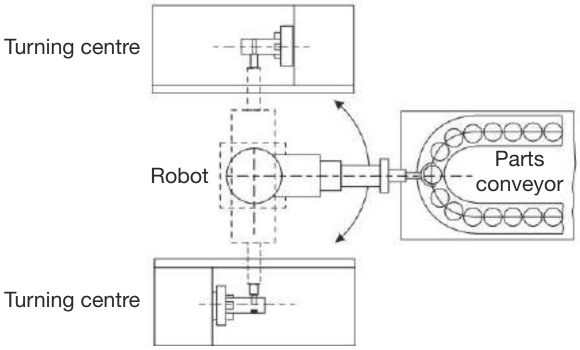


Fig. 22.7 *Flexible machining cell with turning centres and a robot serving as the material handling unit*

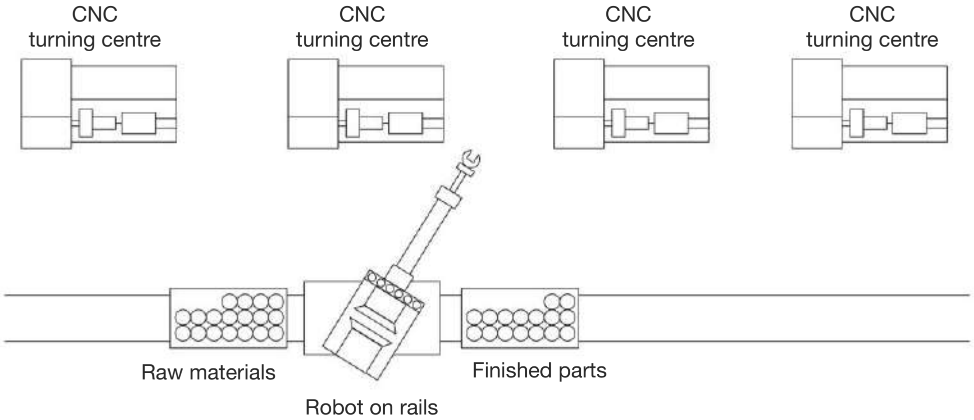


Fig. 22.8 *Flexible machining system with turning centres and a robot on rails serving as the Material-handling unit*

Table 22.3 shows sample FMS in operation in USA for the last more than a decade.

Table 22.3 *List of some FMS in operation USA*

Company	*FMS Location*	*Product*	*Supplier*
Avco Lycoming	Stratford, CT	Aerospace engine components	Kearney & trecker
Boeing	Auburn, WA	Aircraft components	Shin Nippon Koki
Borg Warner	York, PA	Compressor components	Comau
Ford	New Holland, PA	Sheet metal parts for farm equipment	Trumpf
General Dynamics	Fort Worth, Tx	Components for F-16 aircraft	Westinghouse
Ingersoll Rand	Roanke, VA	Hoist Equipment	White-Sunstrand
Mazak	Florence, Ky	Part for machine tools	Mazak
McDonnel Douglas	Torrence, CA	Aircraft components	Cincinnati Milacron
	St. Louis, Mo	Missile bodies	Giddings & Lewis

22.3 TOOL-MANAGEMENT SYSTEMS

Tooling is one of the most important element in a manufacturing system and hence in FMS Special attention need to be given to see that the right tools are made available for machining without any delay.

22.3.1 Tool-Supply Systems

In an automated manufacturing system, cutting tools have to be taken out and supplied into the system at intervals depending upon their utilisation. If one considers the stand-alone machining centre, tool magazine of the automatic tool changer be supplied at the beginning of the shift, all the tools in refurbished condition. If any of the tools has to be changed during the operation, then the machine tool may have to be stopped for changing. This would be expensive in terms of the lost production time on the machine, and hence alternative better means have to be found for replenishing the tools in the system. The tools need to be replaced while the machine is cutting.

But with a number of machine tools, the problem gets compounded further. Also in FMS, whenever the part spectrum to be manufactured gets changed, the tooling required may have to be altered accordingly. Hence more and varied solutions have been tried by various machine tool manufactures. A brief comparison of these systems is presented in Table 22.4. The basic concept in all the systems to get a secondary (auxiliary) tool storage from where the required tools can be transferred to the main tool magazine where and when necessary without much effort and loss of cutting time.

Table 22.4 *Tool-supply concepts in manufacturing systems*

Tool storage and transport	*Storage capacity per machine tool*	*Time fraction per tool change over*	*Bound capital for tools in the manufacturing system*
One machine integrated circulating magazine	40–100 tools	Large	Large
Two machine integrated circulating magazine	60–120 tools	Small	Large
Interchangeable tool magazines	20–60 tools	Medium	Medium
Successive Interchange of a segment of tools from the magazine	40–100 tools	Small	Medium
Successive interchange of single tools from a stationary auxiliary magazine	20–40 tools	None	Medium-small
Separate tool highway	20–40 tools	None	Small

In the conventional method of tool changing where the machine tool will have to stop for the complete tool magazine refurbishing with new tools or for single tool exchange. Such storage units as drums, chains, discs and other forms are used. There is a limit to the maximum number of tools available at the machine tool in this form. The limit may be of the order of 120 or so. Refurbishing of the entire tool magazine is normally done during the start of the shift. Care has to be taken to see that the tools loaded complete all the machining till the end of the scheduled period. In an automated system this would call for large machine stoppage time for complete tool exchange as well as make the cost of tools in the system prohibitive. There is no secondary tool storage system available close to the machine tool.

In the second case, the traditional system is modified slightly. The tool magazine is split into or more smaller magazines, so that the machine tool can be running while one of the tool magazines is being replenished. In cases where a number of tools are to be augmented, more than two tool drums (or discs) have also been used as in the Yamazaki Minnokamo factory in Japan. Sometimes the second and subsequent tool drums (discs or chains) carry special tooling required less often for special jobs. One of the disadvantages is that if a job requires more tools than can be accommodated in the small capacity of the drum or disc, more frequent disc transfers would be required. This would make the tool change time small,but would increase the cost of tools in the system higher than in the previous case. Hence this method is not widely practiced.

In the third system, an entire tool magazine is swapped for replenishment so that tool resharpening and replacement into the magazine can be done in the tool crib. An automated guided vehicle carries the tool magazine from the machine tool to the tool crib. Though this reduces tool changeover time, the additional cost of a replaceable tool magazine and the system of transporting it to the tool crib and back, makes it a more expensive proposition. However, it is possible to reduce the total number of tools in the system by making for a tool magazine with fewer tools. Though a number of systems, such as Yamazaki, Cincinnati Milacron, etc. were tried in the early eighties, this method is not being widely used in any of the newer systems.

Another alternative to the above system is, instead of replacing the entire tool magazine in one go, breaking it into smaller segments called cassettes, so that each individual cassette is replenished at a time in place of the complete magazine. This would mean a small tool change time and comparatively straightforward way of replenishing the tools. The number of tools in the system would be more or less the same as in the first case.

There are a number of ways in which this method is adopted. In the Werner and Kolb system called *Quick Tool Change* (QTC), the cassette is in the form of a linear array capable of storing eight tools. A simple transfer machine which removes old tools with new tools by a comb cassette is incorporated. The chain tool magazine used in this case is divided into a variably overlapping storage and changing area as well as fixed areas for standard tools and worn tools. The machine tool automatically deposits the tools in the respective areas after their use in spindle. The movement into the respective areas of the magazine chain takes place parallel to the machining time. The tools, which arc worn out are first transferred to the tool positions meant for transfer. Then the tool chain indexes to bring all the tools to the transfer position. The empty comb cassette would then transfer the tools from the chain. The cassette is withdrawn while a new cassette with fresh tools is brought to the tool change position. These tools then get transferred to the tool chain.

Another approach for a selective change of tools between the primary tool magazine and the secondary tool storage is to have a robot arm stationed between them. The robot arm can transfer tool, as and when required. The secondary tool storage is exclusively meant for the single machine tool. This is the most generally adopted system in the case of a turning centre or turn-mill centre when modular tooling is employed. The modular tooling system may be the *block tooling* system of Sandvik or *Widaflex* or *Multiflex* tooling of Widia, Karl Hertel's *Flexible Tooling system*, or Kennametal's *KM Tooling*. In this system the tools to be transferred in the secondary tool storage are arranged as a linear matrix, from where the robot arm transfers it to the main tool turret (Fig. 11.22 and 11.23). This type of tool magazine stores a large number of tools to provide for an uninterrupted production cycle. The same gantry robot transfers the workpiece (with different grippers) kept on pallets.

Though this system suits a turning centre admirably, because of the low weight of the turning tools and parts, operating it would be difficult in the machining centres. In the case of turning centres, the tool change robot may also be used for work handling by adopting it as gantry with larger work envelope as used by Boehringer, Monforts, Traub and others. The gantry robot has the necessary capability to transfer the turning tools as well as the power tooling used in the turn-mill centres. This allows for the turning centre to effectively become a *flexible turning cell.* Also, the large expense involved in the making of the stand-alone material handling unit for each of the machine tools would make the system economically unsuitable. The tool transfer time in these systems is negligible since the transfer is affected when machining is being carried out.

One system which is currently emerging as the most promising and which has been adopted by practically all FMS manufacturers, is the single tool transfer method. Here each individual tool is transported to the tool magazine by means of a tool change robot traveling along a tool highway. This involves a single secondary tool storage consisting of all the tools for the complete system of machine tools. This would effectively reduce the duplication of expensive tooling, since the same tool can be used on more than one machine tool. It also helps in the simple upkeep of all the tools required for the system. The replenishment of the tools in the secondary tool store is generally manual.

Another system of similar type, called a tool hive is used by Yamazaki. In this system, all the tools are located in a simple tool matrix of linear arrays (secondary tool store) as shown in Fig. 22.9. The tool transfer robot travelling along a mono rail system automatically loads and unloads tools from their designated pockets in the tool hive to the ATC position as directed by the control system as shown in Fig. 22.10. The tool hive is capable of storing upto 480 tools in various modules starting from 160. The tool hive can be used for a single machine tool in which case the tool highway would not be required.

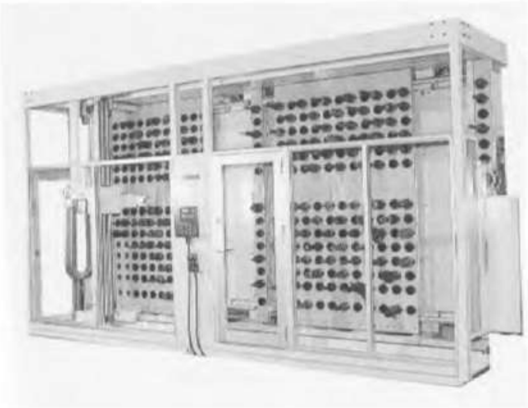


Fig. 22.9 *Secondary tool storage (tools hive) for FMS (Courtesy, Yamazaki Mazak Corp., Japan)*

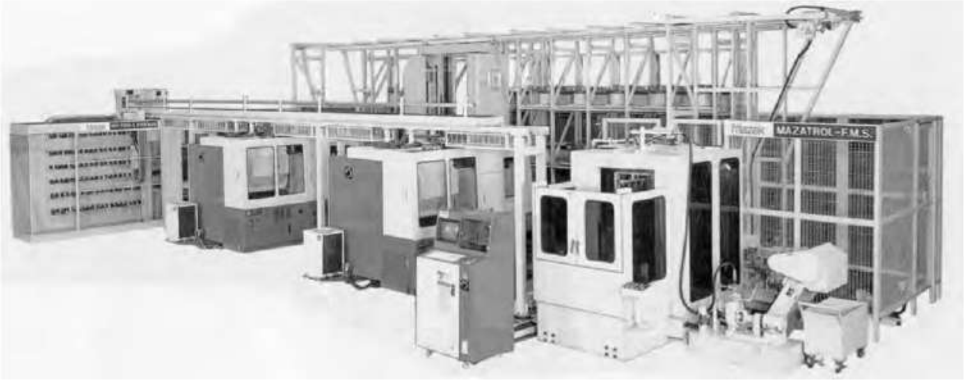


Fig. 22.10 *Tool hive integrated with a FMS using tool transfer robot moving on the monorail. (Courtesy, Yamazaki Mazak Corp., Japan)*

In the tool hive there is a fixed position where the operator loads simply the tool to be transferred into the system. The tool transfer arm then transfers the tool from this location to the pocket number entered by the operator. From this pocket the tool is transferred to the machine tool magazine by the transfer robot arm shown in Fig. 22.9, which can move along two axes. Similarly when the tools in the system are worn out, they are transferred to the tool hive, from which the operator can remove them and refurbish them for operation in the next shift. By carefully adjusting the tool drum capacity at individual machines, and proper control software, the required tools are transported to the machine tools, without making the machine tool wait for the tools.

22.3.2 Tool-Monitoring Systems

In the case of tool monitoring systems, the tool has to be continuously monitored while it is cutting. This would allow for continuously looking for tool wear, as well as the times when the tool breaks because of unforeseen conditions in the machining system. The various methods adopted for these two functions are slightly different.

Tool-Wear Monitoring Tool wear is a phenomenon whose behaviour can be explained qualitatively but not quantitatively. Though some tool life equations do exist, their universal adaptability or their utilisation even in restricted work-tool material zones for all parameter ranges are doubtful. Further, direct in-process measurement of tool wear, is difficult in view of the location of the wear and the measurement techniques employed.

An important process parameter is the tool wear, which may be measured directly or indirectly.

The following principles are generally used for direct measurement of tool wear.

(i) Tool wear is measured by relating it to changes in the resistivity of a resistor embedded in the tool tip. In this method there is no need to interrupt the process.

(ii) The profile of the tool tip is recorded periodically using optical methods and tool wear is determined from the variations.

(iii) Opto-electrical methods using TV cameras and photodiodes etc. are employed to record variations in the cutting edge to measure the width of the worn edge.

All the above methods are complex and are more expensive to implement on a regular basis on the production CNC machine tools. As a result, tool wear measurement has to be indirect. Some parameters used for measuring tool wear are

- Cutting power
- *Cutting forces* By measuring cutting force using force measuring sensors. The cutting force increases with the increasing dullness of the tool and can therefore be related to tool wear.
- *Vibrations* By measuring vibrations of the tool edge, i.e., tool chatter wear can also be indirectly estimated.
- Acoustic emission
- *Tool temperatures* By measuring the tool tip temperature and relating it to the wear of the tool.

Of these variables, mainly the cutting forces and power based tool monitoring systems are commercially and widely available, whereas the others are still not proven to be widely used in practice.

The power consumed during a machining process is a function of the forces acting. Further, the cutting forces depend upon the quality and condition of the cutting edge. As time progresses, the power consumed by the tool for the same material removal increase with increase in tool wear. Thus power measurement is an indirect way of monitoring the life of the tool. Power may be measured by a power meter (watt hour meter) installed in the spindle motor circuit. However, a more effective method is to determine the resistance offered to the tool motion, which can be measured in terms of the power consumed by the axes motors. It was found that in applications like drilling, measuring the current consumed by the feed motor in the spindle (Z-axis) direction would be a good indicator for tool condition.

Another system of tool condition monitoring is by the measurement of the torque on the main spindle as used by Maho. In this system the spindle torque is measured in terms of the differential twist separated by a small distance.

Direct measurement of cutting force is a better method for tool condition monitoring, rather than power. Hence a number of systems based on force measurement are available commercially. The problem encountered often is that the force sensor should be located close to the source of power, i.e. the tool. However, with the ability to change tools during the machining, it is necessary to have a force sensor located in the tool holding structure rather than with the tool itself. Thus Sandvik has come up with a plate sensor which can be conveniently mounted under the tool turret in case of turning centres.

The plate sensor consists of a structure, which is simple and has strain gauges to directly measure the feed forces. The simplicity of the construction of the plate sensor helps in its easy adaptability to any machine tool. The threshold signal for limiting tool wear from the plate sensor is linked to the CNC controller which can automatically call up a sister tool for replacement of the worn out tool, before a catastrophic failure. This also allows for uninterrupted cutting, and reduces the costly break- downs of the CNC machine tool.

Another possibility of tool condition monitoring is through the measurement of vibrations of the cutting tool. The vibration signature of a cutting tool is a good indicator of the quality of the cutting edge. The vibratiosn spectra at the beginning when the tool is sharp can be compared with those at each time, and the shift taking place in amplitudes and dominant frequencies can be measured. These are useful for identifying the tool failure criterion. The failure criterion used is the power spectral density of the cutting tool vibrations, which is a good indication of the energy consumed for cutting by the cutting tool.

Another method generally used for tool life monitoring by many control manufacturers is the software monitoring of the actual amount of time a tool is cutting versus the suggested tool life. This is discussed later.

Tool-Breakage Monitoring Another problem often faced is the breakage of tool during cutting, which if not detected in time may lead to various problems associated with spoiled jobs, particularly in unmanned machining shifts. Hence it is necessary to have systems which can detect the breakage of tools through some means and give an alarm to the operator, or automatically replace the tool by a sister tool from the secondary tool storage.

In the plate sensor or any other cutting force measuring system, it is possible to detect a drop in cutting force almost to zero from peak during a breakage. The force drops since the tool may lose contact because of tool breakage. Hence the controller can be given the signal for either sister tool change, or stoppage till the operator attends. This is the most effective system and commercial systems are available from Sandvik and Krupp Widia.

Another method used by a number of machine tool manufacturers is to check the tool length or tool offset at the end or beginning of the cut using the tool probes described earlier. This value is compared with the tool values stored for the new tool. If the difference is more than a certain value (typically 1 mm), it is considered as tool breakage. This type of system is simple, but can detect tot breakages only after a cut. Any tool breakage during cutting remains unnoticed. However this system does reduce the effective damage caused by broken tools.

Ernault-Toyoda uses a patented *'current loop'* through the machine for tool breakage monitoring. As shown in Fig. 22.11, a small current flows through the machine and the loop closes only when the tool touches the workpiece and remains in contact with it. When the tool just touches, a signal is transmitted to the CNC control, which can be used for various purposes depending upon the situation Normally the tool is expected to come in contact with the workpiece with in a specified distance which can be determined from the CNC part program. If the contact is not detected within that distance, the tool is broken and hence the machining cycle breaks. Proper action has to be the initiated to continue with the machining. They use this novel facility for reducing the tool idle time by accelerating the tool movement when the contact is not established between the tool and workpiece by doubling the programmed feed rate.

Another approach employed by Deckel is the use of an infrared beam to detect the tool breakage. At the end of each machining sequence, the spindle moves the tool automatically to the measuring, location where the tool is expected to break an infrared beam. If the beam is not broken, the too length is small and hence indicates tool breakage. Instead of infrared beam compressed air flow can also be used for measuring the tool breakage. As shown in Fig. 22.12, the airflow is broken a full tool, whereas the flow is continuous in the case of a broken tool. This difference can be measured with the help of a pressure sensitive switch that can communicate to the MCU about the tool condition.

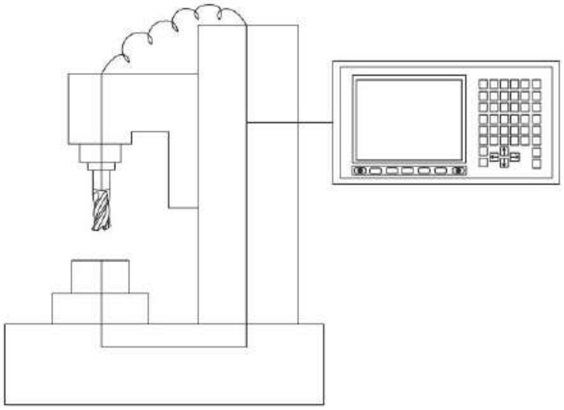


Fig. 22.11 *Schematic of a set-up for tool breakage monitoring using current loop through the machine tool*

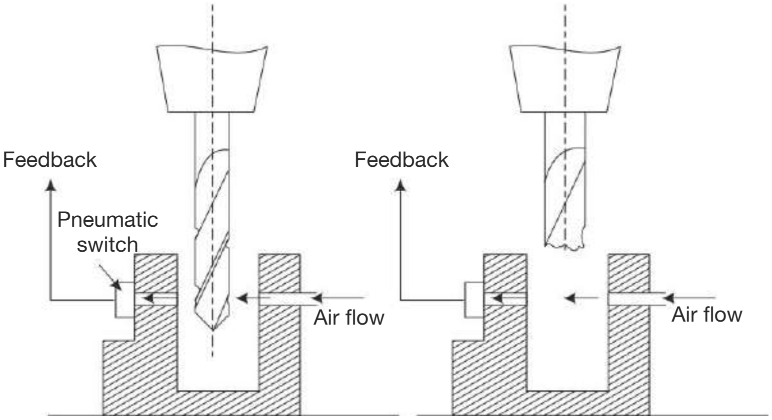


Fig. 22.12 *Typical system for tool breakage monitoring using compressed air at a fixed position probe in a machining centre*

22.3.3 Tool-Management Systems

In the previous sections, we have discussed various methods for monitoring the status of the tool as actually used on the machine tool. We shall extend this further, and now present software systems, which have to track the tools in the manufacturing systems.

Tool-Life Monitoring There are various systems available as part of modem CNC controllers, which are termed tool management systems that are able to keep track of the actual time for which each of the tool is in use. When the tools are entering the system (loading in the tool magazine), the expected tool life is entered in the controller memory along with all other tool related information such as offsets. This value generally a conservative estimate of the tool life expected based on the work materials and the process parameters used. This value can be generally obtained from machinability data banks used on th shop floor or from previous experience.

As machining proceeds with various components, the controller records the actual time for which each tool is used. This time used when subtracted from the expected tool life gives the left-over tool life for each tool. When the left-over tool life is less than a certain nominal value, the controller initiates the action for replacing it by a sister tool. Controllers manufactured by Yamazaki, Fanuc, Sinumeric and others have this type of tool management option available with them.

It is also possible to have tool life monitored outside the controllers, for example the tool life of a the tools present in the manufacturing system consisting of more than one machine tool may be monitored in an IBM PC compatible.

A true tool-management system is much more than a simple tool life monitoring system an interfaces with the factory information system as shown in Fig. 22.13. This system would not only see that the tools be kept as required but also look after all the associative functions.

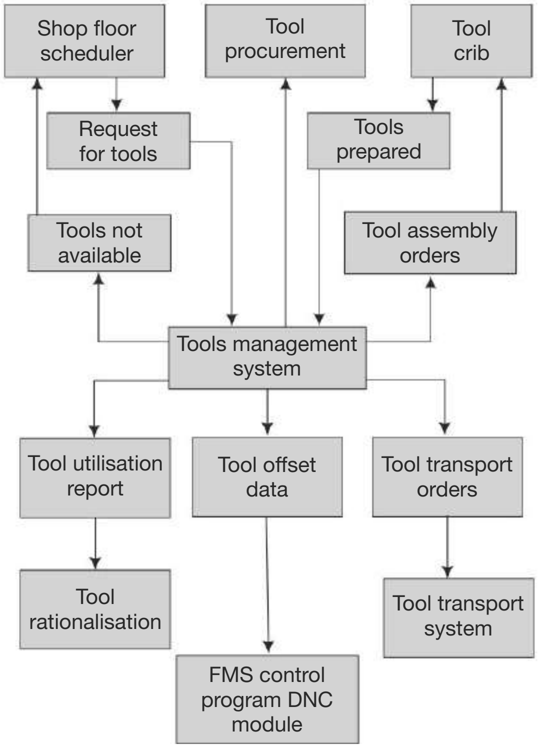


Fig. 22.13 *Schematic of information flow in a tool management system in a manufacturing organisation*

The shop scheduling system, while scheduling the jobs for the day (or any scheduling horizon) may interact with the tool management system such that the tool availability is ensured for the part being manufactured for that day. Similarly, with inputs from the scheduler, the tool management system may generate the requirements for the tools to be prepared and thus the requisite instruction may be given to the tool crib for tool preparation requirements. If any of the requisite tools or some elements are not available, then procurement procedures could be initiated. If the tools are ready at the tool crib, the necessary transport orders for moving them to the machine tool locations using AGV or any other transport mode may be initiated. Further, the tool offset data may be compiled and tool offset programs can be downloaded to the CNC machine tools as and when required. Finally, periodically the system can generate the tool utilisation, showing the various tools and their utilisation reports. This would help in tool rationalisation by removing the less used tools from the system and modifying the manufacturing procedures.

22.4 SYSTEM LAYOUTS

Some example possibilities of FMS layouts are shown earlier. The broad categories of layouts that have been used are [Groover 2001]:

In-line Layout All the machine tools are kept along a straight line as shown in Fig. 22.14. This is the simplest form and is generally used for smaller number of machines in a system. The parts move in well defined sequences and the workflow is generally in both the directions. The part handling at the individual workstations is performed by the transport vehicle, which will have the necessary pallet changer. Often the machine tools used in such a system are identical, so that the part routing will not be a problem.

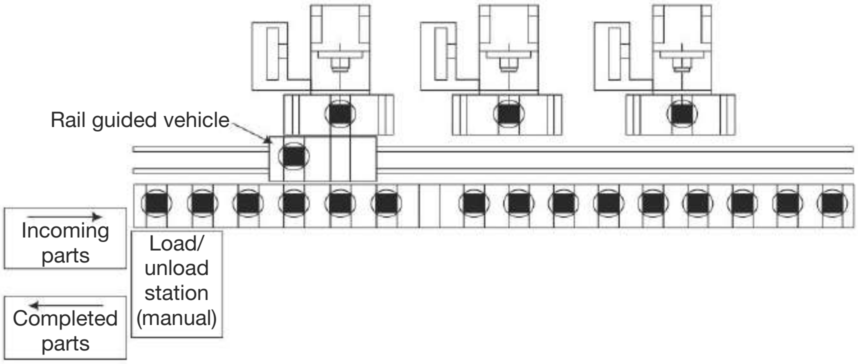


Fig. 22.14 *In-line FMS layout*

Loop Layout In this system the workstations are arranged in a loop as shown in Fig. 22.15. Parts generally move in a single direction in the loop similar to a conveyor, with the ability to stop at defined positions for transferring the parts to the workstation. For the purpose of moving parts from the conveyor to the workstation may have to be carried by means of a secondary part exchange system such as a pallet changer as shown in Fig. 22.15. An alternative form of the loop could be rectangular.

Ladder Layout In this system the workstations are arranged in a loop with rungs as shown in Fig. 22.16. The rungs help in reducing the congestion and allow for smooth part flow between machines.

Open Field Layout In this system there are multiple loops for appropriate arrangement of all the facilities as shown in Fig. 22. I7. This type of system is generally suitable for a large group of parts to be machined. The facilities may consist of a number of workstations with different varieties. The material handling is provided with AGVs along the guide path.

22.5 FMS CONTROL

All the elements present in an FMS, such as the CNC machine tools, material handling units (AGVs), workpieces and tools are to be controlled in real time. This means that the FMS control software ensures for timely supply of tools, workpieces and part programs to connected machine tools as shown in Fig. 22.18. This should have modules for the following:

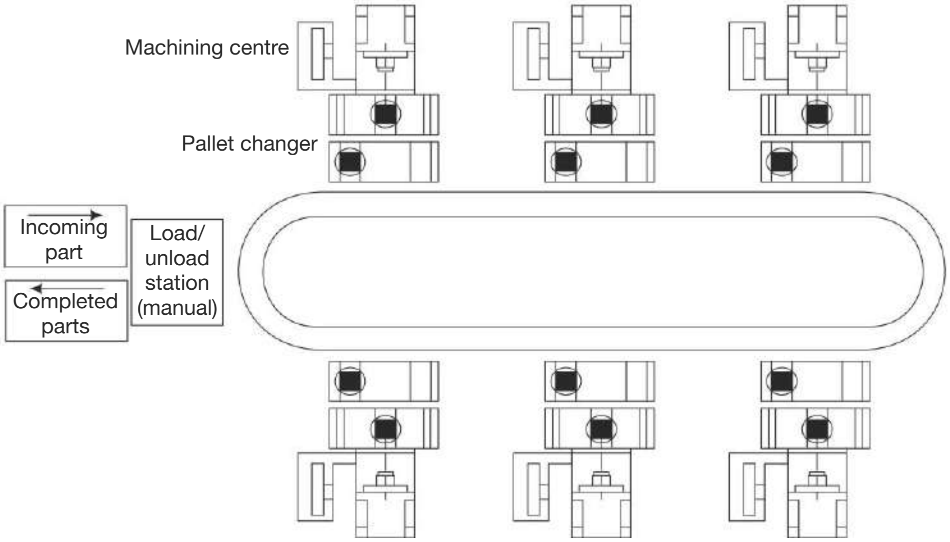


Fig. 22.15 *FMS loop layout*

- *Production scheduling* to schedule various production operations of the FMS based on the parts to be made entered at the Remote Job Entry (RJE) terminal.
- *Transport management* to take care of the workpiece and tool movements in the FMS under the direct control of the FMS supervisor.
- *Tool management* to arrange the availability of the right tool in the right condition at the right time in the right place. This provides the necessary tool offsets required by the various programmes.
- *Simulation* is a powerful tool, which can be used for the design purpose as well as for knowing the condition of the present status of the production operations within FMS.
- *Production control* coordinating various production operations of the FMS modules by direct communications with their controllers (CNC, AGV, etc.)
- *Machine diagnostics* to obtain any malfunctions of the FMS modules.
- *Managing* part programs, data files of tools and workpieces and their storage positions.

Maintenance planning based on the feedback on the health of the FMS components to properly plan the maintenance schedules.

Some of the operations shown above are time critical and hence need to be done on real time basis, while many functions are not time critical and hence can be done when the processing time is available. For this purpose a distributed control is generally preferred with the real time control done with a slave computer dedicated for the application

The typical operating procedure of an FMS is as follows:

- In the beginning of the day's job orders will be entered, either directly or downloaded from the plant computer.
- The system checks that the route sheet and process plans for all the jobs to be done are available. Else action has to be taken to initiate to obtain this information. In the mean time those jobs are to be removed from the job list. They can re enter when the process plans are available.

- Check for the availability of the machine tools, raw material and cutting tools. It is necessary to ensure that the required tools and workpieces in a form suitable for immediate use be made available before the start of the day. For this purpose, it is possible to have a look ahead capability in the software to make a trial schedule run a little earlier to provide instructions to the tool crib and work preparation areas.
- Once having all the equipment in place, schedule the parts based on the priorities assigned. Make a simulation run of the schedule to see that everything is in order. Based on the prepared or feasible schedule create the job list sequence and create the various tasks in chronological order to be executed by the FMS controller.
- Send tool orders to the tool crib and fixture and blank orders to the work preparation areas.
- Start executing the created sequence of tasks as shown in Fig. 22.19.
- It is possible to enter any new orders, which may change the created task chronological order. Also some priorities for the jobs may be changed in between, which needs rescheduling of the remaining tasks and provision may need to be provided in the software for this function as well.
- The current status of jobs can be seen on the FMS terminal from where manual intervention by the supervisor can be initiated when necessary. In addition a number of reports on the various elements of FMS can always be made

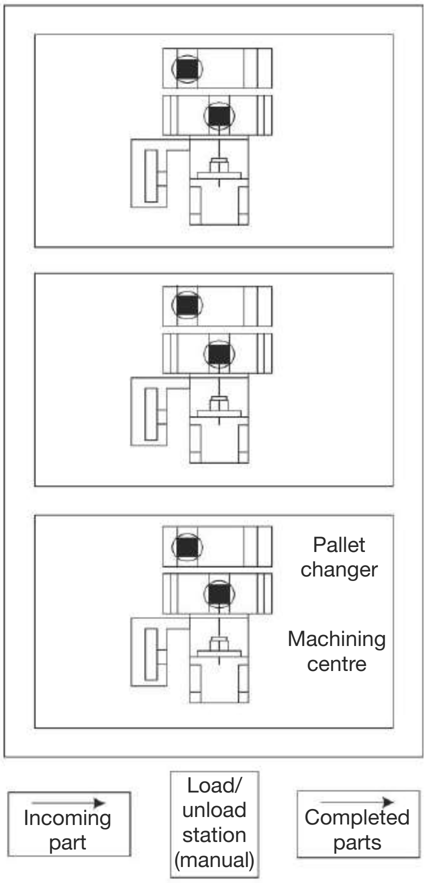


Fig. 22.16 *FMS ladder layout*

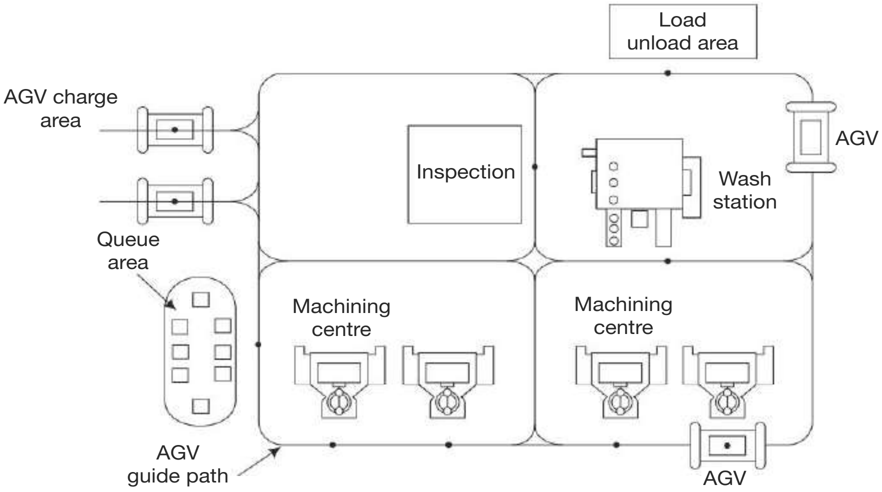


Fig. 22.17 *Open field FMS layout*

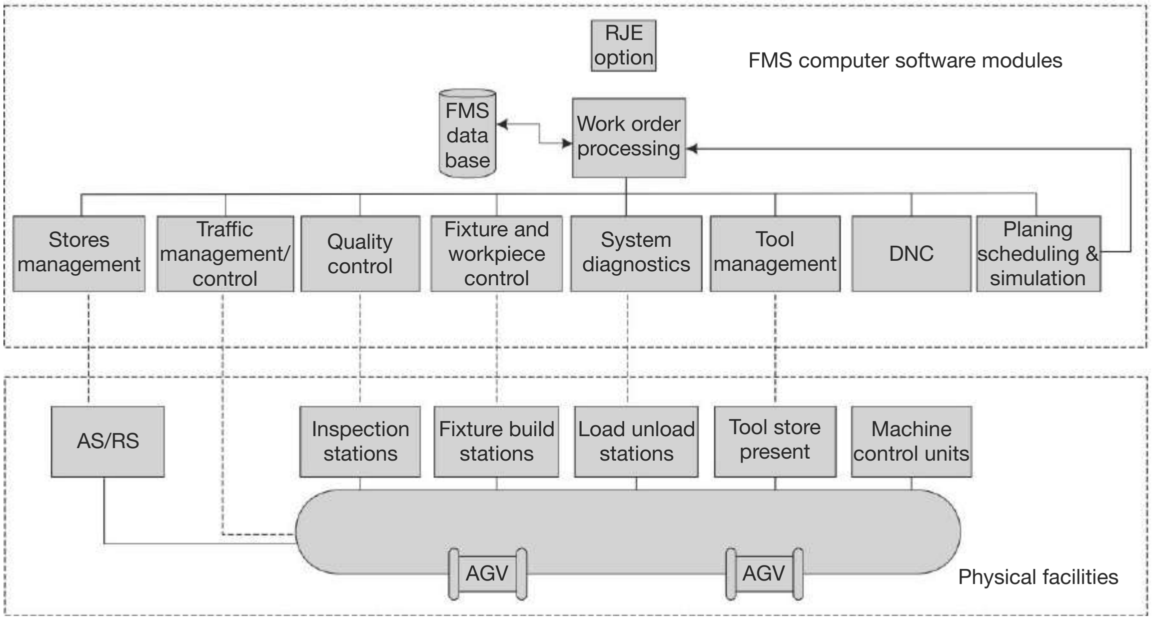


Fig. 22.18 *Schematic of software modules and their linkage to the physical facilities in FMS*

available. This will help in the initiation and implementation of optimal decisions by the supervisor when necessary.

FMS Controller Functions Typical FMS controller software functions and their interface with the hardware are shown in Fig. 22.20.

Capacity planning module checks the FMS for the early due date of the parts released for the FMS. This will be able to specify as a rough guide the following information, which needs to be accepted by the user.

- Machine loading
- Pallet/fixture requirement
- Tools required
- Due dates for completion of the parts

Selection of the planned order depends upon

- Tool mix per machine tool magazine
- Transport units required
- Machine capacity/loading required
- Part programs required

In case of any problems it may be possible to provide some choices to the operator to

- Choose alternate process plan
- Modify current process plan
- Select alternate product mix
- Split batch quantities

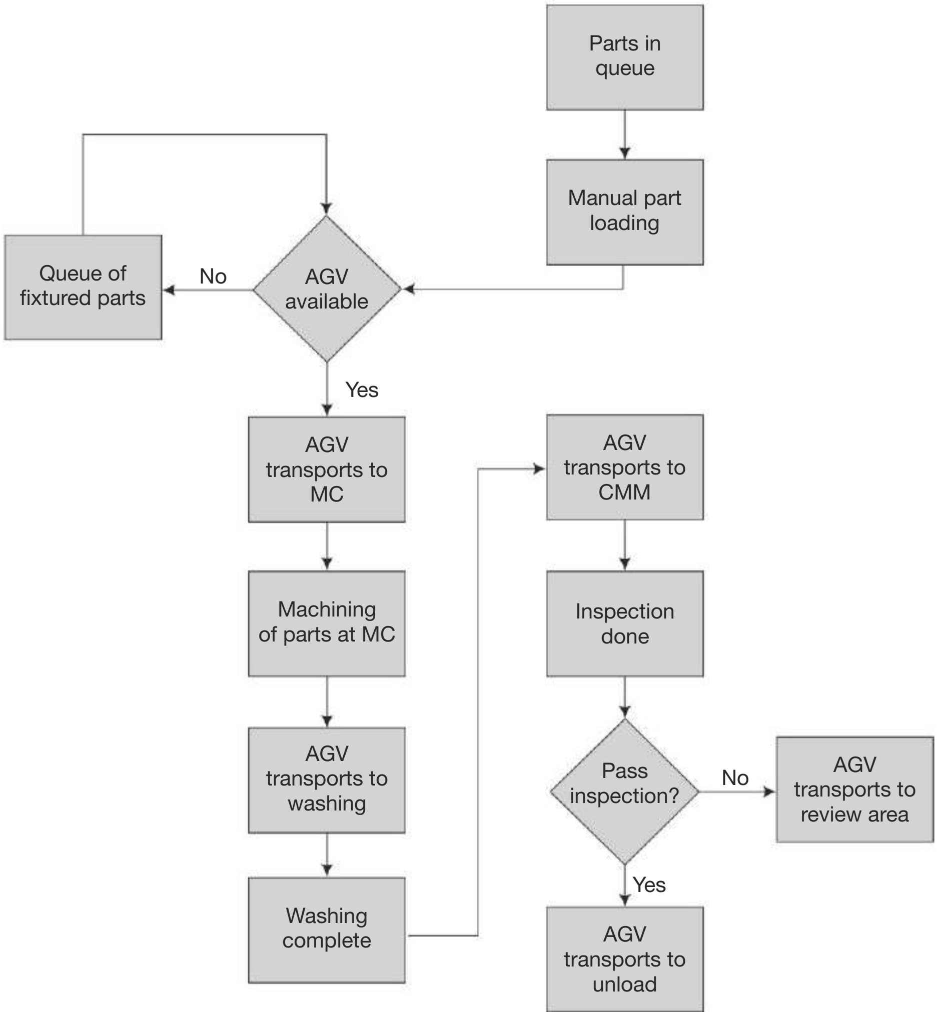


Fig. 22.19 *Part and pallet flow in FMS*

(a) Planning module can interactively give complete picture to the operator who can then make the necessary modifications if necessary and accept the results.

(b) Scheduling module creates the actual sequence of the parts and operations within the specified constraints of the resources.

(c) Simulation module is similar to a discrete event simulation that can show that the schedule created can be done within the given time frame or not taking all the factors into account.

(d) Fine planning module is meant to tweak the created schedule by the operator for any specific requirement.

In order to carry out the above software functions the data related to the FMS should be properly presented.

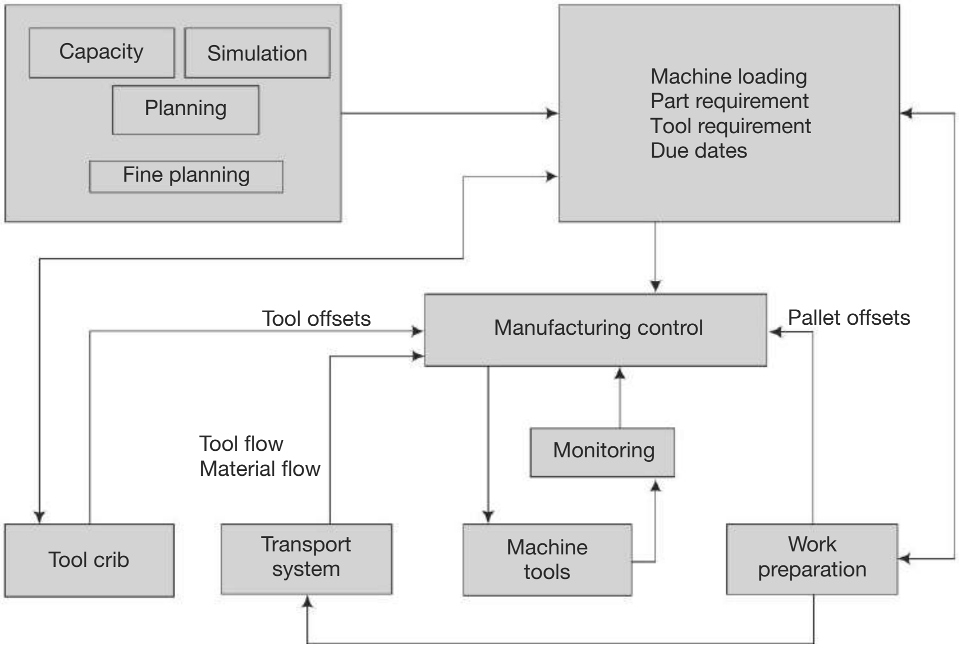


Fig. 22.20 *FMS control host functions*

(e) Master data have to be setup only once when the FMC is put into operation. This consists of the system specific data relating to the architecture of the system in terms of the machine tools and material handling equipment present, their relative positioning and other information of the layout.

(f) Resources data in terms of the resources required for the manufacturing of the specific part mix are to be entered. These are tool master data, and workpiece carrier master data (data relating to workpiece carriers and clamping fixtures).

(g) Control data (product specific data) are the actual resources required to work on the specific part mix in question such as NC programs, tool layouts and work schedules (technical control data) and manufacturing orders (organisational control data).

Manufacturing Control This is the most complex of all the modules in the FMS software. It has to execute the schedule created in real time and make corrections to it continuously depending upon the exigencies in the system operation. It has to carry out the real time control of the workstations, load/unload stations, pallets, human operators, tools and material handling units. This has to be continuously done with the aim of optimising the utilisation of all the resources of the FMS.

Monitoring It is necessary to continuously monitor the various functional units within FMS in order to achieve the unattended operation. Some of the functions monitored are

- CNC Machine tool status
- Tool lives
- Cutting abnormalities
- Tool breakages
- Error diagnostics

Tool Crib In the tool crib, which receives orders from planning for the assembly of tools required for the specific parts. After the tools are assembled, they are to be set and tool offsets measured in the setting equipment. The tool offset files will be entered into the FMS computer and also the tools will be kept ready for transportation. The necessary information may be sent to the transport module to arrange for the transport service.

Work Preparation The instructions in terms of which parts need to be setup or unloaded from the pallet will be issued. Based on which the operation will be carried out normally by the human operator and then communicate the data related to the pallet offsets to the machine tool. Further information need to be sent to the transport for getting the AGV.

Transport This module will have to administer the use of the AGVs for various requirements generated within FMS. Typical transport orders could be

- Pallet from load area to machine tool or buffer
- Pallet from machine tool to buffer or unload area
- Tools from tool store to the tool magazine
- Tools from machine tool to tool crib

Machine Tools All the information required for the running of the workstations is coordinated by this module. Examples are the CNC part programs, tool offsets, etc.

The following data will be generated during the course of the functioning of the FMS.

(a) Status data to describe the current situation with regard to resources in FMS such as plant status data, resources data in terms of workpiece carrier data and tool data.

(b) Log data is all operational data and machine data required for later analysis (diagnostics) are recorded, evaluated by the software function modules and filed with details of date and time of day (logged); such data include

Status and Messages

(c) Machine specific messages from the various workstations present in the FMS such as those from CNC, PLC, handling devices, and transport system.

(d) Status and operational messages such as NC start, NC end and NC program run time.

(e) Alarms such as machine fault or breakdown.

(f) Tool specific messages such as tool breakage or end of tool life.

(g) NC messages such as load NC program.

22.6 DEVELOPMENT OF THE CONCEPT

Having understood the various concepts involved in the Flexible Manufacturing Systems and Cells, let us look at a few of the details and procedures that should be used in developing a cell for any given product spectrum. It is understood that the various products that have been grouped into a family to be manufactured in the cell was already completed. The various steps involved in the process are (Nyman, 1993):

- Step 1: Gather and record data
- Step 2: Develop process flow within the cell
- Step 3: Identify equipment required and compare it to what is available
- Step 4: Select/assign equipment
- Step 5: Build a relationship diagram and an initial cell layout
- Step 6: Recapture data in the planned cell structure

We will try to explain briefly some of these aspects with the aim of helping in the design of flexible manufacturing cells.

Step 1: Gather and Record Data

This will be the first step and the most important one. In this step all the technical data required for designing the cell need to be captured. As much detail as possible about the technical details of the processing operations need to be collected. The following is the typical technical description that needs to be collected.

- Product mix and volume that is expected to be manufactured in the cell
- The technical characteristics of the parts that form the family or group may include
  - Size and shape
  - Material
  - Lot size or volume
  - Process steps or unique processes
- Routing and process sheets for all the parts which should include
  - Equipment, tooling and gages
  - Likely upgrades or process changes
- List of equipment available for the cell
  - Capacity constraints, if any
  - Common equipment needed for other cells
- Planning guideline
  - Product life cycle
  - Future volume predictions
  - Management constraints
  - Key design changes or new product information
- Operation guidelines
  - Annual working days
  - Contractual or other constraints
  - Floor area
  - Special utilities
  - Material flow and service relationships with other facilities
- Cost and staffing baseline data

Step 2: Develop Process Flow within the Cell

Next step should be the development of process flow diagrams for all the components to be manufactured in the cell. Typical process flow diagram is shown in Fig. 22.21. This provides the information related to the actual flow of the products through the equipment in the cell. This will help in calculating the total workload in the cell. While developing the process flow it is important to consider any of the possible variations that may occur in the volumes or capacity. Variations generally occur in three areas:

- Variation in capacity due to vacations and holidays
- Product life cycles and introduction of new products
- Seasonality in the market

Any of these variations can easily affect the capacity planning done based on the annual averages. These therefore will have to be taken into account during the capacity planning.

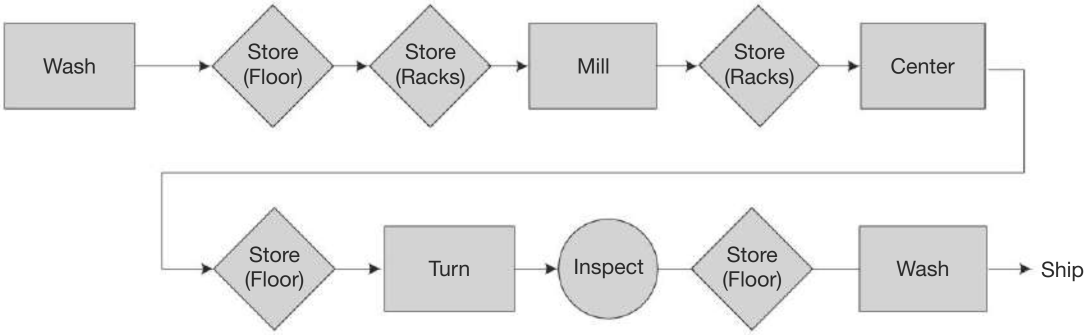


Fig. 22.21 *Process flow for a component*

The current flow diagrams should capture all the steps involved in the current flow pattern. That may include the details such as movement to buffers and wash areas. It is important at this stage to note any of the non-value added operations involved. With this information it will be possible to examine the possibility of the reduction in the non-value added operations. The designers should critically examine the current process flows to see any possibility of reducing the costs and operations. The possibilities are:

- Reducing or eliminating non-value added operations
- Eliminating the operations
- Combining the operations
- Streamlining by standardizing the operations or materials
- Reducing the operations that are done on a very small number of parts in the cell

Any possible improvements to the process flow by altering the sequence or changing some other operation could be examined in this stage. Some examples are:

- Cleaning and deburring operations can often be eliminated by changing the machining techniques
- Simplification of raw material can often be achieved by adding unique characteristics at the beginning of assembly line.
- Careful selection of cutting fluids can result in the elimination of washing with the cutting fluid doing the flushing operation

All the improvements that have been made will then be captured and shown as the current process flow diagrams as shown in Fig. 22.22.

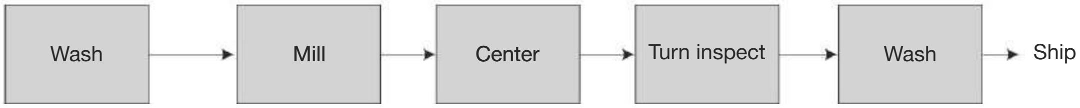


Fig. 22.22 *Improved process flow for the same component shown in Fig. 22.21*

Step 3: Identify Equipment Required and Compare it to What is Available

Once the improved process flow and the required volumes are established, it will be possible to generate the total capacity required. A sample process volume calculations are shown in Table 22.5. At this stage it may be important to consider the setup reduction as a means of getting the maximum benefit from the cellular operation. In these calculations the utilization of all the resources, viz machine tools and the operators will be shown based on the average annual volumes taking into account the variability as explained earlier. This will therefore show the resources that are over- or under-utilized in the new environment. It may therefore be possible at this stage to explore the possibility of adding new equipment to relieve any of the possible bottlenecks or those contributing to the increase in the throughput time in the cell. For example, the wash station could be one such example if it is either a centralized facility or shared by a number of cells.

Table 22.5 *Process volume chart*

Process volume chart							
Part number	*Volume*	*Wash*	*Mill*	*Centre*	*Turn*	*Broach*	*Grind*
12345	550 000	1 100 000	550 000	550 000	550 000		
23456	235 000	235 000	235 000	235 000	235 000	235 000	
34567	178 000	356 000	178 000		178 000		178 000
45678	43 000	43 000			43 000	43 000	
56789	3 000	3 000	3 000	3 000	3 000		3 000
Total	**1 009 000**	**1 737 000**	**966 000**	**788 000**	**1 009 000**	**278 000**	**181 000**

Machine number	*BT-1234*	*BT-5678*	*BT-9123*	*BT-4567*	*BT-8912*	*BT-3456*
Operation time						
Std. Min. per piece	0.34	0.62	0.28	2.73	0.68	2.03
Std. Hrs. per day	39.72	39.93	14.71	183.64	12.60	24.50
Setup time						
Setups per week	0	8	7	8	2	2
Minutes per setup	0	20.5	10.0	37.7	60.0	48.5
Setup hrs. per day	0	0.55	0.23	1.01	0.40	0.32
Total hrs. per day	39.72	40.47	14.94	184.64	13.00	24.82
Operators per machine	2	1	1	1	1	1
Machines required	0.8	1.7	0.6	7.7	0.5	1.0
Operators required	1.7	1.7	0.6	7.7	0.5	1.0

It may be good idea of keeping process volume chart in a spread sheet such as Excel to carryout the what if analysis to balance the machine work load as well as the operator work loads.

Step 4: Select/Assign Equipment

In this stage select the equipment from the master list of the machines from the plant. If there is any conflict or non-availability of the promised equipment, it may become necessary to go through a revision process to see how it could be accommodated. Some of the techniques that could be tried are:

- Eliminate the contested operation through material or process change
- Bring two or more cells closer, so that they can easily share the equipment
- Recombine the product mix so that all the parts that require the contested resource could flow through the same cell
- Redesign the process so that the contested operation is done in the beginning so that it will not drastically affect the scheduling of the cell.
- Redesign the parts to take advantage of the common characteristics of the equipment in the cell

Step 5: Build a Relationship Diagram and an Initial Cell Layout

Having finalized the operations, machines and the operators required, they can be loaded into a conceptual flow diagram from which the physical cell layout could be developed. The methods used could be different depending upon the individual preferences. Most of the processes involved are simple mathematics with common sense. An example method is shown in Table 22.6 and Figs 22.23 and 22.24. The Table 22.6 shows all the process flow for the finalized parts and volumes. This shows all possible routes.

Table 22.6 *Process flow table*

Process flow table						
Part no.	*Volume*	*Op. 10*	*Op. 20*	*Op. 30*	*Op. 40*	*Op. 50*
12345	550 000	Wash	Mill	Centre	Turn	Wash
23456	235 000	Mill	Centre	Broach	Turn	Wash
34567	178 000	Wash	Mill	Turn	Wash	Grind
45678	43 000	Turn	Broach	Wash		
56789	3 000	Mill	Centre	Turn	Grind	Wash

In Fig. 22.23 each resource is analysed for the total number of operations it will have to undergo with respect to other resources. These will show the relative proximity requirements for all the resources. Using Table 22.6 and Fig. 22.23 it is possible to construct a diagram showing how these requirements could be met in a physical layout. Such a diagram is shown in Fig. 22.24. Please note that Fig. 22.24 is not the actual layout of the cell, but only an intuitive way to show the relative flow of material between the resources to establish their correct positions in the final layout.

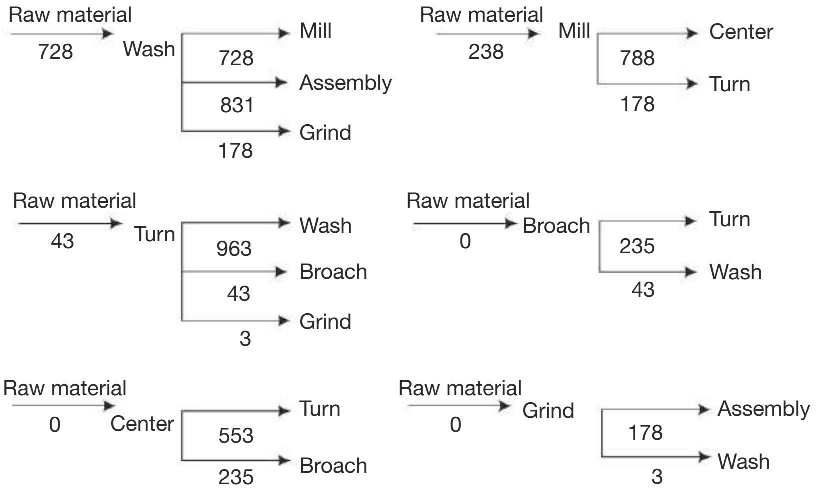


Fig. 22.23 *Operational flow chart for all operations. Figures under the arrows represent the production volumes in thousands.*

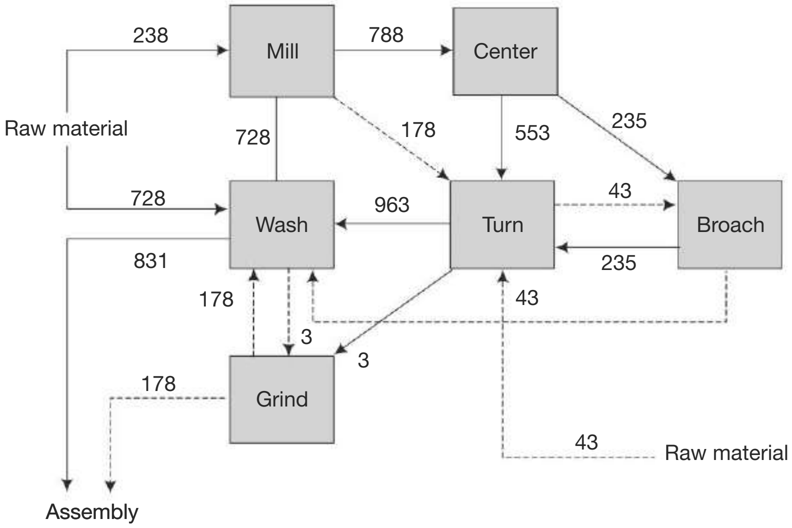


Fig. 22.24 *Modelling flow relationships among operations—production volumes shown in thousands*

Step 6: Recapture Data in the Planned Cell Structure

This is stage in which the cell layout could be finalized in actual space. At this point it is important to consider the objectives of the design. There are many ways in which the cell layout could be finalized depending upon the final flexibility envisaged. For example, a maximum speed cell design will have the machines in fixed sequence that are very close to each other. Alternatively a maximum flexible operation could be to have an open layout allowing for a wider variety of potential flow paths. Tighter layout is faster and is generally suitable for a single product cell. For multiple products, an open layout is preferred.

Figure 22.25 shows a conceptual cell developed from the earlier proximity analysis. Here the total product mix is divided into a homogeneous flow of high volume by combining all the products. From this the flow intensity is captured and shown in block format in Fig. 22.26. Finally the refined cell layout developed is shown in Fig. 22.27 with all the required spaces that are shown at the required positions.

22.7 FMS CASE STUDY

In the following a successful FMS fairly complex variety is described. This is developed by Hattersley Newman Hender Ltd., UK for machining a variety of valve components (Fig. 22.28). It took 4 years to develop with a total outlay of £ 5.5 million.

The FMS is designed to machine 2360 different parts of valves. The major components are the valve bodies and bonnets in the size ranges of 1.5 to 24 inches. The FMS is to replace the existinspecial purpose machine tools used for their manufacture. In view of the complexity of the body shape, fixtures are required for holding the bodies. Based on the past annual figures of production, the work load estimated as shown in Table 22.7.

Based on the annualised requirements the equipment for the FMS was finalised as follows:

5 FMIOO Machining centres (KTM)
2 CNC Out facing machines (KTM)
Component washing

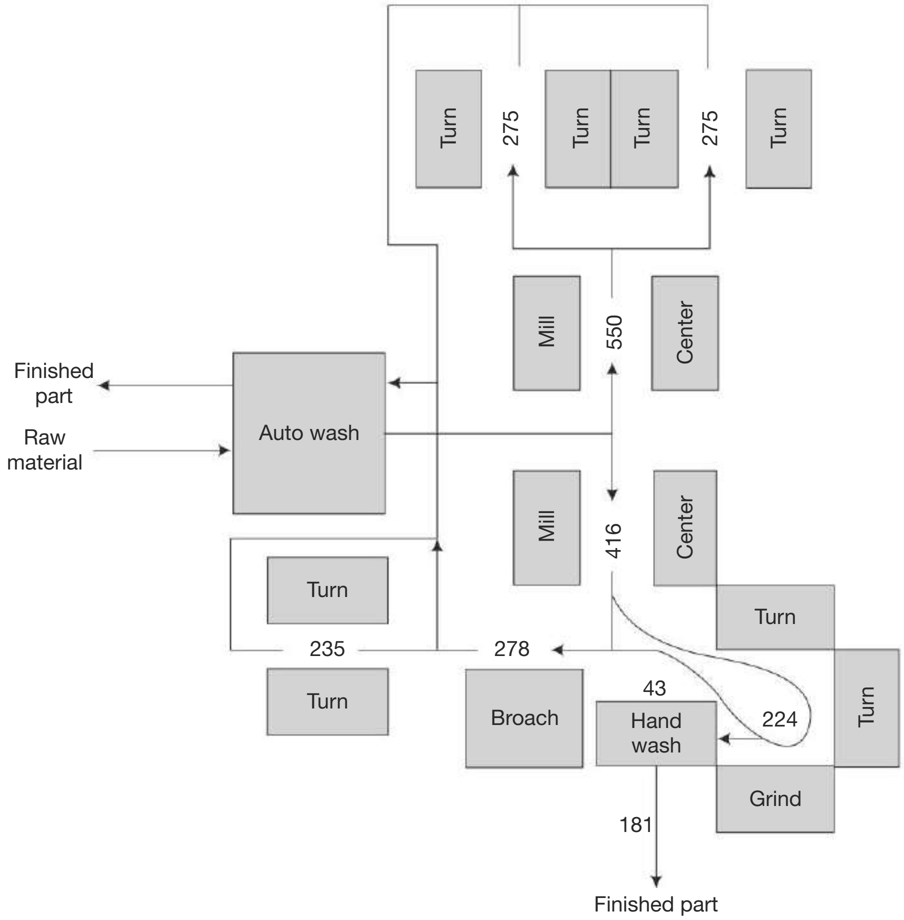


Fig. 22.25 *Conceptual cell developed using the proximity relationship and equipment data—flow volumes shown in thousands*

Seat ring fitting
Automated guided vehicles (AGV)
Fixture storage and setting (manual)
Tool setting (manual)
Raw material supply (AS/RS)
Load and unload stations (manual)
Static buffers
Battery charging for AGV
AGV waiting area
Computer control room

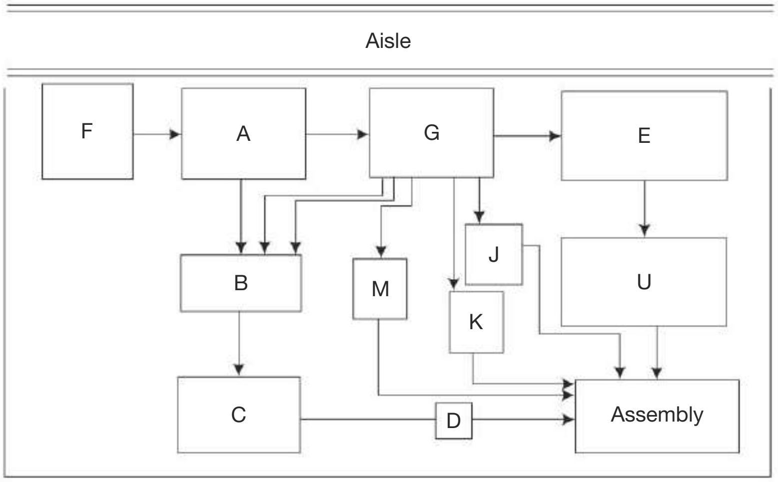


Fig. 22.26 *Machine block layout showing the part flow relationship*

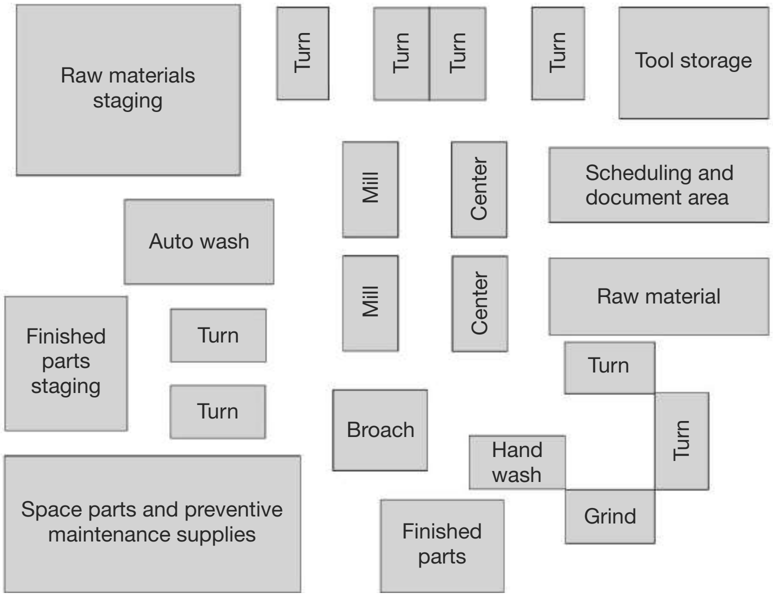


Fig. 22.27 *Completed cell with all the spaces required*

The actual machining time required for individual component is between 8 and 9 minutes with the total load and unload frequency of 2.5 minutes. In view of such small cycle time and the complexity of the operations involved, a major investigation is required. The major areas that were investigated are:

- Tool management
- Fixture management
- Raw material management
- Finished parts management
- Ring fitting facility
- Transport management

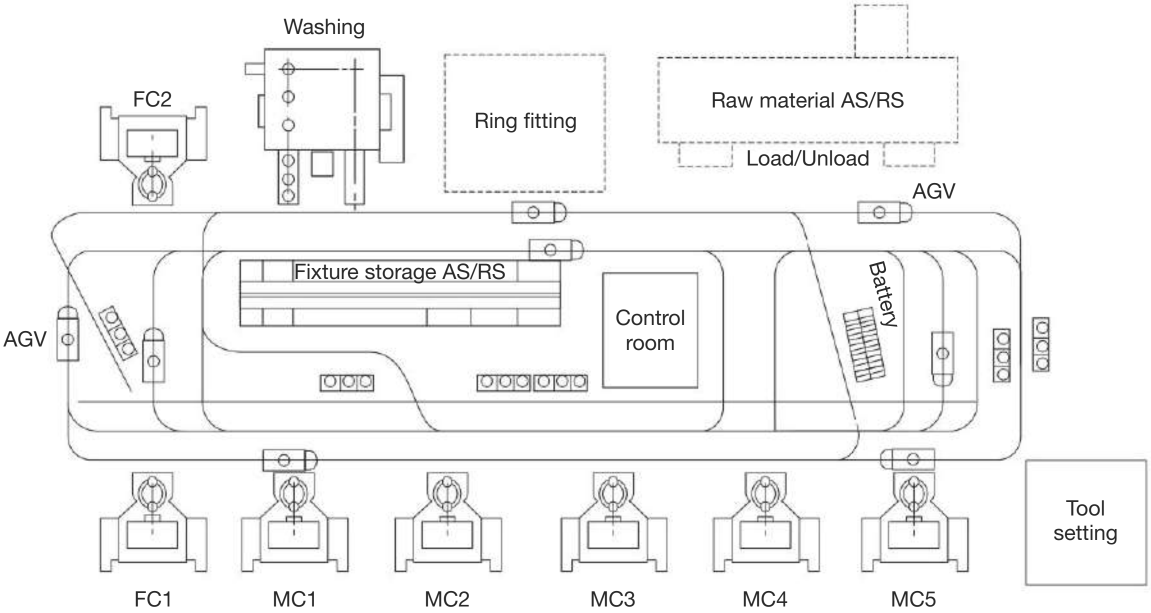


Fig. 22.28 *FMS for machining valve components by HNH*

Table 22.7 *Operation plan and routing of parts*

Part type	*Bonnet*	*Body A*	*Body B*
Operations	Load Machining centre Wash Unload	Load Machining centre Outfacer Wash Unload	Load Machining centre Wash Fit Outfacer Wash Unload
Total value	10%	65%	25%

Tool Management Initially rationalisation of the tools is carried out to reduce the variety of the tools required. Still the required number is large as follows:

- 275 for the machining centres
- 60 for the out facing machines

Tool changing has to be done very fast because of the short machining time of the individual tools as well as the flexibility of components coming to the machine. The machining centre has a tool magazine with 80 tool capability. External tool storage and supply is ruled out in view of the short machining time. As a result the tools are grouped such that not all jobs can be done on all machines. Though this restricts the flexibility was found to be cost optimum. This resulted in

- 75% jobs can be done on all machines
- 15% jobs can be done on certain machines
- 10% jobs can be done on one machining centre

Raw Material and Finished Parts Management Without a stacker crane, it is impossible to make the material available at the load stations required even with any number of operators. Hence an AS/RS is decided for raw material storage with 3 levels and 1 ton capacity. This will not clutter the load/unload stations and provides a response time of 68 seconds. The flow of material within the FMS is shown in Fig. 22.29.

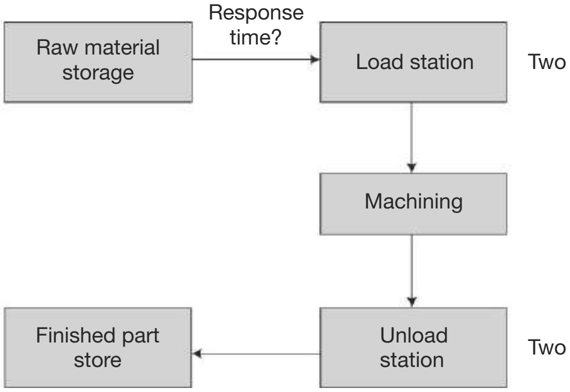


Fig. 22.29 *Movement of Raw Material in the FMS*

In order establish the feasibility of the system and develop the procedures, a simulation study was undertaken. The objectives of simulation that were set forth are

- determine feasibility of operation
- confirm assumptions made
- ascertain the effect of breakdowns
- establish the degree of flexibility
- to find the effect of changing parameters
- to find the total utilisation of the resources

Based on the simulation runs the following points were discovered which allowed for the modification of the system.

- Based on the capacity only 4 machining centres and 2 out facer machines are required. However when the simulation runs were made, the machining centres were very highly used upto 95% which will not give sufficient time for maintenance. Hence one more machining centre was added which brought the utilisation to about 85%.
- With a total of 7 AGVs in the system the total throughput was less. When they are increased to 8, the output increased by about 4%. A further increase to 9 has increased the output marginally and hence it is decided to use 8 AGVs.
- Similar study was done about the number of pallets in the system. As the number increased from 23 to 25 there is continuous improvement. However after that the improvement in output is only marginal.

The functions served by the online control are as follows:

- Update tool master data file
- Record components manufactured

- Record of components to be manufactured
- Record production status of workpieces
- Record position of pallets in the system .Provide error reports
- Download NC part programs as required
- Update NC part programs
- Download transport orders to AGVs
- Communicate with all equipment in FMS to organise work requirements for the sub systems
- Organise fixture stores
- Output instructions to load/unload stations.

Summary

- Flexible Manufacturing Systems (FMS) have been in vogue for the last 30 years and are responsible for bringing in the flexibility, which was not possible with the types of manufacturing systems that were practiced. FMS utilises computer controlled equipment to integrate all the functions of a manufacturing system such that any kind of flexibility desired can be achieved.
- Integration of all the equipment in an FMS brings the machine utilisation to the highest rate that is feasible.
- There are a large number of benefits that can be achieved by adopting FMS for a large range of manufacturing situations.
- Flexibility is a vague term and needs to be carefully defined depending upon the requirements. The types of flexibilities that could be considered are: Machine flexibility, Production flexibility, Mix flexibility, Product flexibility, Routing flexibility, Volume flexibility and Expansion flexibility.
- FMS consists of the manufacturing workstations, material handling equipment, tooling systems and the control equipment, each of them having various options available for a large range of applications.
- Workstations to be incorporated into an FMS should be as autonomous as possible.
- Tooling systems in terms of their capacity and ability to replenish quickly adds to the manufacturing flexibility of FMS.
- There are a number of layouts that should be considered for FMS such as inline layout, loop layout, ladder layout, and open field layout.
- FMS control systems will have to organise all the functions of operating the FMS into proper schedules. The things that need to be considered are production scheduling, tool management, transport management, production control, diagnostics, and maintenance planning.
- Development of an FMS cell requires complete information about the actual operations that are being carried out on all the parts that need to be manufactured in the cell. Gather and record data, Develop process flow within the cell, Identify equipment required and compare it to what is available, Select/assign equipment, Build a relation- ship diagram and an initial cell layout, and finalize the planned cell structure.
- Development of an FMS requires considerable amount of time and money to complete. An example of the development of FMS for machining valve components is shown to prove.

Questions

1. What are the various types of material handling and/or transportation systems used in FMS? Describe their individual domains (areas) of applications for which they are used with their relative advantages and disadvantages, if any.
2. What do you understand by the term flexibility in FMS? What are the various types of flexibilities that are relevant?
3. Explain briefly about production flexibility and routing flexibility in FMS.
4. Briefly explain the facilities that one should look for while selecting the following manufacturing equipment for an FMS.
    (a) Machining centres
    (b) Turning centres
5. Discuss the various tool supply (replenishment) concepts (from secondary tool store to primary tool store on the machine tool) used in the FMS context. While bringing out the specific advantages and applications, mention the most widely practiced principle in the industry.
6. What are the various types of layouts used in FMS design? Explain briefly about their applications.
7. Considering the current manufacturing scene in Asia (excluding Japan, Taiwan and Korea), what in your opinion should be the type of man turing facilities (Conventional, CNC, FMS, and ClM,) desired. Explain your choices (depending upon the scenarios considered) with detailed justifications. (*Hint: The discussion should be on the types of man turing needs of the region.*)
8. Differentiate the fixturing needs in FMS compared to conventional manufacturing. Your answer should clearly bring out the method of holdii well as other facilities available (types of fixturing systems) or desirable in FMS for fixturing compared to conventional manufacturing operations.
9. Show schematically an FMS system (hardware) giving the various control linkages and networking requirements for information flow. Your answer should be complete in terms of the actual control components present and the type of information flow taking place in the system between various nodes (elements).
10. What in your opinion is the minimum set of facilities required in a control software? Explain your answer with the help of a schematic. show the need for multi-tasking (time sharing) in the FMS control ware development.
11. Describe a typical fixture-pallet cycle's entry and exit into the FMS. Mention specifically the types of information that needs to flow from to these elements.
12. Explain the relevance of FMS in discrete part manufacturing in the modern era, in terms of various attributes that clearly show its application wide variety of possible manufacturing situations.
13. Conceptually show the organisation of information for tool manage in a FMS. How is this different from a manufacturing shop?
14. Define FMS, clearly showing the various desirable features that required for proper functioning, with reference to the current day manufacturing scene. Explain the functions served by these features.
15. Explain the situations in modern economies, the 'Economy of Scope' is more relevant, in comparison to the 'Economy of Scale' as has been practiced in the western world since the days of industrial revolution. What aspects of modem developments are calling for a change in the manufacturing approach?
16. In order to manufacture a number of variants of axle shafts for cars transmission (gear box) housings (hollow box shaped with number of faces and holes to be produced), describe the type of machine tools (with specifications) that would be required if an FMS is to be designed.
17. Explain the features you normally look in the machining centres to be suitable for including in FMS. Show how these features help with flexibility desired.

18. What are the various methods available for tool monitoring in a FMS? Explain their limitations with the one system most commonly used in the current practical FMS in the world.
19. Explain the relevance of FMS in discrete part manufacturing in the modern era, in terms of various attributes that clearly show its application in a wide variety of manufacturing situations possible.
20. What are the steps needed to develop the flexible manufacturing cells from scratch?
21. What type of data needed in order to develop a flexible manufacturing cell?

23

COMPUTER AIDED QUALITY CONTROL

Objectives

After completing the study of this chapter, the reader should be able to

- Understand the importance of CAQC in manufacturing
- Learn the differences between inspection and testing
- Understand various types of coordinate measuring machines and their applications
- Utilise non-contact inspection methods
- Learn about Statistical Quality Control (SQC) methods
- Understand Statistical Process Control (SPC) as different from SQC
- Learn about total quality management
- Learn different steps in Six Sigma approach
- Understand how CAQC will be integrated with other functions

23.1 INTRODUCTION

The success of a company in the market is identified to a great extent on the quality of products that it provides to the market. The product should satisfy the requirements of the consumer. The reliable performance of the product during its active service life depends to a great extent on the quality. Another aspect that is important for the organisation is the lead time, or time to market, as the product that comes first into the market garners a major share of the market.

Ideally, quality should be built into the product design and the processes that are used for its manufacture. However, quality can also be improved during the production stage. There are a number of tools available to the engineer to monitor the product quality and thus to control the process to ensure that quality products are automatically produced by it. The consequences of poor quality are disastrous for the companies, since it causes unreliable performance, thereby denting the reputation of the company forever. In clear contrast is the performance of a few companies such as Sony, Toyota, etc., as can be perceived by the people.

Also with the developments in the micro and nanotechnology fields, the sizes of parts produced have come down to micron and nanometre sizes. Thus, the tolerances also will accordingly be coming down to those values. In view of this, the standard for length measurement needs to achieve much better definition. Hence, the International Standards Organisation (ISO) has adopted a new definition for 'metre' in terms of the distance travelled by light in 1/ 299 792 458 of a second, as the measurement of time is much more accurate than other parameters.

The use of statistical techniques championed by Deming have really made an enormous change in perceiving the quality control and assurance techniques as a means of ensuring that the given quality is achieved by adjusting the process itself. In later years, Motorola and General Electric pioneered the Six-Sigma methodologies as a comprehensive philosophy of designing quality into the system. All this calls for continuous measurement of quality and taking corrective action also on a continuous basis. A number of sensors and systems have been developed that can monitor quality continuously with or without the assistance of the operator. Computer Aided Quality Control or CAQC is a broad range of services in the manufacturing organisation which go beyond the simple inspection process. A block diagram is shown with typical inputs and outputs into CAQC in Fig. 23.1. This chapter discusses the various concepts and methodologies that are required to implement CAQC in a manufacturing organisation.

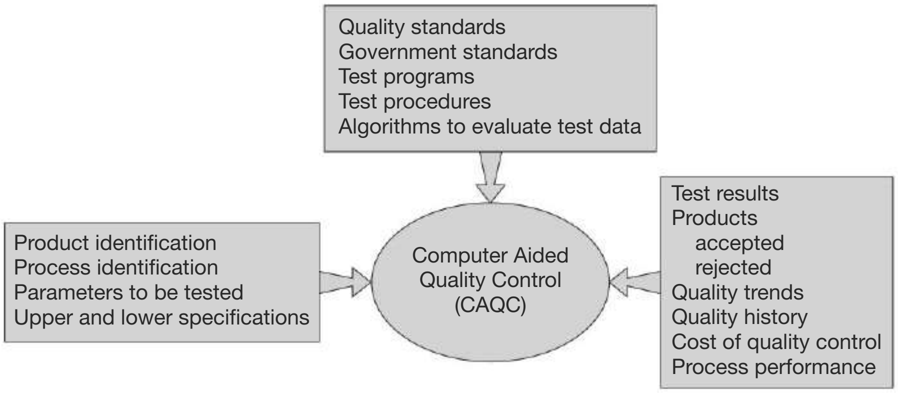


Fig. 23.1 *Typical activities in Computer Aided Quality Control*

23.2 INSPECTION AND TESTING

Inspection is the term that is normally associated with examining a given product and check whether it conforms to the specifications for that particular product. For an engineering component, this means checking for various dimensions, tolerances—both dimensional and form—and the surface integrity characterised by the surface finish and other parameters that are associated with it. If there is any non-conformance noted at this stage, the product will either be completely rejected or sent for reworking depending upon the extent of non-conformance. Either of these actions will put enormous cost to the system in terms of rescheduling the parts, additional part movement, allocation of additional resources, etc. In addition to the testing of in-house produced parts, there are large numbers of items that are brought into the organisation from outside, either from the vendors or subcontractors. These are the raw materials which are used for further processing in the shop, semi-finished goods that are finished in-house, standard items such as fasteners, bearings, etc., and subcontracted items. All these items, with the exception of a few, need to be inspected 100% at the point of entry into the system to ensure that no defects are entering into the system.

Also, testing of finished products according to the laid-down standard procedure is an important element in the product-manufacturing sequence. The actual testing carried out on the product depends upon the final function it is supposed to serve. For example, if it is a fluid-carrying pipe then it should be checked to see whether it is leak-proof when fluid at the designated pressure is flowing through it. For this purpose, normally standard bodies such as BIS (Bureau of Indian Standards) lay down the test procedures which have to be adhered by the manufacturer. Sometimes, the testing may have to follow an accelerated or simulated testing to conform to the final service conditions. The proving grounds used to test automobiles to prove their safety is an example in this category. Similarly, the accelerated tests on computer hard disks to obtain the Mean Time Between Failures (MTBF) under actual working environment are another example for testing. In the same way, the assembled personal computers are torture tested for 24 hours in extreme environments to make sure that under normal working conditions, they are able to withstand momentary variations beyond the designed parameters. All these testing procedures ensure that the product performs well with the final user.

Tests that were described so far are all non-destructive in nature and are normally conducted 100% on the products. Once they pass the tests, they are packaged and sent to marketing. Sometimes, the standards may require certain destructive tests to be conducted that are associated generally with the safety of the product. For example, the destructive tests done on automobiles where an actual car goes through a simulated car crash to prove its safety standard is an example in this category. The test completely destroys the product. Similarly, the incoming raw materials can also be tested for their conformance to standards. This form of test being very expensive, is only done on a sampling basis.

23.3 COORDINATE MEASURING MACHINE

One of the very important developments in dimensional metrology is the versatility in contact-type inspection provided by the Coordinate Measuring Machines (CMM). CMM is a flexible measuring device capable of providing a highly accurate dimensional position along three mutually perpendicular axes. This equipment is extensively used in the manufacturing industry for post-production inspection of a large variety of components and assemblies in regular production. It can also be used for checking the dimensions of various tooling such as jigs and fixtures, dies and moulds, special tools, etc.

Any basic coordinate measuring machine has the following major elements in order for it to properly function in a measuring environment:

- ***High Precision Mechanical Structure*** It is the main operating unit of a CMM which has the ability to move the sensor probe to be positioned at the requisite location in X-Y-Z on the part to be measured. This structure has to be very rigid to provide the necessary accuracy for the measurements. It can be controlled either manually or be CNC controlled for precise positioning of the probe (position sensor) element on any point of its working volume with a very high repeatability. Typical accuracies are of the order of $(1.7 + 3L/1000)$ μm. For higher accuracy, the drive axes are provided with linear glass scales. For faster motion and minimising vibration during high-speed, high-acceleration travel, air bearings are provided for the linear axes. These are located at the bottom face, as well as at the front, rear, and upper surfaces of the slider unit to enable stable linear motion. To provide a thermally stable surface for part placement, granite tables are generally used as bases in most of the CMMs.
- ***Probing System*** The touch-trigger probe is used to collect the basic information such as x, y, z coordinates of the points where the probe touches the part to measure a particular dimension. It is a sophisticated mechatronic device capable of allowing the CMM to record a number of points on the surface of the part, which need to be quickly measured. It is also possible to have an optoelectronic sensor for non-contact inspection in place of the touch-trigger probe.

- ***Machine Control Unit*** It is essentially a computerised system that serves the purpose of controlling the probe positioning as well as getting the data in terms of the measured point sets which are then utilised by the software for further presentation.
- ***CMM Software*** The major component of a CMM is its metrology operating system, which allows it to do the necessary computations on the point sets that were measured and communicate to other equipment in the system. It consists of a number of application software depending upon the industry for the purpose of special analysis such as comparing with CAD data and evaluating the deviations, evaluating special profiles such as those used in gears and gas-turbine blades, etc.

The total measuring envelope of CMM is defined by its maximum *X*, *Y*, and *Z* travel. A schematic diagram of a coordinate measuring system with all its parts as described above is shown in Fig. 23.2.

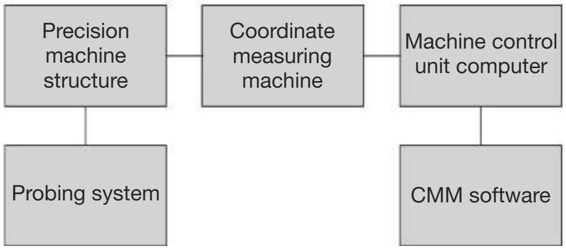
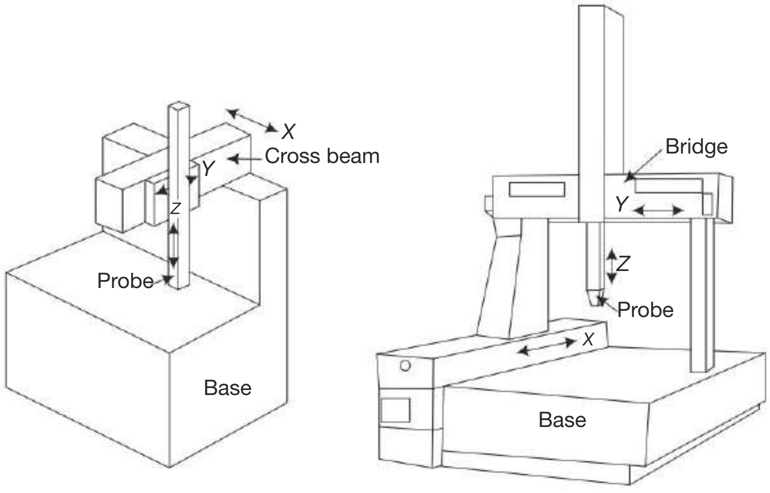


Fig. 23.2 *Typical elements in a Coordinate Measuring Machine*

23.3.1 Types of Coordinate Measuring Machines

There are a number of different types of CMMs that are used in the industry. They can be broadly classified into five categories:

- ***Cantilever Type*** In this configuration, the probe is attached to a vertical ram (indicated by *Z*-axis in Fig. 23.3a) which moves on a cantilever beam (*Y*-axis). The cantilever beam moves along the *X*-axis which is perpendicular to both *Y* and *Z*-axes. The cantilever-type CMM is normally limited to small and medium machines. This configuration provides completely unobstructed work area allowing for easy access to the operator and ability to load parts that are longer than the machine table. Since the cantilever is not supported at the other end, accuracy may sometimes be affected unless some kind of compensation is built into the system.
- ***Bridge Type*** In this configuration, the probe is attached to a vertical ram (indicated by *Z*-axis in Fig. 23.3b) which moves on a horizontal beam (bridge structure) along the *Y*-axis. The horizontal beam moves along the *X*-axis on guideways on the fixed table and carries the probe. There are many other variants possible such as a bridge structure fixed to the table while the workpiece moves (*X*-axis) on a moving table on the base of the CMM. This configuration is referred to as a fixed-bridge CMM. The bridge-type configuration provides for more rigid construction which ensures high accuracy, and it is the most popular configuration. The presence of a bridge on the machine base makes it more difficult to load parts larger than the width of the distance between the two sides of the bridge.
- ***Column Type*** This type has a moving table and saddle arrangement similar to a horizontal boring machine as shown in Fig. 23.4(a). The column moves only along the *Z*-axis while the *X* and *Y* motions are provided by the 2-axis saddle.
- ***Gantry Type*** These are large machines for measuring very large parts such as aircraft fuselages and automobile bodies. The probe quill moves vertically along the *Z*-axis while the carriage on which the quill is mounted moves along the *Y*-axis on the cross beam as shown in Fig. 23.4(b). The cross beam

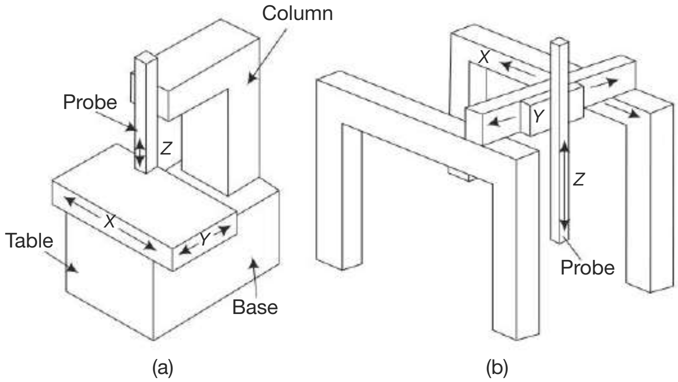


(a) (b)

Fig. 23.3 *Coordinate Measuring Machine types (a) Cantilever type (b) Moving-bridge type*

moves along the *X*-axis and is supported on two elevated rails attached to columns fixed to the floor. This configuration, besides being open, has less inertia of the moving parts while providing structural stiffness.

- ***Horizontal-arm Type*** The probe attached to the horizontal arm moves along the *Y*-direction, while the arm itself moves in the *Z*-direction with the column support as shown in Fig. 23.5. The column moves on the guideways along the *X*-axis. This configuration is useful for measuring large parts. The other variation in this type is with a moving table which provides the *X*-motion.



(a) (b)

Fig. 23.4 *Types of Coordinate Measuring Machines (a) Column type (b) Gantry type*

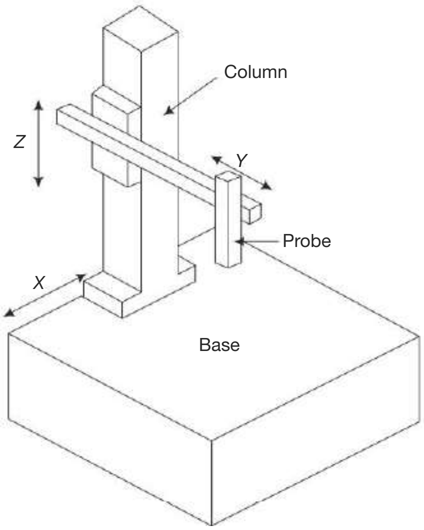


Fig. 23.5 *Fixed horizontal-arm type Coordinate Measuring Machine*

23.3.2 Types of Probes Used

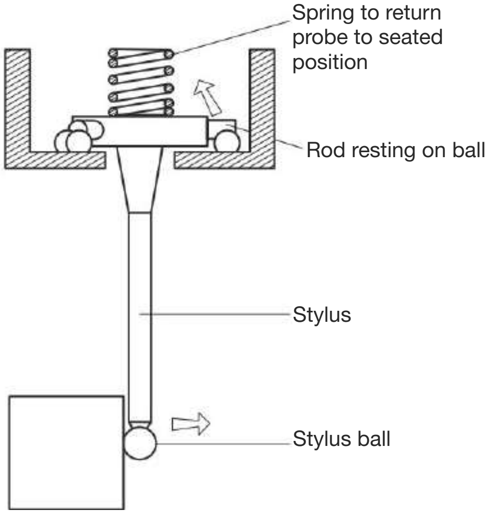


Fig. 23.6 *Schematic of touch-trigger probe internal construction (redrawn from a Renishaw presentation)*

The touch-trigger probes used are same as those described in Chapter 12. A touch-trigger probe consists of a spherical sapphire unit connected elastically to a rigid measuring unit as shown in Fig. 12.43. When the probe moves along a direction and touches a surface, the elastic connection deflects and triggers the measuring system. The trigger is basically an omni-directional switch capable of detecting deflections in any direction. The schematic of the internal construction of a touch-trigger probe is given in Fig. 23.6. Three rods are connected to the stylus which rest on two balls, thus providing 6 points of contact for locating the stylus in its correct position. At a time when the stylus makes contact with the component, the stylus ball is slightly deflected. This is manifested as bending of the stylus and the stylus assembly pivots about the kinematic contacts, resulting in one or two contacts moving apart. This generates a trigger and stops the machine measuring the point of contact. Then the machine backs off and the spring ensures that the stylus is returned to its seating position. From that the physical position of the surface where the probe contacts gets recorded. Since the probe tip is spherical, the contact between the probe and the measured surface is a point contact. The controller automatically compensates for the radius of the probe.

A large variety of probes are used with CMM to cater to the variety of applications. Typical sets of probes used are shown in Fig. 23.7. Ruby stylus balls are the most commonly used material in the industry because they are very hard and wear-resistant. In addition, sometimes balls made of silicon nitride or zirconiums are used for heavy-duty applications. The stylus stem is generally made of non-magnetic stainless steel. This is normally used for all styli with a ball diameter greater than 2 mm and a length of up to 30 mm. For a small ball diameter that is less than 1 mm or if the stylus is up to 50 mm long, tungsten carbide is useful. Sometimes, a ceramic stylus offers the advantage of low weight compared to steel. For very long stylus lengths (up to 800 mm), a carbon-fibre stylus is used.

In general, it is good to keep the stylus as short as possible for the given application, for better accuracy. The stylus ball size used should be as large as possible that helps in minimising the component surface finish effects. Also, when the stylus length is to be increased, use minimum number of joints in the interest of accuracy.

A special class of probes are the *indexable probe heads* which permit orienting the measuring probe in horizontal and vertical planes depending upon the surface being measured. This helps the CMM reach the surfaces that are otherwise difficult to reach. These are generally provided with a power drive and control, thus adding additional axes to the CMM. However, these are expensive and also reduce the overall size of the part that can be measured.

23.3.3 Control Systems

Coordinate Measuring Machines can be either manual or computer controlled. In the manual machines, the operator moves the probe manually to the desired location and instructs the computer linked with the CMM to record the point of interest. Normally, the processing of the information is done by a dedicated computer that

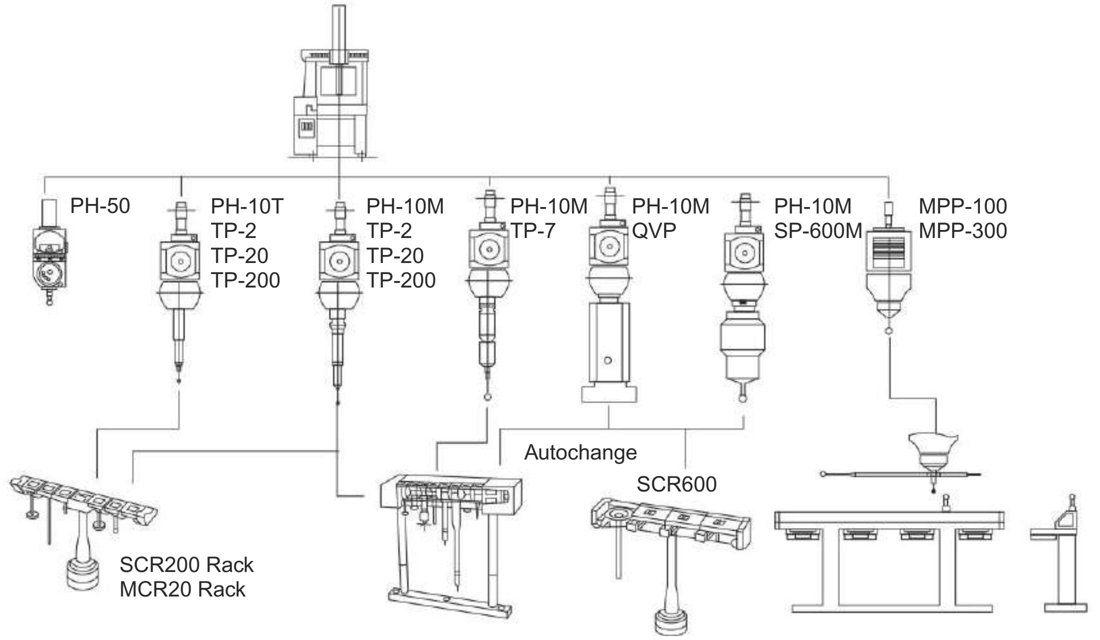


Fig. 23.7 *Different types of probes used in Coordinate Measuring Machine (Courtesy, Mitutoyo)*

has the necessary metrological data-processing software. Most of these systems have reasonably complex routines for the dimensional data evaluation required for manufacturing industries.

In contrast, a CMM with CNC control is most versatile, since the machine can be run with a specified part program like any other CNC machine tool. Since the controller is powerful, it is also capable of carrying out most of the data-processing operations required to evaluate the measurement process. The inspection programs can be prepared using an off-line programming system for operating a CMM with CNC control.

The use of a computer control provides a lot of flexibility to the user. The resident software enables a simple procedure for automatic work alignment which is otherwise time-consuming, skill-oriented and subjective in the case of manual inspection. For example, by touching the surface with the probe at least at three points, the plane can be determined, this being a very convenient and fast method for establishing a reference plane. Similarly, a line or an edge can be defined by touching at two or more points. A circle can be measured by making the probe touch the circle (say, a hole or a shaft) at three or more points so that the centre and diameter of the feature can be automatically calculated.

Many CMM manufacturers offer a suite of data-processing software built into the operating computer of the CMM. Many of these software suites have most of the common geometric function evaluations such as points, lines, planes, circles, cylinders, spheres, and cones. They are also able to evaluate geometric tolerance conditions by determining various types of form and positional relationships (such as flatness, straightness, circularity, parallelism, or squareness) for single features and related groups of features. The range of geometric features available from Mitutoyo's MCOSMOS software is given below:

PartManager The control centre from where the software system is started and part programs are managed

Geometry For online and offline part-program creation using measurements of workpiece features; includes extensive tolerance comparisons and output functions

Online/offline Programming For online and offline part-program creation using measurements of geometric elements directly from the CAD model, with automatic collision avoidance

3D Freeform Surface Evaluation CAD model-based generation of surface-measurement points, and comparison of actual/ nominal data, with graphical output

2D Profile Evaluation Combines scanning and evaluation of workpiece contours and 3D digitisation of surfaces

Statistics Evaluation Real-time collection, analysis and display of SPC data; networking and web-sharing capabilities

Dimensional Interface For bi-directional transfer of measurement programs, in DMIS standard format, between MCOSMOS and various external systems

Aerofoil Evaluation Analyses aerofoil surface profiles, such as turbine blades, from scanned data

Involute Gear Profile Evaluation For the measurement of all kinds of involute profiles and subsequent comparison with factory or international standards

NC Correction Fully automatic machine-tool correction using optimised data from multiple part measurements

Queuing System Executes a list of existing part programs consecutively at pre-defined positions on the measuring table to allow unattended operation when measuring multiple parts.

23.3.4 Non-contact CMM

The touch-trigger probe used in the CMM can be replaced by non-contact probes such as laser probe or digitising camera for inspecting the parts in a non-contact manner. Because of the non-contact nature of operation of these probes, fast and accurate measurements can be obtained. Optical probes such as an area-projection microscope and a centring microscope can be used for inspecting drawings, printed circuit boards, and small, fragile workplaces. In a laser light probe, laser is used to bounce off the surface of the part sensed by the sensor and the surface location is obtained by optical triangulation. Most of these are generally useful for planar dimensions only. A laser probe can also be used for scanning three-dimensional surfaces. Some of the parts that are measured on non-contact CMMs are IC packages, small and tiny components, printed circuit boards, connectors, etc.

23.3.5 Advantages of CMM

There are many advantages to be gained by the use of CMMs, some of which are given here.

- The CMM is a highly flexible measuring machine. It can be practically used for a variety of measurement activities. It not only measures the dimensions, but various forms and contours without needing any fixture or clamping.
- The time required for inspection gets reduced using a CMM since there is no need to set up the part or establish a reference point for inspection purpose. The datum planes are defined on the part located on the CMM table and all other measurements are made with reference to those datum planes. Also, many CMMs have the capability to inspect all dimensions and forms in a single set-up.
- Since the measurements are done with respect to a single datum, the accuracy of measurements is high eliminating any accumulated errors which are a problem with the conventional measurements carried with different datum surfaces and set-ups.
- Operator error is minimised. Since CMM uses direct reading of the measured points electronically, there is no subjective interpretation and in the process, any operator error is eliminated.

- Operator skill is virtually eliminated. The canned software in the CMM measurement operating system has built-in procedures for typical part features, such as bores or centre distances. As explained earlier, once the datum is set by the operator, the rest of the measurements are automatically recorded with respect to that datum under the control software. The CNC versions of the CMM can even run without the operator. Also, all the data and report is completely generated under the control of the measuring program running for the particular part allowing for less skilled operators to perform relatively complex inspection procedures.
- Productivity increases with the use of a CMM. This is achieved because the set-up time is reduced, the time for measuring the data is reduced as also the time taken for generating the final inspection report. All this contributes to the overall productivity improvement.
- No offline analysis is to be done for inspection reports, since everything is completed by the computer that is linked with the CMM.
- Since no inspection fixtures are used with CMM, there is a cost saving associated with this.
- Because of the reliability of the measuring procedure with CMM, there is a reduction in good-part rejections.

23.4 NON-CONTACT INSPECTION METHODS

In non-contact inspection, the part is not physically contacted by the measuring instrument. As a result, this process is faster compared to contact inspection. Some parts are too complex or fragile for traditional measuring methods. The part surface is not affected in any way with the inspection process in non-contact inspection. Also, since the probe does not physically contact the surface, there is no wear and tear of the inspection probe. In some methods, there is no need to orient the workpiece and it can be measured as placed in the position. Some of the areas where non-contact inspection finds application are as follows:

Dimensional Dimensions, shape, positioning, orientation, alignment, roundness, etc.

Structural Assembly (holes, slots, rivets, screws, clamps) Foreign objects (dust, burr, swarf)

Surface Pits, scratches, cracks, wear, finish, roughness, texture, seams-folds-laps, continuity

Operational Non-conformance of operations to specifications

23.4.1 Vision Systems

Industrial automation benefits greatly by the use of machine vision. A number of industrial processes have actually benefitted from vision systems. A few examples are delicate electronics component manufacturing, metal-product finishing, machine parts, and Integrated Circuits (IC) manufacturing. The use of machine-vision technology improves productivity and quality management and provides a competitive advantage to industries.

Processing of information in a visual-inspection system has to follow the following steps:

Image Acquisition The actual images of the parts are acquired in digital form through cameras, digitisers, etc.

Image Processing These acquired images are filtered to remove background noise or unwanted reflections from the illumination system. Some amount of image restoration may be done to improve image quality in case of the introduced geometric distortions by the acquisition system (e.g., the camera).

Feature Extraction The next step is to extract the features from the image. The image features are extracted by taking care of the non-overlapping or uncorrelated features, so that better classification can be achieved.

Size, position, contour measurement via edge detection and linking, as well as texture measurements on regions are some examples of features extracted.

Decision-making From all the features found in the previous step, combine the features that are relevant for the given application. The reduced feature set is processed further so as to reach a decision based on the type of application. For example, during production, the system decides if the produced parts meet some quality standards by matching a computed description with some known model of the image (region or object) to be recognised.

A typical machine-vision system is shown in Fig. 23.8. The scene containing the parts needs to be appropriately illuminated and arranged in order to facilitate the reception of the image features necessary for processing and classification. A camera placed above the parts at a proper angle will acquire the image of the parts that are to be inspected. The cameras are normally fixed and have sufficient field of view to cover all the parts that come into their view. The features to be inspected are known in advance. Industrial automation systems are normally designed to inspect only known objects at fixed positions.

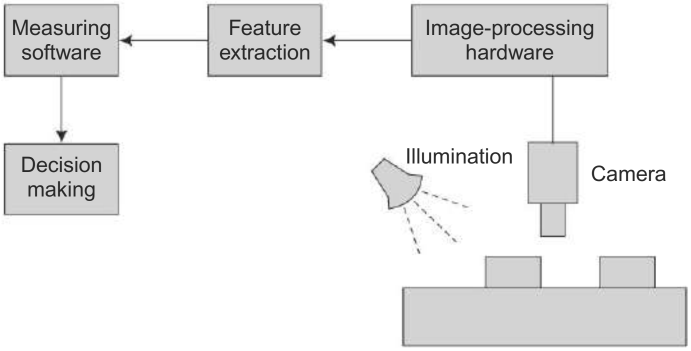


Fig. 23.8 *Typical vision-system components and procedure*

Image-processing hardware is essentially a computer that is embedded with the necessary image-acquisition hardware board (frame grabber). Because of the acquisition and image-analysis process being highly time-constrained or computationally intensive, the processing power of the main processor may not be sufficient. The acquisition-board hardware therefore consists of application-specific hardware such as Digital Signal Processing (DSP) and Field Programmable Gate Arrays (FPGAs).

Once the images are identified and measurement completed, these are used to

- Control a manufacturing process (e.g., for guiding robot arms placing components on printed circuit boards, painting surfaces, etc.)
- Propagate to other external devices for further processing (e.g., classification)
- Characterise defects of faulty items and take actions for reporting and correcting these faults and replacing or removing defective parts from the production line

23.5 STATISTICAL QUALITY CONTROL

Statistical quality control is controlling the quality of products using statistical methods. The dimensional measurements are expected to follow the normal distribution. For a number of measurements (x_i), the standard deviation (σ_x) can be calculated as

$$\sigma_x = \sqrt{\frac{\Sigma\,(x_i - \bar{x})^2}{n}}$$

where n = number of data points (usually called the population)

$\bar{x}$ = average (usually called the population mean)

In normal distribution, 68.26% of all data points lie within ± 1σ of the mean, 95.46% of all data points lie within ± 2σ of the mean and 99.73% of all data points lie within ± 3σ of the mean.

Control charts are the most effective tools in SQC to determine the variability of a given process. Since the process dimensions are expected to follow normal distribution, 3σ is considered as the limit for controlling the process. The control chart will have a nominal value that the manager would like to control with specified Upper Control Limit (UCL) and Lower Control Limit (LCL). As long as the process is within these limits, it is considered to be stable. When the values fall outside these limits or any abnormal variation occurs within the limits, it gives indication to the manager that there is some problem with the process that needing attention. Two charts are used together: *R*-chart ('range chart') and *X*-bar chart ('average chart'). Both the process variability (measured by the *R*-chart) and the process average (measured by the *X*-bar chart) must be in control before the process can be said to be in control. Process variability must be in control before the X-bar chart can be developed because a measure of process variability is required to determine the $\overline{X}$-chart control limits. The functions served by the control charts are

- detecting special causes of variation of the process,
- measuring and monitoring common causes of variation in the process,
- identifying the time to look for problems and adjust, and
- identifying the time to make a fundamental change in the process.

***R*-Charts** A range chart or *R*-chart is used to monitor the variability of the production process. Range is the difference between the highest and lowest data points in a group. To calculate an *R*-chart, subtract the smallest measurement from the largest measurement in each of the subgroups to get the *R* value. The control limits for the *R*-chart are

$$\mathrm{UCL_R} = D_4\,\overline{R}, \quad \text{and} \quad \mathrm{LCL_R} = D_3\,\overline{R}$$

where, D_3 and D_4 are constants to provide three times the standard deviation limits for a given subgroup size (See Table 23.1), and

$\overline{R}$ is the average of R values of all the subgroups under consideration.

$\bar{x}$-Chart An $\bar{x}$-chart is used to measure the mean values of the observations. The control limits (UCL and LCL) are given by $\bar{\bar{x}} \pm 3\,\bar{s}_x$, where $\bar{s}_x$ = average standard deviation. That is, for each subgroup of data (e.g., each day's data) a standard deviation is calculated (s_x). The average standard deviation ($\bar{s}_x$) is the arithmetic average of s_x for all subgroups. A simpler way of calculating the control limits is

$$\mathrm{UCL}_X = \bar{\bar{x}} + A_2\,\overline{R}, \quad \text{and} \quad \mathrm{LCL}_X = \bar{\bar{x}} - A_2\,\overline{R}$$

where $\bar{\bar{x}}$ is the average of all the subgroups or a target value set for the process. This is used to draw the centre line of the chart, and

A_2 is constant to provide three times the standard deviation limits for a given subgroup size (See Table 23.1).

The application of *R*-charts and $\overline{X}$–charts is explained with the help of an example below:

Table 23.1 *Constant factors for calculating R-chart and $\bar{x}$-chart*

Subgroup sise	*Factor for UCL and LCL for $\bar{x}$-charts* A_2	*Factor for LCL for $\overline{R}$-charts* (D_3)	*Factor for UCL for $\overline{R}$-charts* (D_4)
2	1.880	0	3.267
3	1.023	0	2.575
4	0.729	0	2.282
5	0.577	0	2.115
6	0.483	0	2.004
7	0.419	0.076	1.924
8	0.373	0.136	1.864
9	0.337	0.184	1.816
10	0.308	0.223	1.777

Example 23.1 A set of 20 observations for a given process measurement are given in Table 23.2. Develop the control charts for this data.

Table 23.2 *Data for Example 23.1*

Subgroup number	*Observation, mm*			
	1	*2*	*3*	*4*
1	12.736	12.756	12.723	12.769
2	12.753	12.804	12.761	12.751
3	12.746	12.766	12.789	12.758
4	12.720	12.786	12.761	12.738
5	12.804	12.842	12.786	12.819

Step 1 From the given data (small sample to help with understanding the procedure), compute the column for *R* by subtracting the smallest value of the observation from the largest value in that particular row. Then take the mean (average) of all *R*'s as $\overline{\mathbf{R}}$ as shown in Table 23.3.

Table 23.3 *Computations for Example 23.1*

Subgroup number	*Observation, mm*				
	1	*2*	*3*	*4*	*R*
1	12.736	12.756	12.723	12.769	0.046
2	12.753	12.804	12.761	12.751	0.053
3	12.746	12.766	12.789	12.758	0.043
4	12.720	12.786	12.761	12.738	0.066
5	12.804	12.842	12.786	12.819	0.056
				Average	**0.0528**

Step 2 To construct the range chart, taking the appropriate constants from the Table 23.1, calculate the UCL and LCL.

$$UCL_R = D_4\,\overline{\mathbf{R}} = 2.282 \times 0.0528 = 0.1205 \text{ mm}$$

$$LCL_R = D_3\,\overline{\mathbf{R}} = 0 \times 0.0528 = 0 \text{ mm}$$

Step 3 From the data calculated, plot the range chart as shown in Fig. 23.9. As can be seen since all the points are within the limits, the process is in statistical control. If any of the points are outside then take fresh samples and repeat the exercise.

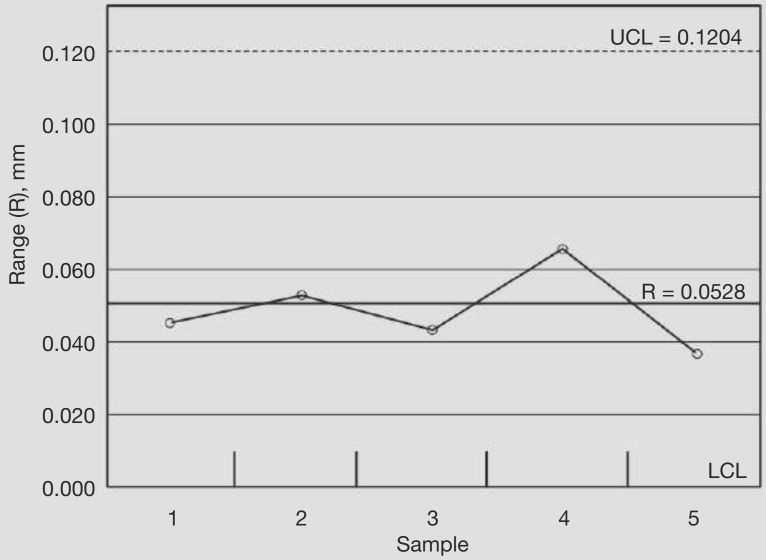


Fig. 23.9 *Range chart or R-chart for Example 23.1*

Step 4 From the data, calculate the mean of all values in each of the subgroup, and then calculate the mean of the means as shown in Table 23.4. Take the appropriate constants from the Table 23.1, and calculate the UCL and LCL.

$$UCL_X = \overline{\overline{x}} + A_2\,\overline{R} = 12.7685 + 0.729 \times 0.0528 = 12.807 \text{ mm}$$

$$LCL_X = \overline{\overline{x}} - A_2\,\overline{R} = 12.7685 - 0.729 \times 0.0528 = 12.730 \text{ mm}$$

Table 23.4 *Computations for Example 23.1*

Subgroup number	*Observation, mm*					
	1	*2*	*3*	*4*	*R*	$\overline{X}$
1	12.736	12.756	12.723	12.769	0.046	12.746
2	12.753	12.804	12.761	12.751	0.053	12.767
3	12.746	12.766	12.789	12.758	0.043	12.765
4	12.720	12.786	12.761	12.738	0.066	12.751
5	12.804	12.842	12.786	12.819	0.056	12.813
				Average	**0.0528**	**12.7685**

Step 5 From the data calculated, plot the $\overline{X}$-chart as shown in Fig. 23.10. As can be seen, the mean of the subgroup 5 goes beyond the UCL, indicating the possibility of the process is out of control in that subgroup, and therefore needs to be further explored for the cause. Some of the causes to examine are the poorly trained worker, fluctuations in operation of the machine tool and normal variation of the raw material used.

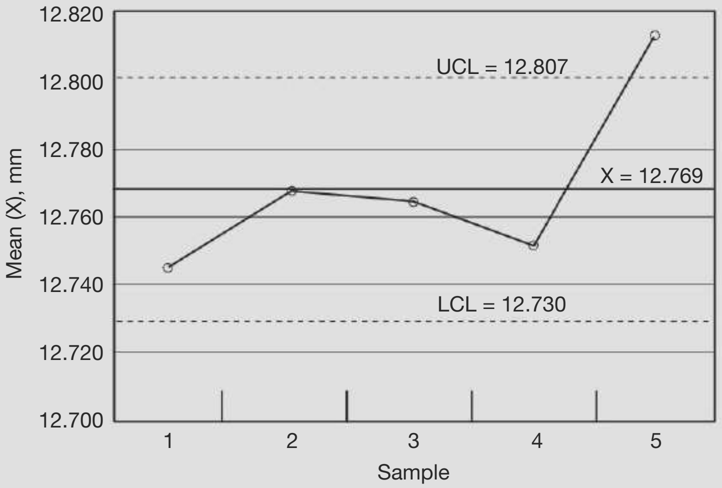


Fig. 23.10 *X-bar chart for Example 23.1*

If a plotted point falls within the control limits then the process is assumed to be in control at that moment of production. If it falls outside the control limits then the process is said to be out of control at that moment and the presence of some assignable cause is indicated. Sometimes, even though all the points are inside the control limits, indication of trouble or presence of assignable causes is evidenced from unusual patterns of points, such as a predominantly large number of points on one side of the central line, a series of points all falling close to one of the control limits or a series of points exhibiting a trend. A run is defined as a sequence of similar type of observations. Obviously, a run of 8 or more points has a very low probability of occurrence in a random sample of points. So, any type of run of 8 or more points is often taken as a signal of an out of control.

Other controls that are used in SQC are

- control chart for standard deviation (s-chart)
- control chart for number of defects (C-chart)
- control chart for percent defective (100p-chart)
- control chart for number of defectives (np-chart)

23.6 STATISTICAL PROCESS CONTROL (SPC)

Statistical process control is an extension of SQC in that it utilises the prevention approach compared to SQC which utilises the detection approach. As a result, SPC concentrates on the process rather than the product. It should be noted that a product is of the required quality because it is manufactured properly and not because it is inspected. Some goals for SPC can be identified as

- Improve quality and reliability of products and services without increased cost
- Establish measurement and verification system on an ongoing basis
- Provide a systematic way to direct efforts in problem solving

Variation will always be present in all the processes that are used in the industry. SPC identifies changes between products being produced over a long period. Corrective action, therefore, is applied before any defective material is produced. A process is in statistical control when all the ‘assignable causes’ have been eliminated and the only source of variation is the natural process variation. As explained above, out-of-control conditions are quickly identified by using control charts. SPC logically identifies responsibilities and accountabilities, and eliminates ‘finger pointing’ and confusion. Thus, it puts in place an efficient system to correct problems in the production area so that there are fewer tendencies to hide or ignore problems. Benefits of using SPC can be outlined as

- Prevention of defects rather than detection of errors as in SQC
- Greater machine up-time
- Avoidance of unnecessary capital expenditure on new equipment
- Less warranty costs
- Increased ability to meet production delivery dates
- Increased productivity

23.7 TOTAL QUALITY MANAGEMENT (TQM)

The main goal of Total Quality Management (TQM) is to satisfy the customer. TQM is a philosophy to guide an organisation to continuously improve itself by following a set of guiding principles. TQM can be defined as systematic and efficient operation of activities in the whole organisation to supply quality goods and services to customers at the right time and at the right price. It should be realised that a customer does more than simply take delivery of the goods bought. Experience with offers, services, telephone calls, presentations and invoices also influence the customer’s satisfaction. Satisfied customers increase the sales which in turn improves the cash flow. To turn cash flow into profit, quality cost needs to be minimised. It should be realised that inspection is not the way to improve quality while cutting quality cost. Controlling the manufacturing process saves the extra cost of inspection, rework and scrap.

TQM requires

- Committed management for long-term organisational support
- Focus on customers, both internally and externally
- Effective involvement of the entire workforce
- Continuous improvement in business and production processes
- Treating suppliers as partners
- Established performance measures for the processes

TQM requires a cultural change. The total scope of the TQM is shown in Fig. 23.11. It is a long-term commitment from the management to improve the overall operation of the organisation. It has many tools at its disposal which are given in Fig. 23.11, the discussion of which is beyond the scope of the book.

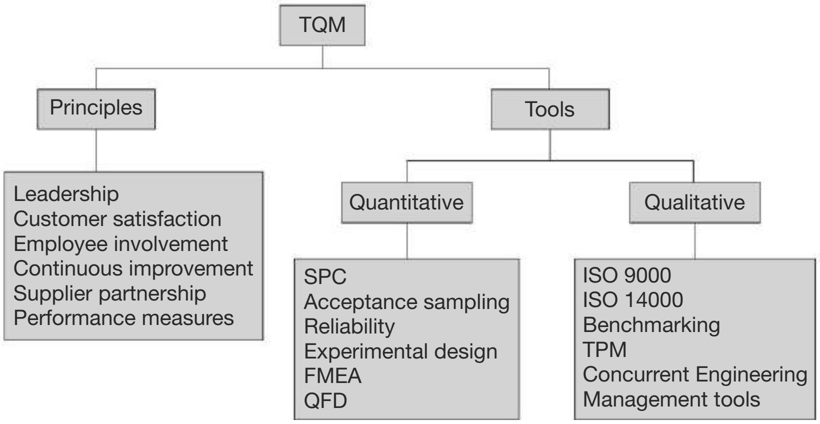


Fig. 23.11 *Total TQM activity in an organisation; FMEA–Failure Mode and Effects Analysis, QFD–Quality Function Deployment, TPM – Total Productive Maintenance*

TQM has been promoted vigorously by Deming through his fourteen points:

1. Create constancy of purpose toward improvement of product and service.
2. Adopt the new philosophy. We are in a new economic age.
3. Cease dependence on inspection to achieve quality.
4. End the practice of awarding business on the basis of initial cost.
5. Improve constantly and forever every activity.
6. Institute training and education on the job, including management.
7. Institute leadership.
8. Drive out fear.
9. Break down barriers between departments.
10. Eliminate slogans and exhortations.
11. Eliminate work standards that prescribe numerical quotas.
12. Remove barriers that rob workers of their right to pride of workmanship.
13. Institute a vigorous program of education and self-improvement.
14. Put everybody in the company to work in teams to accomplish the transformation.

Key requirements of the TQM process are

- Everyone must understand quality and of the need to change
- Management must create the environment for continuous improvement
- Management must provide the resources to support the improvement process
- Everyone must contribute to the end product or service used by the customer

Some of the problems noticed with the implementation of TQM are the following:

- TQM seems to lack focus.
- It has large number of tools and techniques that can be applied to improve quality. But the rate of visible improvement is slow for the effort expended.

23.8 SIX SIGMA

Six Sigma is a process improvement methodology developed at Motorola in the 1980's to reduce defects in its processes. Its goal was to achieve a level of performance equal to a defect rate of 3.4 defects per million. Six Sigma uses a powerful framework (DMAIC) and statistical tools to uncover root causes to understand and reduce variation. Although Six Sigma eliminates the defects in the process through in-depth statistical metrics used to analyse quality at all levels of the supply chain, it does not address the question of how to optimise process flow. The two main assumptions in Six Sigma are

- people in an organisation understand and appreciate the fact that numbers can represent features and characteristics of a process, and
- through the reduction of variation of all the processes, the overall performance of the organisation can be improved.

The DMAIC model for process improvement is used as the basis for Six Sigma implementations. DMAIC stands for

- Define opportunity
- Measure performance
- Analyse opportunity
- Improve performance
- Control performance

DMAIC Methodology The first step in the Six-Sigma methodology is Define (D), where the project is selected. For the selected project, initial goals or targets are set, and a project charter or Statement Of Work (SOW) is developed. It is necessary to establish correctly the current status of the project being undertaken. The actual costs associated with poor quality should be correctly analysed and estimated. Targets for improvement in terms of quality and costs are set. The team members to carry out the tasks are identified. Based on this information, the team then determines more precisely the criteria that are critical to the customer. The deliverables and tools used in this phase are given in Table 23.5.

Table 23.5 *Define phase of Six-Sigma operation*

Deliverables	*Tools*
Project Charter	Spreadsheet/Word Processor
Process and Problem	Flowchart or Process Map
Team, Customers and Critical Concerns	FMEA
Improvement Goals and Objectives	Pareto Chart and Control Charts
Estimate Sigma and Cost Of Poor Quality (COPQ)	QFD / House of Quality
Gantt Chart / Timeline	Process capability
High-Level Process Map	Cause-and-Effects Analysis
Documentation	

The second step in the Six-Sigma methodology is Measure (M). In this phase, the baseline measure of the existing system is taken which is used for gauging the improvement. The critical customer requirements are compared to the existing data. It is important in Six Sigma projects to test repeatability and reproducibility (R&R) of a measurement system wherever critical measures are taken. The project charter and processing map from the previous phase are refined, so that any unwanted steps present in the existing process are detected. The deliverables and tools used in this phase are given in Table 23.6.

Table 23.6 *Measure phase of Six-Sigma operation*

Deliverables	*Tools*
Baseline Figures (Sigma and Cost)	Surveys / Interviews / Focus Groups
Process Capability	Spreadsheets
Measurement System Analysis (MSA) or Gage R&R	Pareto Chart / Control Charts
Refined Project Charter, including COPQ	Measurement System Analysis
Refined Process Map	FMEA
Fix Gantt Chart / Timeline	Process Capability
Step Documentation and Next Steps	Cause and Effects Analysis

The third step in the Six-Sigma methodology is analyse (A). This is a major step in which all the possible root causes for the problem are identified. The team checks with all possible resources in identifying the cause. The identified causes are validated using new or existing data and applicable statistical tools, such as scatter plots, hypothesis testing, ANOVA, regression, or Design Of Experiments (DOE). The deliverables and tools used in this phase are given in Table 23.7.

Table 23.7 *Analyse phase of Six-Sigma operation*

Deliverables	*Tools*
Identified Root Cause(s)	Fishbone Diagram (5-Why)
Validated Root Cause(s)	Cause and effect analysis
Step Documentation and Next Steps	FMEA
	Regression, ANOVA
	Design of experiments
	Response Surface Methods
	Hypothesis testing
	Multi-variable testing

The fourth step in the Six-Sigma methodology is improve (I). First, the team brainstorms and comes up with a number of ideas for improvement. These ideas are then evaluated based on probability of success, time to execute, impact on resources, and cost. The deliverables and tools used in this phase are given in Table 23.8.

Table 23.8 *Improve phase of Six-Sigma operation*

Deliverables	*Tools*
Selected Root Cause(s) and Countermeasures	Affinity Diagram
Validated Solutions or Improvements	Hypothesis testing
Improvement Implementation Plan	Multi-variable testing
Revised Flowchart or Process Map	Design of experiments
Step Documentation and Next Steps	Trial and Error / Simulation
	Mistake proofing

The last step in the Six-Sigma methodology is control (C). The identified plan is executed by planning to have early warning signals when the process gets out of control. Some mistake-proof devices that utilise light, sound, logic programming, or no-go design to help control a process are developed and implemented. The system is run for three to six months, making sure that in this period, the system behaves the way it was planned. At the end of the successful running, it is handed over to the owner to continue to operate. If at this point, Six-Sigma is not achieved, a separate project can be kicked off in the future to address any residual root cause. The deliverables and tools used in this phase are given in Table 23.9.

Table 23.9 *Control phase of Six-Sigma operation*

Deliverables	*Tools*
Control Plan	Mistake proofing
Response Plan	FMEA (Sustain)
Validated In-Control Process and Benefits	Standardisation
Step Documentation and Final Report	Capability Studies
Handover to Owner	MSA or Gage R&R

23.9 INTEGRATION OF CAQC WITH CAD AND CAM

Conventional manual inspection methods requires skill in setting the parts and instruments that is time-consuming and monotonous. This may call for moving the components to the inspection section, which calls for additional time delays and disruption of the regular production cycle. Hence, it is desirable that automated inspection is integrated with manufacturing wherein 100% inspection can be carried out online and the measurements are fed back to enable adjustments in the manufacturing process to ensure that the product is always within the designed specifications. In this chapter, a number of technologies that lead to the quality and the equipment that could be automated are discussed.

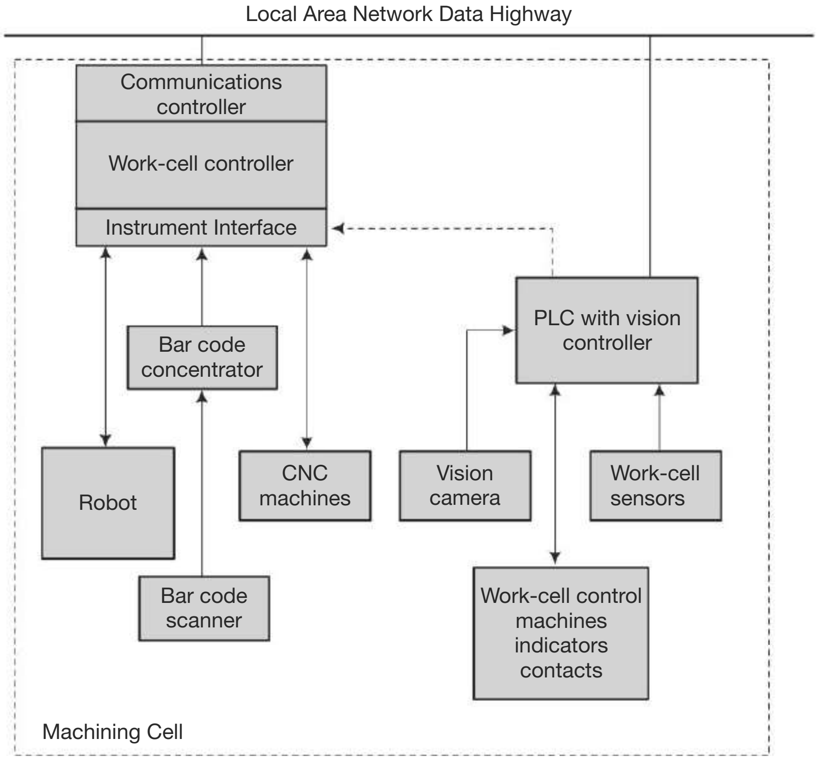


Fig. 23.12 *Communication system in a machining cell*

Getting real-time data in a manufacturing enterprise on all aspects of measurements being done helps in monitoring quality control and assuring quality. This data, when it comes, has to be integrated with the appropriate software modules to achieve the stated functions. An example of the type of integration of measurement activity in a machining cell is shown in Fig. 23.12.

The measuring equipment used in this cell is the vision camera and the associated equipment for grabbing the dimensions. However, other possibilities can be a CMM, other sensors (special measuring set-ups utilising electronic probes) and metrology instruments which can be provided with direct communication facilities such as a serial port (either RS232C or USB) so that they can be directly linked to the work-cell controller which can consolidate the data and then transmit through the LAN to the other databases in the system for further processing. Most of the current-day metrology instruments such as micrometers, height gauges, etc., which were analog, have their digital equivalents available that have built-in communication ports through which they are able to directly communicate the dimension being measured.

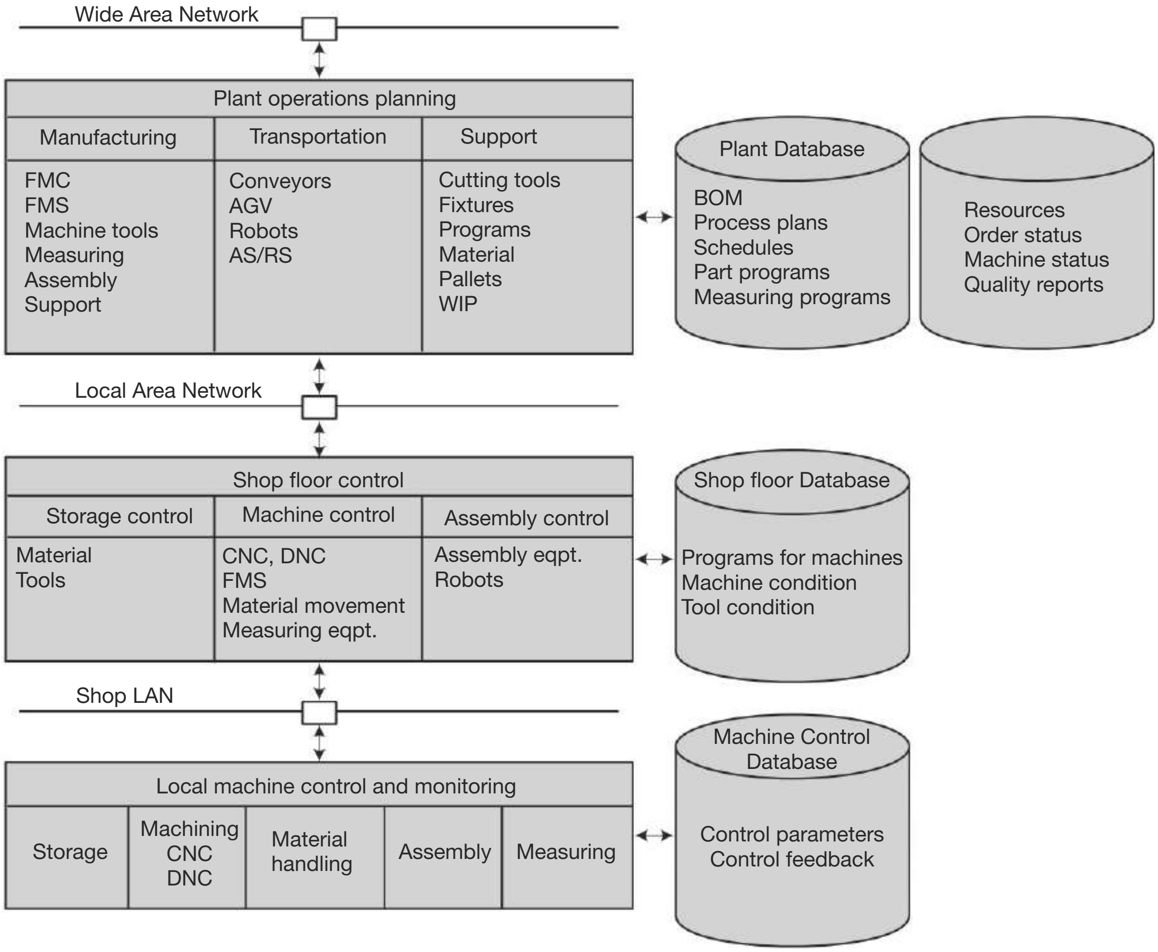


Fig. 23.13 *Schematic for a plant-wide linking of the manufacturing facilities and their databases*

All the manufacturing units in the enterprise can be linked through appropriate networks so that the information is accessed in real time, and also the reports are generated in real time. The software can be provided in such a way that whenever a particular process gets out of control, it can initiate a quick response to alert the manager responsible for the process so that no defective parts are produced. This is possible by linking all the systems as shown in Fig. 23.13 for the entire plant in a single location.

The measuring programs required for operating the CMMs in the shop can be generated manually or automatically through appropriate software systems. One example is the DMIS discussed in Chapter 5. DMIS not only helps in generating the part programs to operate the CMM, but also does a number of quality evaluation functions such as tolerances, acceptance, etc., as well. This needs to be supplemented by appropriate software functions to generate other SQC functions such as control-chart generation.

Summary

- This chapter provided a brief look into various aspects that deal with quality in a computerised environment.
- CAQC involves a broad range of services that aims at reducing the overall cost of manufacturing while improving the quality of the product.
- Inspection is used to examine a given product's conformance to specification, while testing is used to examine the product's ability to perform under normal working conditions.
- The coordinate measuring machine is versatile equipment used for a variety of measurement functions in the shop. There are a number of types of CMMs available for different types of jobs. Different types of probes are used to take the measurements. These can be either manually operated or CNC (operated with a part program under computer control). The software available as part of the control system has many functions to help in identifying the form tolerances, as well as other functions for SQC.
- Non-contact inspection is used for fragile and complex workpieces. Machine vision is the most common method of non-contact inspection.
- Statistical quality control, utilising statistical tools, is used to make sure that the process used for production is under control. R-chart and $\overline{X}$-chart are commonly used to check the process under control.
- Statistical process control is an extension from SQC and concentrates on the process to eliminate defects.
- Total quality management is a philosophy to ensure customer satisfaction by providing highest quality products at low cost.
- Six Sigma is a philosophy to ensure that the defect rate in the organisation is brought down to 3.4 per million parts. Systematic procedure with a number of statistical and other tools is used for the purpose.
- Integrating CAQC with other systems in the organisation helps in obtaining real-time information about quality and helps in keeping the processes under control.

Questions

1. Explain what you understand by computer-aided quality control. Give a block diagram linking the various parts of the system.
2. Explain 'testing' as used in manufacturing industries.
3. Briefly explain Coordinate Measuring Machine.
4. Explain different types of CMMs you are familiar with.
5. What is a probe in CMM? Explain how it works.
6. Mention the advantages of CMM.
7. What are the important features available in CMM software?
8. Explain briefly about non-contact CMM.
9. Write the applications of non-contact inspection.
10. Write briefly about the vision system.
11. Define machine vision.
12. Explain the steps in a vision-processing system.
13. Describe in details of the function and application of a machine-vision system.
14. Explain briefly the function of SQC.
15. Explain the method of construction of R-chart or range chart. Give its application.
16. Explain the method of construction of $\overline{X}$-chart. Give its application.
17. Briefly explain about SPC. How is it different from SQC?
18. Briefly explain the Six-Sigma process. How is it useful for manufacturing industries?
19. Briefly explain TQM.
20. Explain the methods in which CAQC is interlinked with other functions in a manufacturing organisation.

24

COMPUTER INTEGRATED MANUFACTURING

Objectives

A number of instances where automation can be applied have been discussed in the previous chapters. Though each of these bring in advantages to manufacturing, they are only 'islands of automation' if implemented separately and the advantages gained are incremental in nature. Integrating all the islands of automation into a single system enhances the overall benefits. Such a system is called Computer Integrated Manufacturing (CIM). In this chapter, we will discuss about the methodologies of achieving CIM. After completing the study of this chapter, the reader should be able to

- Learn the historical development of CIM methodologies
- Appreciate the basic definition of CIM and the benefits that can be achieved by integration
- Understand the CIM implementation strategies
- Learn the meaning of 'Lean Manufacturing' and the methods to be adopted for implementing lean manufacturing

24.1 HISTORICAL BACKGROUND

The clamour for CIM, or factory of the future, is not new. The dreams of many a technologist of yesteryears was to develop a factory which is completely automatic and starts producing with a simple push of a button. In the case of discrete part manufacture, this involves a large amount of automation as well as coordination involving a large number of systems.

Dr J Hatvany* [Doumeingts] characterised the historical development of CIM as follows:

• Dream period	1950–1970
• Nightmare period	1970 –1980
• Realism period	1980 –…

---

* Lectures delivered by Prof. G. Doumeingts of University of Bordeaux-France to EEC-India participants, 1993.

During the early years (1950 to 1970), a large number of scenarios of CIM have been proposed. However, many of these could not be realised because of the level of technologies during that time. This is particularly true of the computer technology, which relied mostly on electronic valves, transistors and small-scale integration in the integrated circuits of that period.

The next phase is what is called the 'nightmare period', since the technologies were developing at a faster rate, but were not sufficiently mature enough to provide the full capability of CIM at reasonable cost. Further, the reliability of the systems was not sufficiently high to warrant the use of full automation round the clock. An example of the installation during this period is the Mollin's System 24 installation, which has proven the technological feasibility but is not financially successful.

After the earlier dreams about the pushbutton factories of the 60's and 70's, the actual realisation of the CIM potential in industries started to be translated into realities after 1980's. This was possible because of developments in

- Machines
- Computers
- Control systems
- Man–machine interfaces
- Decentralised approach

All the technologies required for CIM have matured sufficiently. With the maturing of the technologies, the cost of implementation is falling so rapidly that it is now possible for everyone concerned with manufacturing to look at the possibility of using it.

Further impetus for the use of CIM is provided by the change in the market scenario. In the past, narrow markets allowed for a very small number of products and technologies to be used. The life of the product was very long with an assured market. The users of the products did not have much choice. This allowed the industries to spend a large amount of time in the development of the product as well as the process. The high cost of such large development effort could be amortised by the assured markets for long times. This kind of situation called for the **economy of scale** to absorb the large developmental costs.

In the new economic situation, with the purchasing power of a large number of individuals going up, the clamour for a variety of products has increased. Further, the consumer has become more diligent and knowledgeable so that the quality and features demanded have become more. The markets have become wider with the collapse of the trade boundaries between the countries. Thus, the products have to satisfy a larger number of different customers. This can be seen based on the number of automobile models that were developed in the last century throughout the world in Table 24.1. This table represents the total production of cars by various manufacturers and the number of different models offered by the various manufacturers.

Thus

- the market has widened,
- with lot of products, and
- a variety of technologies and machines.

Obviously, with a large variety of products, the life as well as the volume of a certain product is limited. This means modularising and standardising in order to reduce the product-development time and cost. In this scenario, the **economy of scope** (variety) is more relevant than economy of scale. For this purpose, one requires flexibility of products and flow manufacturing. They require multi-functional machines and control in real time.

Table 24.1 *Change in number of different models of cars manufactured over the years*

Manufacturer	*Years*	*Total output*	*No. of models*	*No. of platforms*
Ford, USA	1950–54	1 488 679	4.0	3.6
	1960–64	2 041 147	7.2	5.8
	1970–74	2 683 094	11.6	8.0
	1980–84	1 704 633	16.2	6.2
	1990–93	1 696 376	12.5	7.5
General Motors, USA	1950–54	2 690 327	9.4	6
	1960–64	3 755 623	20.8	10
	1970–74	4 670 189	31.0	13.6
	1980–84	4 374 365	32.0	11.0
	1990–93	2 968 965	31.8	15.8
Fiat, Europe	1950–54	132 311	3.6	3.0
	1960–64	721 252	8.4	4.8
	1970–74	1 500 843	14.4	8.8
	1980–84	1 123 002	10.6	7.2
	1990–93	1 513 356	13.5	7.0
Volkswagen, Europe	1950–54	128 756	1.0	1.0
	1960–64	892 856	2.6	1.4
	1970–74	1 450 289	11.0	7.4
	1980–84	1 343 859	8.2	4.4
	1990–93	1 783 173	9.0	5.0
Toyota, Japan	1950–54	—	—	—
	1960–64	116 353	2.0	2.0
	1970–74	1 565 609	8.2	5.4
	1980–84	2 369 111	16.2	8.8
	1990–93	2 827 028	24.3	13.8
Nissan, Japan	1950–54	—	—	—
	1960–64	100 955	2.0	2.4
	1970–74	1 217 150	9.4	9.8
	1980–84	1 792 074	16.8	13.8
	1990–93	1 598 416	22.3	17.5

In the new economic situation, it is required to decrease innovation time because of the rapid evolution of products. In order to reduce the cost, it is required to decrease the lead time and inventory. In actual practice, it is observed that out of the total time taken by the part, only 5% is for the machining time and the rest is for handling and waiting time. In this scenario, the Japanese concepts of Mechatronics and Just-In-Time (JIT) production are important for developing competitive industrial enterprises.

24.2 INTEGRATION

The Society of Manufacturing Engineers (SME) defined CIM as *'CIM is the integration of the total manufacturing enterprise through the use of integrated systems and data communications coupled with new managerial philosophies that improve organisational and personal efficiency'*.

CIM basically involves the integration of all the functions of an enterprise. The new CIM wheel (Fig. 24.1) of the Society of Manufacturing Engineers illustrates this concept well and demonstrates the inter-relationship among the various segments of the enterprise. CIM is generally considered as a new approach to manufacturing, management and corporate operation. It is generally interpreted that CIM includes most of the advanced manufacturing technologies such as computer aided design, computer numerical control, robots, just in time production, etc. However, CIM goes beyond all those technologies and provides a new way of doing business that includes commitment to customer satisfaction, total quality, and continuous improvement. As discussed in the earlier chapters, a single enterprise-based database that supports all the information needs for manufacturing in every department becomes an essential part of CIM. This database removes the communication barriers between various departments of an enterprise allowing for complete integration of all departments.

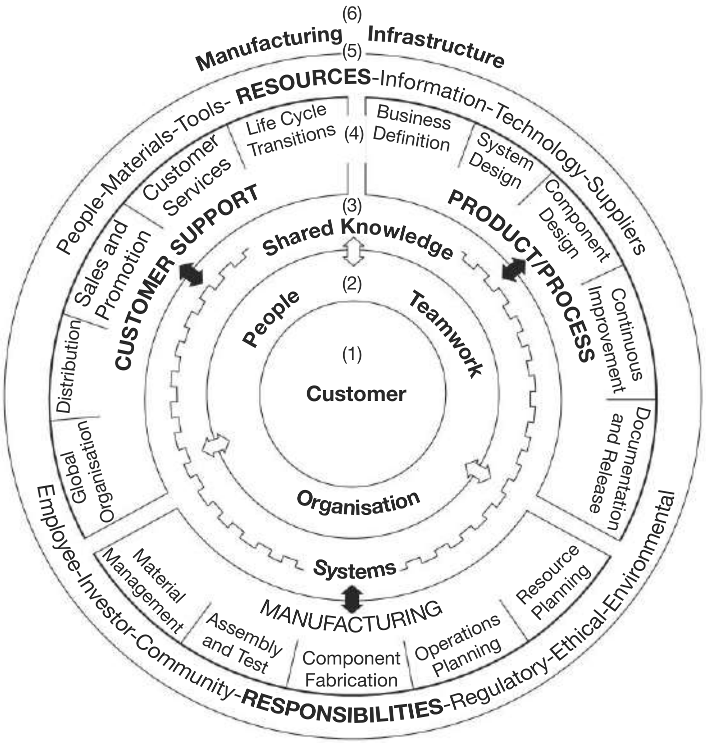


Fig. 24.1 *The new manufacturing enterprise wheel suggested by Society of Manufacturing Engineers (Courtesy, of the Society of Manufacturing Engineers, Dearborn, Michigan)*

The SME Manufacturing enterprise wheel has 6 defined areas.

1. The success of an enterprise depends on the customer, and thus the customer becomes the hub of the wheel. With a clear understanding of the customer requirement and the market place, the enterprise will succeed.
2. The next level focuses on the organisational structure of the enterprise. This deals with the organising people, training, motivation and cooperation in teamwork. There are a number of techniques used to achieve these goals such as organisational learning, leadership, standards, quality circles, and rewards.
3. The third level in the wheel focuses on the shared knowledge of the enterprise. This includes all the databases and archival knowledge and experience, all of which can be utilised to support the people and the processes.
4. All the systems that are actually used in the total enterprise are present in this part of the wheel. All the processes are grouped into three major categories, namely, product and process, manufacturing and customer support. Each of these has the components that actually perform the necessary functions.
5. Resources and responsibilities of the enterprise are included in this section. The resources are the people, materials, tools, information, technology and suppliers. The responsibilities will be to the employees, investors and the communities that it will be serving while undertaking the statutory, ethical and environmental safeguards.
6. The final part of the wheel is the actual manufacturing infrastructure. This includes the entire infrastructure such as customers and their needs, suppliers, distributors, prospective workers, natural resources, financial markets, educational and research institutions and competitors.

There should be a tight integration between all the segments as shown to achieve the benefits of CIM.

In the earlier chapters many of the components of CIM have been dealt in sufficient detail. In order to achieve the CIM, it therefore becomes necessary to integrate all the functions (either automatic or manual) through some means such that the benefits of automation can be achieved. For example, FMS may be considered as ‘mini CIM’ since a number of functions of CIM are already available within an FMS. If the FMS control program is linked to the business-data processing unit, CIM can be realised. However, one difference is that FMS relies on complete automation with very little manual intervention, save for the work and tool-preparation areas. However, in CIM there can be a large number of manual operations present besides the automated equipment, all of which will have to be taken into account while planning and implementing CIM.

To illustrate the benefits of integration, an example of possible information flow for tooling systems in a CIM environment is shown in Fig. 24.2. With the integration of tooling, it is possible to consider that as a resource and take it into consideration during the production planning. Similarly, when the rescheduling is to be done on the shop floor, it is possible to consider the tooling as a constraint to release orders for the specific work centres. The tooling system can collect information and provide a better feedback to the machinability data systems to improve the cutting process parameter selection as well as the tool-life prediction algorithms. In the same manner, it is possible that by collecting the information and archiving, it is possible to see the tools usage pattern, which can help in rationalising the tools, whereby a reduction in tool variety as well as tool inventory can be done over a period of time.

It is also possible to build similar interactions with other sub-systems of the manufacturing process as well, whereby it is possible to obtain full synergic benefits of integration to the total enterprise by the adoption of CIM as a whole. One point which may be mentioned here is that CIM is to be conceived as a total system and will have to integrate all the functions carefully. The CIM solution is not available off-the-shelf and has

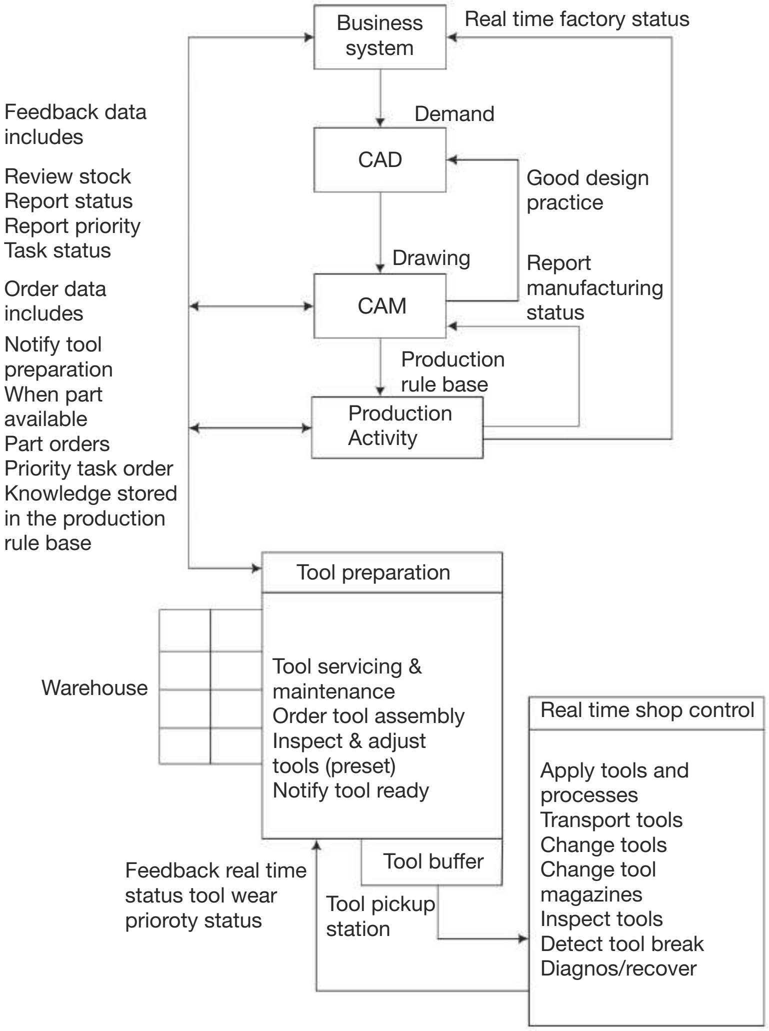


Fig. 24.2 *The interaction of the tooling system with other functions in CIM*

to be tailored for individual cases from ground up. However, many of the modules are individually available which need to be selected for the given application and then customised and integrated. All this requires careful planning and management commitment since the path takes a longer time for implementation as well as achieving the results.

24.3 CIM IMPLEMENTATION

CIM structure development requires a very clear understanding of the total information flow within the system. A hierarchical structure of the CIM developed in ESPIRIT 809 [van Houten, 1992] is shown in Fig. 24.3. This has five levels for an individual factory. The same can be extended for an enterprise by adding another top level that links the corporate to all the individual factories. As explained earlier, it is important to make arrangements to get the necessary information as required at the right time. The architecture shown in Fig. 24.3 includes all the components, viz., factory, system, cell, workstation and equipment levels.

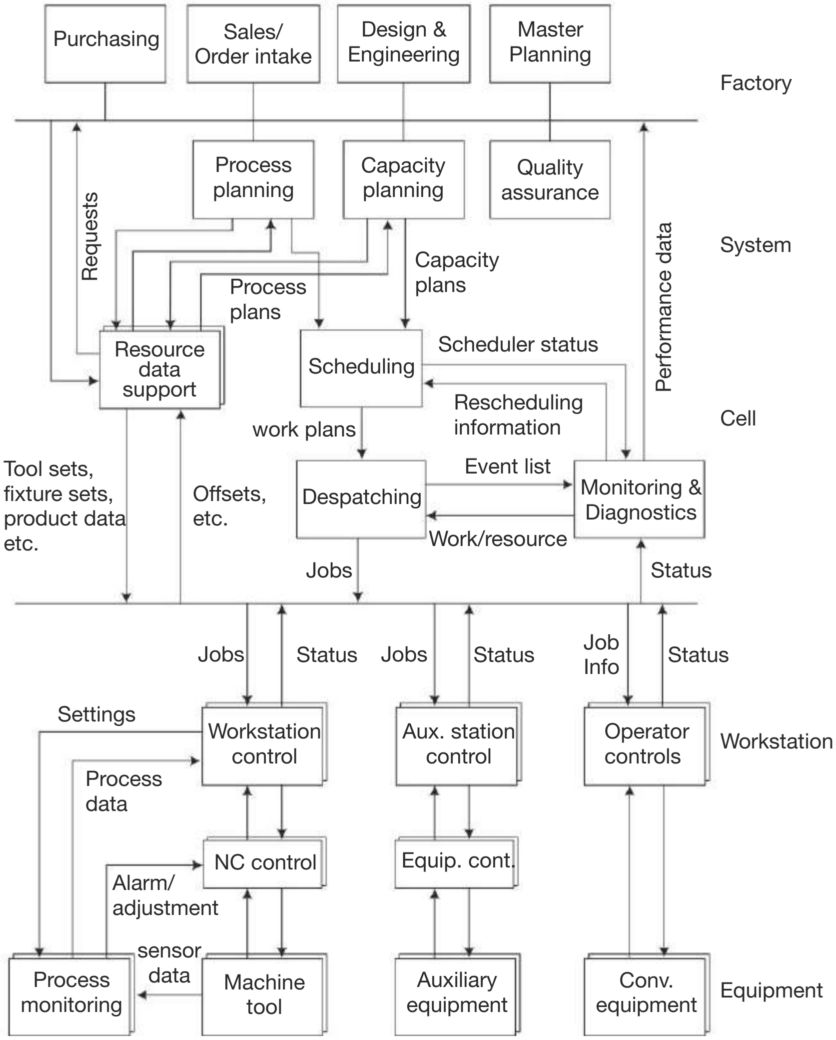


Fig. 24.3 *Functions and hierarchical levels in CIM [redrawn from van Houten, 1992]*

CIM is generally considered by many consisting of all computer controlled and automatic equipment with no manual intervention. However, the reality is much different. The aero-engine manufacturing plant, shown in Fig. 24.4, demonstrates this concept well. The manufacturing requirements are taken care of by means of four manufacturing cells that are fully automatic. These are two FMS units, one turning cell and a broaching cell. In addition to that, the other manufacturing operations, such as deburring, shot peening and inspection operations are all manual. The FMS are supported by two automatic storage and retrieval systems for fixtures and parts. The movement of the parts between the various cells is handled by automated guided vehicles through the AGV route shown in Fig. 24.4.

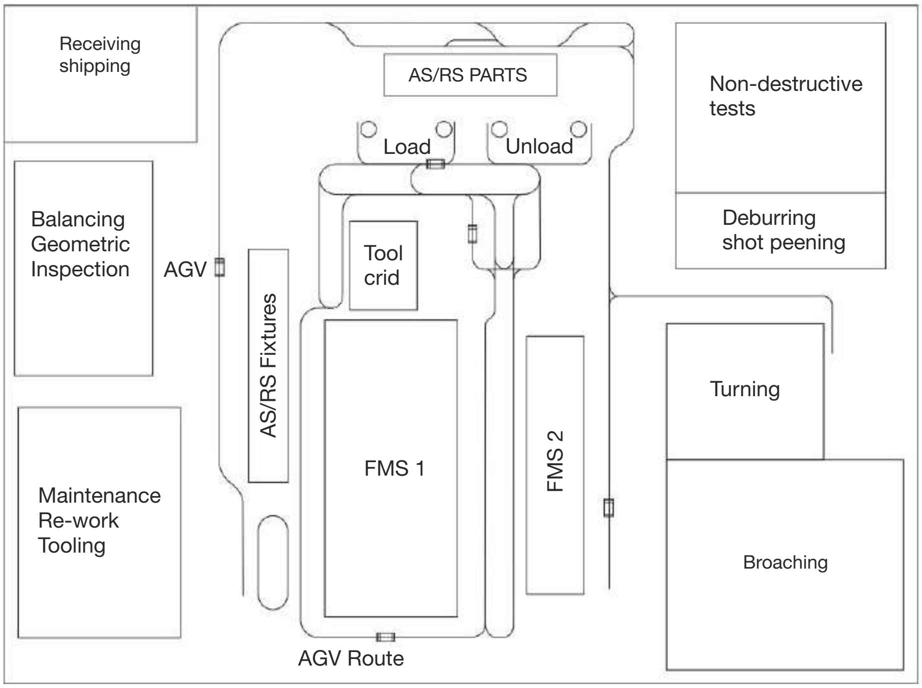


Fig. 24.4 *CIM implementation at Snecma plant manufacturing aero-engine parts*

Major functions of the information and the associated interfaces are shown in Fig. 24.5. Following concurrent engineering practices becomes an important requirement for the implementation of CIM. Also, the use of flexible manufacturing practices greatly helps in running a CIM plant. Design and engineering is the first step in satisfying the customer requirements. Utilise the CAD and CAE techniques as discussed in Chapters 4 to 8.

The analysis techniques such as FEA, flow analysis, and kinematics are used in order to optimise the products. Also, artificial intelligence techniques are widely used in some of the advanced CAD systems that are required for specific applications such as tool design. The detailed design carried at this stage is used by all the downstream applications such as CAPP and CAM. At this stage, the importance of neutral data formats for transferring the data models between various systems have to be carefully taken care of to see that the product data is properly transferred without any loss of design information or the intent. For this purpose, STEP standard has reached reasonable maturity at this point of time, and is quite extensively used.

The hardware used for design and analysis has to be properly organised taking into account the information needs. A typical local area network (LAN) for the purpose is shown in Fig. 24.6. Common databases required for the design and analysis are stored in the local databases. Also, the corporate databases are accessible through appropriate bridges for common data. Also, other users are able to access the data from the design and analysis LAN through bridges as shown.

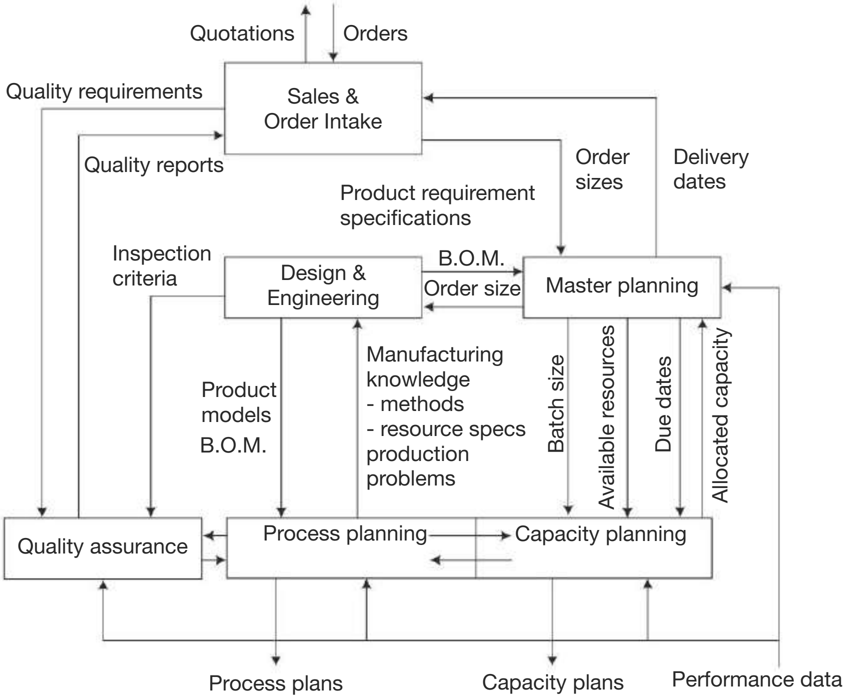


Fig. 24.5 *Major functions in CIM [redrawn from van Houten, 1992]*

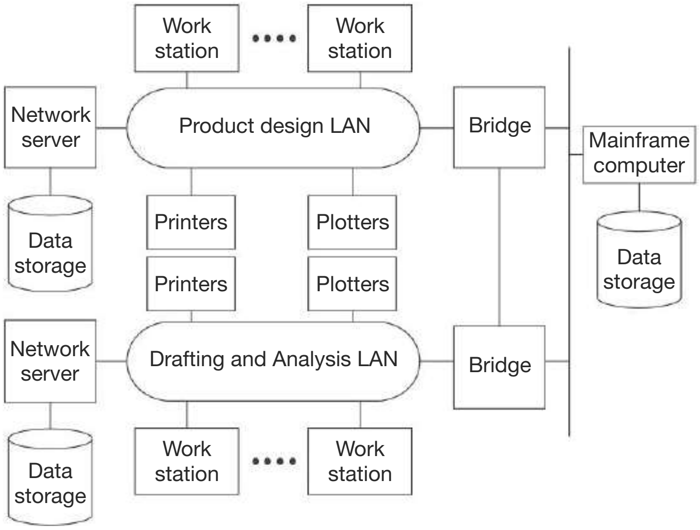


Fig. 24.6 *An example of LAN utilised for the design room*

The interface requirements for planning functions are shown in Fig. 24.7. Customer's requirements are taken into account while developing the products and quality requirements. The planning department takes the

Bill Of Materials (BOM) from the design and prepares the production plans. As discussed in earlier chapters, the master production schedule is broken down into planned orders with specific due dates. Similarly, the quality assurance department develops the inspection requirements from the product-design specifications. Also, it gets feedback from the processes by monitoring and diagnostic function.

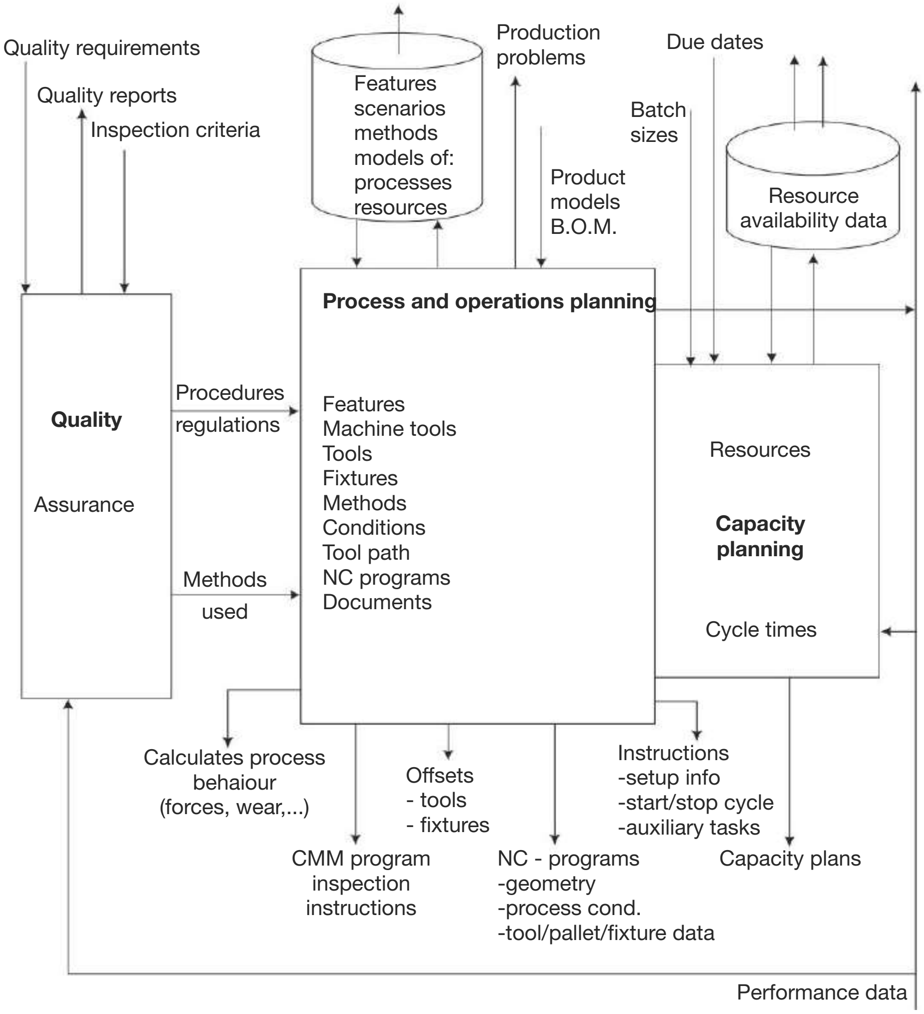


Fig. 24.7 *Some interface requirements in CIM [redrawn from van Houten, 1992]*

Process planning is another function which is being utilised as a strong interface between design and manufacturing. Computer Aided Process Planning (CAPP), as discussed earlier, is used to identify the set-ups required, operations required, information for tooling design, and information for other applications of CAM. There are a number of CAPP systems available, and many are being developed as proprietary systems for specific applications. The CAM applications take the information from CAPP and calculate the tool paths and generate the CNC part programs. Also, it is necessary to develop any of the inspection programs required for the Coordinate Measuring Machine (CMM). The feedback from the performance data is utilised for modification of the process definitions. Since the systems have to work in real time, it is necessary to develop alternate process and inspection plans depending upon the availability of manufacturing resources.

Once the process plans, inspection plans, and the production plans are ready, it is possible to generate the necessary schedules and prepare the capacity plans. All the information required for the operation of the manufacturing is now available. This information is then passed on to the shop-floor control systems, whose interfaces are shown in Fig. 24.8. Capacity plans give the manufacturing quantities for all products for a specified manufacturing period. The jobs also come with their preferential sequences to help in the optimal utilisation of the resources.

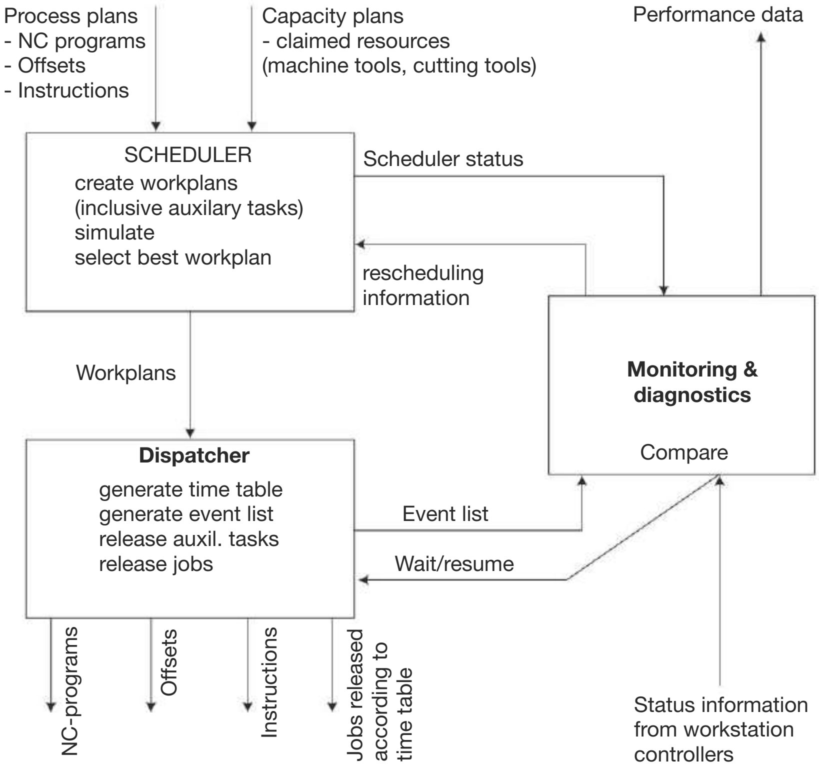


Fig. 24.8 *Shop-floor control and interfaces at the cell level in CIM [redrawn from van Houten, 1992]*

As shown in Fig. 24.8, Shop Floor Control (SFC) systems incorporate scheduling, manufacture, monitoring and diagnostics at the equipment level used in the shop floor. Capacity plans provide all the necessary information required for scheduling the operations. Other necessary information such as CNC programs, tool offsets, and manufacturing instructions are also available at this point. The equipment-level networking is shown in Fig. 24.9. The area controller controls and monitors the execution of various elements of the manufacturing system, which include downloading of part programs and tool offsets, downloading of control commands, start/stop cycle, monitoring the status of jobs and processes, and collecting the performance and maintenance diagnostics.

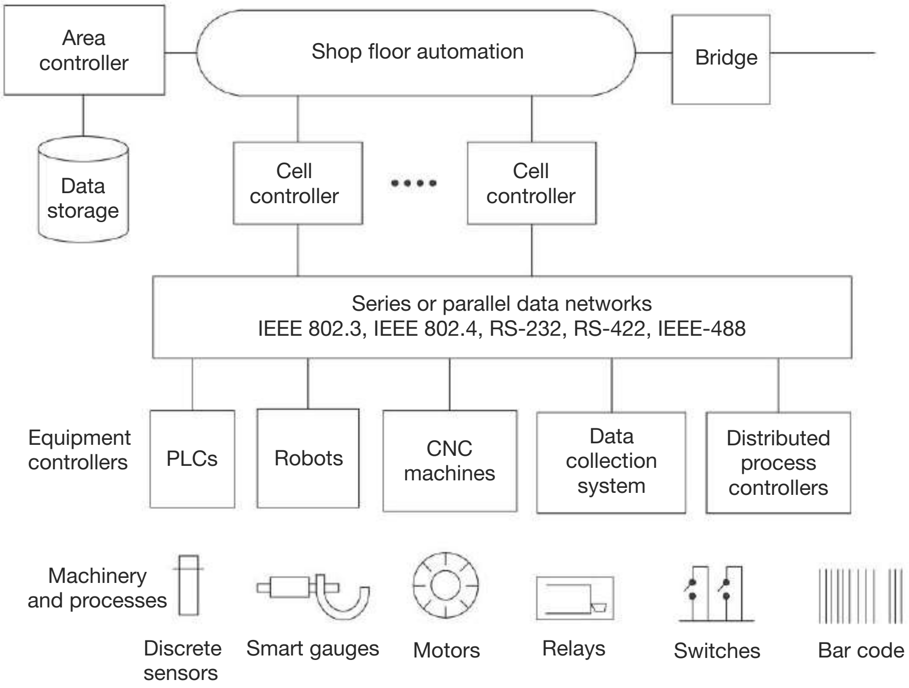


Fig. 24.9 *Shop-floor control interfaces with the equipment*

Most of the automated equipment is directly linked to the system and has the automatic transmission of information to the controller. However, if there are any manual operations then they need to have messaging systems through shop terminals located at strategic locations.

24.4 BENEFITS OF CIM

CIM provides a means to manage and control the uncontrollable disturbances while meeting customer demands and requirements. In the following is listed a few of the benefits that can be achieved by the use of CIM.

1. CIM improves the operational control through
   - reduction in the number of uncontrollable variables
   - providing tools to recognise and react quickly to deviations in the manufacturing plan
   - reducing dependence on human communication
2. CIM improves the short-run responsiveness consisting of
   - engineering changes
   - processing changes
   - machine downtime or unavailability
   - operator unavailability
   - cutting-tool failure
   - late material delivery

3. CIM improves long-run accommodations through quicker and easier assimilation of
    - changing product volumes
    - new product additions and introductions
    - different part mixes
4. CIM reduces inventory by
    - reducing lot sizes
    - improving inventory turnovers
    - providing the planning tools for Just-in-Time manufacturing
5. CIM increases machine utilisation by
    - eliminating or reducing machine set-up
    - utilising automated features to replace manual intervention to the extent possible
    - providing quick transfer devices to keep the machines in the cutting cycle

Some quantifiable benefits achieved by applying CIM are the following:

- Engineering design costs can be reduced by 15 to 30%.
- The overall lead times can be reduced by 30 to 60%.
- The product quality can be increased dramatically, as measured by the yield of acceptable product, offering 2 to 5 times the previous level.
- Productivity of the manufacturing operations can be increased by 40 to 70%.
- Operating time related productivity can be increased by 2 to 3 times.
- The productivity of engineers and engineering managers can be increased by 5 to 35 times, measured in terms of extent and depth of analysis in the same or less time.
- Work in process can be reduced by 30 to 60%.

24.5 LEAN MANUFACTURING

Manufacturing methods, which are collectively called 'Lean Manufacturing' pioneered by Toyota, Japan, have been adopted successfully by many manufacturing companies as a way to reduce costs, better satisfy customers, and increase profitability. Lean manufacturing is a way of thinking that develops a culture of eliminating non-value adding activities while responding to customer needs. Lean manufacturing reaches into every aspect of a company. The process of becoming lean may mean transforming the current company's existing style of operations to an entirely different one. Lean manufacturing may involve process reengineering, adopting new technologies or adding new and different equipment. This may generally involve significant changes in human resources such as education, training, practices and policies.

The three steps involved in lean manufacturing [Womack and Jones, 2003] are the following:

- Define *value* in terms of specific products or services with specific capabilities offered at specific processes through a dialogue with specific customers. This is based on the fact that providing wrong goods or services creates waste or *muda* as it is called in Japanese.
- Identify the *value stream*, which is the set of all the specific actions required to bring a specific product or service or a combination of the two through the three critical management tasks of any business. The management tasks are *problem-solving task* to develop the product design from conception, *information-management task* providing all the necessary information from order taking to the delivery, and the *physical-transformation task* converting the raw material to finished product in the hands of the customer.

- Make the value, creating steps following the *flow* methodology. Conventional thinking and manufacturing practices are based to a great extent on batch and department concept. However, batches mean a long wait and queues at each of the value stream. These are the places, where in the traditional systems, one finds enormous amounts of waste in the form of buffers and WIP. Ford adopted the flow manufacturing principles in 1913 in assembling the Model T and reduced the total effort by 90 per cent. He applied this for all the upstream manufacturing operations and made the Ford car affordable for everybody. However, this is a special case of flow manufacturing involving really large-volume production (Ford was producing model Ts in millions). It is necessary to apply methods that will bring the same amount of productivity for small and medium batch quantities. So in lean thinking, the work functions, departments and firms should be defined that actually contribute to the value. Another important element in flow manufacturing is the use of the *pull* system. The customer will pull the product from the stream that will generate the necessary upstream processes to fill the void created by the pull operation, rather than making in large volumes and pushing into the market.

Lean thinking normally tries to break the problem into small parts so that optimal solutions can be found for these. Generally, massive solutions involving fully automatic equipment does not always solve the problem. An example may be given here from Pratt and Whitney, Aircraft engine manufacturer [Womack and Jones, 2003].

This is an operation involving the grinding of the base of turbine blades so that they snugly fit into the disk holding it to the engine. The original process involved a lot of manual operations supervising the machines, gauging the parts, and moving the parts from machines to storage areas and to the machines. First, a total automation of the shop is planned by installing custom-built Hauni-Blohm blade grinding centres at a cost of $80 million. These are fully automatic machines with 12 axes in operation which perform all the grinding operations on the blades and complete them in 3 minutes. There are a total of 12 grinding centres, which are fed by robotic devices and the parts are carried to and from storage by Automated Guided Vehicles (AGV). Thus, it is a completely automated cell without any manual operations.

However, using such an automated grinding system called for some additional problems, which need to be undertaken by new solutions. The grinding centres applied large grinding forces on the blades, because of which the conventional fixturing system using standard locators concentrated forces at a few points, destroying the blade. Hence, a new method of fixturing by encapsulating with a low melting temperature alloy had to be used for the blades. This called for an encapsulation process involving liquid metal, expensive moulds, and long changeover times with a batch process. So a number of blades are encapsulated at a given time and then they are stored till they are used by the grinding centres. This called for the use of AGV and AS/RS.

After the grinding operation, the low-temperature alloy has to be removed from the blade. It is necessary to ensure that there are no traces of this alloy on the blade, otherwise they cause hot spots and rapid failure of the blade in the engine. So sophisticated tests involving X-rays and an atomic absorption process using caustic chemicals to test for the trace elements of the alloy was used for this purpose. The finished grinding system cell is as shown in Fig. 24.10.

The changeover times are typically eight hours from one family of blades to another. This did not allow for the use of small batches of blades, but large batches, which was the original assumption. However, Pratt required making small numbers of a wide variety of blades. Also, the cell had to be provided with skilled technicians who debugged the elaborate computer system controlling the entire process.

The complex automated cell is replaced by a lean cell (Fig. 24.11) with three-axis grinding machines utilising ingenious quick-change fixtures to hold the blades without the need for encapsulation. Each cell has

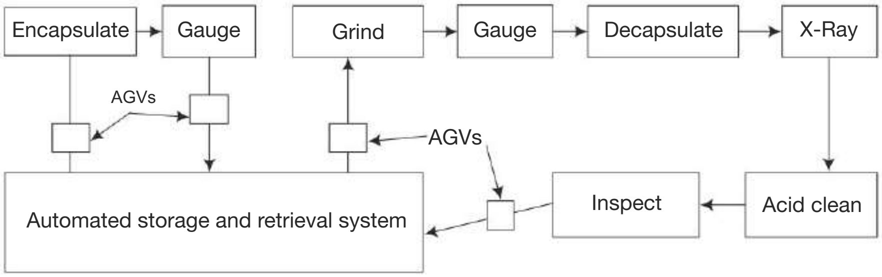


Fig. 24.10 *Automated blade grinding system with fully automated grinding centre [Womack and Jones, 2003]*

one multifunctional worker. The worker moves the parts from one machine to the other by hand, gauge parts and change over the machine for the next part in less than two minutes. This allows for small batches and production to be carried out when needed.

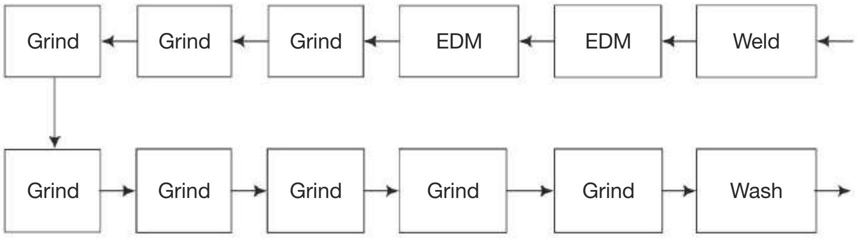


Fig. 24.11 *Lean blade grinding system [Womack and Jones, 2003]*

The processing time for the blade has increased from 3 minutes to 75 minutes in the lean system, which decreased the throughput time from ten days to 75 minutes. Also, the downtime for changeovers is reduced by more than 99 per cent. The space required could be reduced by 60 per cent. Total manufacturing cost could be reduced by more than half for a capital investment of less than $1.7 million for each of the new cell. The overall comparison of the two systems is shown in Table 24.1.

Table 24.2 *Comparison of lean manufacturing with old practice*

	Automated Blohm Grinder	*Lean system*
Space utilised (sq. m)	600	230
Part travel (m)	762	24
Inventory (average for cell)	1640	15
Batch size (number of blades)	250	1
Throughput time	10 days	75 min
Changeover downtime	480 min	100 s
Grinding cost per blade	1.0	0.49
New blade type tooling cost	1.0	0.30

As can be noticed from the above comparison, lean thinking does not have to call for fully automatic systems with no manual operations. They make use of simple processes with multifunctional workers. The application of lean-manufacturing concepts to the shop floor will have typical goals to reduce waste by using processes that [Jordan and Michel, 2001]

- reduce process time
- reduce human effort
- reduce/eliminate defects
- minimise tool/equipment investment
- reduce/eliminate inventories
- reduce space requirements
- reduce financial commitments
- eliminate non-contributing operations

Summary

- Computer Integrated Manufacturing (CIM) is a method to integrate all phases of product-cycle operations. It is not necessary that all the operations need to be automatic or computer controlled.
- CIM has been thought out for a long time, but the technologies for the complete implementation of CIM matured in the recent years. As a result, CIM implementation is now feasible for practically most of the industries.
- CIM basically involves the integration of all the functions of an enterprise as defined by SME.
- CIM implementation can be done by dividing the enterprise operation into hierarchical levels and following proper standards at each level, viz., factory, system, cell, workstation and equipment levels.
- 'Lean Manufacturing' pioneered by Toyota, Japan, is a way to reduce costs, better satisfy customers, and increase profitability.

Questions

1. Define Computer Integrated Manufacturing.
2. Describe the factors responsible for the success of CIM applications in the present time.
3. Explain the conditions that have led to the necessity of the application of CIM in modern manufacturing industries.
4. Explain what you understand by 'economy of scale' and 'economy of scope'. Which of them is more relevant now? Explain the reasons.
5. Explain the SME manufacturing enterprise wheel.
6. Explain with an example how the integration benefits any manufacturing operation.
7. Explain the aspects that one should consider in implementing CIM.
8. Briefly explain the advantages that will be gained by the implementation in CIM.
9. What do you understand by the term 'lean manufacturing'?
10. Explain the steps used in implementing lean manufacturing.
11. What are steps to be considered on a shop floor for implementing lean manufacturing principles?

BIBLIOGRAPHY

1. Amirouche, F.M.L., *Computer Aided Design and Manufacturing,* Prentice Hall, New Jersey, 1993, ISBN 0-13-175704-0.
2. Asfahl, C. Ray, *Robots and Manufacturing Automation,* 2nd Edition, John Wiley, 1992.
3. Barsky, B.A., *Computer Graphics and Geometric Modelling using Beta-splines,* Springer Verlag, Berlin, 1988.
4. Bauer, J., Browne, R., Bowden, J., Duggan and G., Lyons, *Shop Floor Control Systems: From Design to Implementation,* Chapman & Hall, London, 1991, ISBN: 0-412-36040-3.
5. Bedworth, D. D., M. R., Henderson and P. M., Wolfe, *Computer-Integrated Design and Manufacturing,* McGraw-Hill, New York, 1991.
6. Black, J.T.,: *The Design of a Factory with a Future,* McGraw-Hill, New York, 1991.
7. Bralla, J.G., *Design for Excellence,* McGraw-Hill, New York, 1996, ISBN 0-07-007138-1, 326 pages.
8. Brauer, J.R., *What Every Engineer should know about Finite Element Analysis,* Marcel Dekker, New York, 1988, ISBN 0-8247-7832-4.
9. Browne, Harhen and Shivnan, *Production Management Systems: A CIM Perspective,* Addison Wesley, Reading, 1988.
10. Burbidge, J.L., "*Production Flow Analysis for Planning Group Technology."* Journal of Operations Management, Vol. 10, no. 1 (1991), pp. 5-27.
11. Burbidge, J.L., *Production Flow Analysis for Planning GT*. Oxford: Oxford University Press/Oxford Science Publications, 1989.
12. Burbidge, John L., *Comment on clustering methods for finding GT groups and families,* Journal of Manufacturing Systems, 1993, Vol. 12, no 5, pp 428 – 429.
13. Chandrupatla, T.R., and A.D., Belegundu, *Introduction to Finite Elements in Engineering,* Prentice Hall, New Jersey, 1991, ISBN 0-87692-682-0.
14. Chang, T. C., *Expert Process Planning for Manufacturing,* Addison Wesley, Reading, 1990, ISBN: 0-201-18297-1.
15. Chang, T.C., R.A., Wysk and H.P., Wang, *Computer Aided Manufacturing,* 2nd Edition, Prentice-Hall, New Jersey, 1998.
16. Chao Hwa Chang and Michael A. Mekanoff, *NC Machine Programming and Software Design*, Prentice-Hall, New Jersey, 1989.

17. Chasen, S. H., *Geometric Principles and Procedures for Computer Graphic Applications*, Prentice Hall, New Jersey, 1978.
18. David Saloman, *Computer Graphics and Geometric Modelling,* Springer, New York, 1999.
19. Deb, S.R., *Robotics Technology and Flexible Automation,* Tata McGraw-Hill, New Delhi, 1994, ISBN 0-07-460090-7.
20. Enderle, G., K., Kansy and G., Pfaff, *Computer Graphics Programming: GKS - The Graphics Standard*, Springer-Verlag, Berlin, 2nd Edition, 1987.
21. Faux, I.D., and M.J., Pratt, *Computational Geometry for Design and Manufacture*, Ellis Harwood, Wiley, London, 1979.
22. Frank Potts, J., and J., Walter Oler, *Finite Element Applications with Microcomputers*, Prentice Hall, Englewood Cliffs, 1989, ISBN 0-13-317439-5.
23. Gibbs, D., and T.M., Crandell, *An Introduction to CNC Machining and Programming*, Industrial Press, New York, 1991, ISBN 0-8311-3009-1.
24. Groover, M. P., and E. W., Zimmers, Jr., *CAD/CAM: Computer Aided Design and Computer Aided Manufacturing*, Prentice-Hall, New Jersey, 1984.
25. Groover, M. P., *Automation, Production Systems and Computer Integrated Manufacturing*, 2nd Edition, Prentice-Hall, New Jersey, 2001.
26. Groover, M.P., Mitchell Weiss, Roger N. Nagel, and Nicholas G. Odrey, *Industrial Robotics,* McGraw Hill, New York, 1986.
27. Hall, R., *Attaining Manufacturing Excellence,* Dow-Jones-Irwin, Homewood, Illinors, 1987.
28. Hammond, Gary, *AGVs at work*, IFS (Publications), UK, 1986.
29. Hannam, R., *Computer Integrated Manufacturing from Concepts to Realisation,* Addison Wesley, England, 1997.
30. Hans B. Kief and T.F., Waters, *Computer Numerical Control A CNC Reference Guide,* Glencoe, Illinois, 1992, ISBN 0-07-112673-2.
31. Harrington, J., *Computer Integrated Manufacturing*, R.E. Kreiger, Huntington, N.Y., 1973.
32. Held, M., *On the Computational Geometry of Pocket Machining,* Springer Verlag, Berlin, 1991.
33. HMT Limited, *Mechatronics and Machine Tools,* McGraw-Hill, New York, 1999, ISBN 0-07-134634-1.
34. Hodges, B., *Industrial Robotics,* Butterworth Heinemann, Oxford, 1992, ISBN 0-7506-0781-5.
35. Hsu, T., and D.K., Sinha, *Computer Aided Design - An Integrated Approach*, St. Paul West Publishers, 1992.
36. Jacobs, Paul F., *Rapid Prototyping & Manufacturing, Fundamentals of StereoLithography,* SME, McGraw Hill, New York, 1992, 434 pp., ISBN: 0-07-032433-6.
37. Jai Kumar, R., *Post Industrial Manufacturing*, *Harvard Business Review*, 1986, Nov. – Dec., pp. 69 – 76.
38. James Madison: *CNC Machining Handbook,* Industrial press, New York, 1996, ISBN 0-9311-3064-4.
39. Jha, N. K. , *Handbook of Flexible Manufacturing Systems,* Academic Press, San Diego, 1991.
40. John Davis, B., *CIM Software and Interfaces*, *Computers in Industry,* Vol. 33, 1997, pp. 91 – 99.
41. Johnson, R.H., *Solid Modelling*, North Holland, Amsterdam, 1987.
42. Jordan, Jr., J.A., and F.J., Michel, *Valuing Lean Manufacturing Initiatives*, *SME Paper no. MS01-104,* Society of Manufacturing Engineers, 2001.
43. Larry Horath: *Computer Numerical Control Programming of Machines,* Macmillan, New York, 1993, ISBN 0-02-357201-9.
44. Lee J. Krajewski and Larry P. Ritzman, *Operations Management Strategy and Analysis*, Addison-Wesley, reading, 1996, ISBN 0-201-40016-2.
45. Luggen, W. W., *Flexible Manufacturing Cells and Systems,* Prentice-Hall, New Jersey, 1991, ISBN: 0-13-321977-1.

46. Luggen, William, *CNC: A First Look Primer,* Delmar Publishers, New York, 1997, ISBN: 0-8273-7245-0.
47. Mahadevan, B., *The New Manufacturing Architecture*, Tata McGraw-Hill, New Delhi, 1999, ISBN 0-07-463208-6.
48. Maleki, R. A., *Flexible Manufacturing Systems The Technology and Management*, Prentice Hall, Englewood Cliffs, 1991.
49. Marciniak, K., *Geometric Modelling for Numerically Controlled Machining,* Oxford University Press, 1992, ISBN: 0-19-856353-1.
50. McMahon, C., and J., Browne, *CAD CAM From Principles to Practice, 2nd Edition,* Addison Wesley, England, 1998.
51. Michael Pinedo and Xiuli Chao, *Operations Scheduling with Applications in Manufacturing and Services*, Irwin McGraw-Hill, Boston, 1999, ISBN 0-07-289779-1.
52. Mike Lynch: *Computer Numerical Control Accessory Devices*, McGraw Hill, New York, 1994, ISBN 0-07-039226-9.
53. Miller, R.K., *Automated Guided Vehicles and Automated Manufacturing*, Society of Manufacturing Engineers, 1987.
54. Morris A. Cohen and Uday M. Apte, *Manufacturing Automation*, McGraw Hill, New York, 1997, ISBN 0-256-14606-3.
55. Mortenson, M.E., *Geometric Modelling*, John Wiley & Sons, New York, 1985.
56. Newman, W.M., and R.F., Sproull, *Principles of Interactive Computer Graphics*, McGraw-Hill, New York, 1979.
57. Nyman, L.R., (Ed.), *Making Manufacturing Cells Work*, McGraw Hill, New York, 1993.
58. Opitz, H., *A Classification System to Describe Workpieces,* Pergamon Press, Oxford, 1970.
59. Parrish, D., *Flexible Manufacturing,* Butterworth-Heinemann, Oxford, 1990, ISBN: 0-7506-1011-5.
60. Pimentel, J.R., *Communication Networks in Manufacturing,* Prentice-Hall, New Jersey, 1990, ISBN: 0-13-168376-7
61. Ranky, P.G., *Flexible Manufacturing Cells and Systems in CIM,* CIM ware Ltd, 1990, ISBN: 1-872631-02-9.
62. Rao, P. N., N. K., Tewari and T. K., Kundra, *Computer Aided Manufacturing,* Tata McGraw-Hill, New Delhi, 1993.
63. Rehg, J.A., and H.W., Kraebber, *Computer Integrated Manufacturing, 2nd Ediiton,* Prentice Hall, New Jersey, 2005.
64. Rehg, J.A., *Introduction to Robotics in CIM Systems, 5th Edition,* Prentice Hall, New Jersey, 2003.
65. Rembold, U., B.O., Nanji and A., Storr, *Computer Integrated Manufacturing and Engineering,* Addison Wesley, England, 1993, ISBN 0-201-56541-2.
66. Requicha, A.A.G., and H.B., Voelker, "*An Introduction to Geometric Modelling and its Application in Mechanical Design*", *Advances in Information Systems*, Vol. 8, 1981, Plenum Press, New York.
67. Rogers, D.F., *Procedural Elements for Computer Graphics*, McGraw-Hill Book Co., New York, 1985.
68. Scheer, A.W., *CIM-Computer Integrated Manufacturing Towards the Factory of the Future,* Springer Verlag, Berlin, 1991.
69. Schlechtendahl, E.G., *CAD Data Transfer for Solid Models,* Springer Verlag, Berlin, 1989.
70. Scholz Reiter, *CIM -Interfaces: Concepts, Standards, & Problems of Interfaces in CIM*, Chapman & Hall, London, 1992.
71. Seames, W.S., *Computer Numerical Control: Concepts and Programming,* 2nd Edition, Delmar, New York, 1990.
72. Singh, N., *Systems Approach to Computer-Integrated Design and Manufacturing*, John Wiley, New York, 1996.

73. Smith, G.T., *CNC Machining Technology, Vols. 1, 2 & 3,* Springer Verlag, London, 1993.
74. Strasser, W., *Theory and Practice of Geometric Modelling,* Springer Verlag, 1989.
75. Su-chen Jonathan Lin, *Computer Numerical Control: From Programming to Networking*, Delmar Publishers, New York, 1994, ISBN: 0-8273-4715-4.
76. Talavage, J., and R. G., Hannam, *Flexible Manufacturing Systems: Applications, Design and Simulations,* Marcel Dekker, New York, 1988.
77. Teicholz, E., (Ed.), *CAD/CAM Handbook*, McGraw-Hill Book Co., New York, 1985.
78. Teicholz, E., W., Stawood and J., Tchijov, *Computer Integrated Manufacturing,* (Ed.) R.U. Ayres, Chapman & Hall, London, 1992.
79. Thyer, G.E., *Computer Numerical Control of machine Tools,* Heinemann, Oxford, 1988, ISBN 0-434-91959-4.
80. Valliere, *Computer Aided Design in Manufacturing,* Prentice-Hall, New Jersey, 1990, ISBN: 0-13-165408-X.
81. van Houten, F.J.A.M., *Manufacturing Interfaces*, Annals of CIRP, Vol. 41, 1992, pp. 899 – 910.
82. Vickers, G. W., M. H., Ly and R. G., Oetter, *Numerically Controlled Machine Tools,* Ellis Horwood, 1990, ISBN: 0-13-625526-4.
83. Viswanadham, N., and Y., Narahari, *Performance Modelling of Automated Manufacturing Systems,* Prentice-Hall, New Jersey, 1992.
84. Wakil, E. L., *Processes and Design for Manufacturing*, Prentice-Hall, New Jersey, 1989.
85. Warnecke, H.J., and R.D., Schraft, *Industrial Robots: Application Experience,* I.F.S. Publications, Bedford, 1982.
86. Watt Alan, *3D Computer Graphics*, Addison-Wesley, Harlow, England, 1993.
87. Womack, J. P., and Daniel T. Jones, *Lean Thinking: Banish Waste and Create Wealth in your Corporation,* Revised Edition, Free Press, New York, 2003.
88. Womack, J. P., Daniel T. Jones, and Daniel Roos, *The Machine that Changed the World: The Story of Lean Production,* Harper Perennial, New York, 1991.
89. Zeid, I., *CAD/CAM: Theory and Practice,* McGraw-Hill, New York, 1991.
90. Zeid, I., Mastering *CAD/CAM,* McGraw-Hill, New York, 2005.
91. King, J.R., (1980), Machine component grouping in Production Flow Analysis: An approach using rank order clustering algorithm, *International Journal of Production Research,* Vol. 18, pp. 213 – 232.
92. King, J.R., and V., Nakoranchai (1982), Machine component group formation in group technology: Review and Extension, *International Journal of Production Research, Vol. 20,* pp. 117 – 133.
93. Hollier, R. H., (1963), The layout of multi-product lines, *International Journal of Production Research,* Vol. 2, pp. 47 – 57.

GLOSSARY OF TERMS

2D—Graphics Display of a scene or an object along two axes *X* and *Y*.

2D 2 dimensional. Usually *X* and *Y*.

3D—Graphics Displayed representation of a scene or an object that appears to have three axes of *X*, *Y*, and *Z*.

3D 3 dimensional. Usually *X*, *Y* and *Z*.

Absolute Coordinates The location of a point in terms of distances and/or angles from a fixed origin.

Absolute System A numerical control system in which all positional dimensions, both input and feedback, are given with reference to a common datum point.

Access Time The time it takes to transfer an instruction or unit of data from computer memory to the processing unit of a computer.

Additive Colour A colour model associated with the RGB (red, green, blue) method of representing colour. Equal amounts of the primaries will combine to produce the perception of white light. This is normally used in video system/monitors.

Address The coded representation of a specific computer workstation or host computer used in transferring data through a network. A unique number is assigned to each computer.

AGP (Accelerated Graphics Port) A bus that provides a direct connection from the graphics card to the main memory. It allows for only one graphic card connector on the motherboard.

AI Artificial Intelligence

Algorithm A computational method for solving problems.

Aliasing Refers to the jagged lines or edges that can appear in computer drawn images. Aliasing occurs when smooth lines or edges in an image are drawn with pixels.

Alphanumeric A system of code that consists of the characters *A–Z* and numerals 0–9.

ALU Arithmetic and logic unit.

Analog Circuit A circuit in which the output varies as a continuous function of the input, as contrasted with digital circuit.

Analog Pertaining to a system that uses electrical voltage magnitudes or ratios to represent physical axis positions.

Annotation The process of inserting text or a special note, explanation or to provide relevant detail on a CAD/CAM drawing.

ANSI (American National Standards Institute) The American National Standards Institute (ANSI) is a privately funded, non-profit organization, which coordinates the development of voluntary standards in the United States and is the agency that approves standards (as American National Standards).

Anti-aliasing A technique for smoothing out jaggies—the jagged edges on diagonal lines and curves on-screen. To compensate, graphics cards blur the edges by adding various shades of grey or colour to surrounding pixels (this is called dithering).

API (Application Programming Interface) Provides standard documented access to software functions, allowing customers and third parties to develop and customize their own commercial applications.

APICS American Production and Inventory Control Society.

Application Profile A number of application protocols required for a specified task or industry sector. Associated with STEP.

Application Protocol (AP) Defines the context for the use of product data and specifies the use of the standard in that context to satisfy an industrial need. Associated with STEP.

Application A computer software program that performs specific functions such as page layout, word processing, accounting, drawing and spreadsheet formation.

APT Programming Automatically Programmed Tools. A universal computer assisted programming system for multi-axis contouring programming.

ASCII American Standard Code for Information Interchange. A data transmission code, which has been established as an American Standard by the American Standards Association. It is a code in which 7 bits are used to represent each character.

Aspect Ratio The shape of the display device on which an image will be viewed. The aspect ratio of a rendered image is expressed as the width of the image divided by its height.

Assembly A final gathering of piece-parts and subassemblies to make one unique assembled product.

Assembly Drawing A drawing, which can be created on the CAD system to represent a major subdivision of the product, or the complete product.

Assembly Modelling The process by which individual solid models are brought together to form an assembly model.

Associative Geometry A system that lets you place graphic elements based on a relationship (for example, parallel to) with existing graphic elements. Elements placed associatively maintain the relationship when the existing graphic element is manipulated.

Associativity Operating under a single integrated database structure, allows engineers to make changes in any application (i.e., design, drawing, manufacturing, assembly, mould, cabling, etc.), have those changes reflected instantly throughout all applications as well as in every deliverable (including drawings, bill-of-materials, NC tool paths, etc.), and vice versa.

A**utomatic Dimensioning** A CAD capability that will automatically compute and insert the dimensions of a design or drawing, or designated section of it.

Auxiliary Function A function of a CNC machine tool denoted by M-word address.

Axis A principal direction along which the relative movements of a tool or workpiece occur. Three linear axes, occurring at 90 degree angles from each other, named X, Y and Z.

Axis Inhihit A feature of an NC unit that enables the operator to withhold command information from a machine tool slide.

Axisymmetric A solid geometric entity that is symmetric and typically revolves about a common axis.

Backbone A central high-speed network that connects smaller, independent networks.

Background A feature which facilitates in the execution of low priority work when high priority work is not using the computer.

Backlash A relative movement between interacting mechanical parts as a result of looseness.

Ball End Mill Milling tool commonly used for creating complex surface machining operations. End of tool is rounded so that the radius on tool end equals half of tool diameter.

Base A number base is a quantity used implicitly to define some system of representing numbers by positional notation.

Batch A number of items being dealt with as a group.

Batch Processing A manufacturing operation in which a specified quantity of material is subject to a series of treatment steps. Also, a mode of computer operations in which each program is completed before the next is started.

Baud A unit of signalling speed equals to the number of discrete conditions or signal events per second; e.g., 1 bit per second in a train of binary signals.

BCL Basic Control Language. EIA ANSI 494.

Behind the Tape Reader A means of inputting data directly into a machine tool control unit from an external source connected behind the tape reader.

Benchmark A standard example against which measurements can be made.

Bezier Curves A quadratic (or greater) polynomial for describing complex curves and surfaces.

Bi-Directional Associativity The associative relationship between a 2D drawing and a 3D model. The associativity is where both the drawing and the model can update each other.

Bill of Materials A listing of all the parts that constitute an assembled product.

Binary A numbering system based on 2. Only the digits 0 and 1 are used when written.

Binary Code Code based on binary numbers expressed as either 1 or 0, true or false, on or off.

Binary Digit (BIT) A character used to represent one of the two digits in the binary number system, and the basic unit of information or data storage in a two-state device.

BIT Binary Digit A binary digit has only two possible states.

Bitmap Generally, a bitmap is associated with graphics objects. The bits are a direct representation of the picture image.

Blending Blending is the combining of two or more objects by adding them on a pixel-by-pixel basis.

Block A set of words, characters, digits or other elements handled as a unit. In a CNC part program it consists of one or more characters or rows across that collectively provide enough information for an operation.

Block Delete A function that permits selected blocks of code to be ignored by the control system, at the operator's discretion.

Boolean Algebra An algebra named after George Boole for logical operations.

Boundary A 3D outline of a data volume.

Boundary Evaluation An operation that generates a B-Rep solid from a CSG solid.

Bounded Objects In solid modelling, an object is considered bounded if it has a complete set of bounding surfaces and is restricted to occupying a finite volume.

B-rep (Boundary Representation) A database method that defines and stores a solid as a set of vertices, edges, and faces (points, lines, curves and surfaces) which encloses its volume completely.

Bridge A device that filters or passes traffic, based solely on the destination network address.

B-Spline (Bi-Cubic Spline) A sequence of parametric polynomial curves (typically quadratic or cubic polynomials) forming a smooth fit between a sequence of points in 3D space. The piece-wise defined curve maintains a level of mathematical continuity that is dependent upon the polynomial degree that is chosen. B-splines are used extensively in the mechanical design applications in the automotive and aerospace industries.

Buffer Storage A place in which information in a control system or computer is stored for use at a later time.

Bug A programmed error or oversight, a glitch.

BUS A conductor used for transmitting signals or power between elements.

BYTE A sequence of eight adjacent bits operated on as a unit.

Cache An area in the memory used for temporary storage. Cache is used to optimise operation and retrieval speeds.

CAD/CAM Computer Aided Design and Computer Aided Manufacturing.

CAD Computer Aided Design. Software for geometric design of parts.

CADD Computer Aided Drafting.

CAE Computer Aided Engineering. The use of computer and digital technology to support basic error checking, analysis, optimisation, manufacturability, etc., of a product design. Finite Element Analysis (FEA) is one example of CAE.

CAM Computer Aided Manufacturing. Software used for preparing the CNC part programs.

Canned Cycle A preset sequence of events initiated by a single ‘*G*’ coded command.

Cartesian Coordinates A three dimensional system whereby the position of a point can be defined with reference to a set of axes at right angles to each other.

CASE (ComputerAided Software Engineering) An umbrella term for a collection of tools and techniques, which are said to promise revolutionary gains in analyst and programmer productivity.

Cathode Ray Tube (CRT) A display device in which controlled electron beams are used to present alphanumeric or graphical data on a luminescent screen.

CCD (charged coupled device) An electronic memory made of a metal-oxide semiconductor (MOS) transistor that can store patterns of charges sequentially.

CD-ROM (Compact Disc Read Only Memory) The CD-ROM when used in computer can store text, sounds and images, as well as video information.

CE (Concurrent Engineering) A systematic approach to creating a product design that considers all elements of the product life cycle from conception of the design to disposal of the product.

Central Processing Unit (CPU) The portion of a computer system consisting of the arithmetic and control units and the working memory.

Chip A single piece of silicon cut from a slice by scribing or breaking. A chip can contain one or more circuits.

CIM Computer Integrated Manufacturing Automated use of computer and digital technology to completely integrate all manufacturing processes with engineering design.

Circular Interpolation A mode of contouring control, which uses the information, contained in a single block to produce an arc of a circle.

CISC Complex Instruction Set Computing.

CL DATA Processor output that contains information regarding cutter location.

Clearance Distance The distance between the tool and the work piece when the change is made from rapid approach to feed movement to avoid tool breakage.

CLFILE Cutter location file. A file of data generated from a machine tool path created on a CAD*I* CAM system or on an APT processor. The CL file contains *X*, *Y*, and *Z* coordinates and other NC information to be post-processed to program NC machines.

Client A computer or computer program that is one side of the client-server communication.

Client-Server Architecture An information-passing scheme that works as follows. A client program, such as Netscape, sends a request to a server. The server takes the request, processes it, and transfers the information to the client.

Clock A device, which generates periodic synchronisation signals.

Closed Loop A signal path in which outputs are fed back for comparison with desired values to regulate system behaviour.

CMM (Coordinate Measuring Machine) Computer controlled equipment used to inspect part dimensions.

CNC Computer (Computerised) Numerical Control A numerical control system wherein a dedicated, stored program computer is used to perform some or all of the basic numerical control functions.

Coincidence Refers to geometry that occupies the same spatial location. For example, coincident vertices are points that occupy the same *X*, *Y*, and *Z* coordinates. Coincident lines can have differing lengths while one occupies the same locations the other.

Command An operative order, which initiates a movement or a function.

Compile To generate a machine language program from a computer program written in a high-level source code.

Compression The process of removing irrelevant information and reducing unneeded space from a file in order to make the file smaller.

Concentric Having a common centre or origin point with varying radii.

Configuration A particular combination of a computer, software and hardware modules, and peripherals at a single installation and interconnected in such a way as to support certain requirements.

Conic Section Curve formed by the intersection of a plane with a cone.

Console The part of a computer system used by the operator for communication with the computer system.

Contouring A numerical control method where the control program generates a contour by keeping the cutting tool in constant contact with the work piece.

Controller Device attached to a machine tool, which reads and stores machine data file and passes movement information to the machine tool. Also controls other machine activities such as turning coolant on or off.

Coons Patch Surface defined by four boundary curves, tangents, corner twists, and blending functions. Curvature and continuity can be maintained between patches.

Coordinate System Geometric relation used to denote the location of points in 3D space. The most common is the rectangular coordinate system, whereby points are located by traversing the *X*, *Y* and Z-axes of 3D space.

Coordinates Ordered set of absolute or relative data values that specify a location in a coordinate system.

Coplanar Refers to two or more entities that lie on the same plane.

CPU Central Processing Unit of a computer. The memory and logic area of a computer that includes processing and execution of instructions.

CRT Cathode Ray Tube. A device that displays alphanumeric data to the machine operator.

CSG (Constructive Solids Geometry) A scheme for representing solid objects. It is a tree representing instances of solids and Boolean operations (union, intersection, difference).

Curve Smoothing The process of curve or surface approximation using polynomial equations to generate a curve that passes near, but always through, a set of control points or mesh vertices.

Cutter Diameter Compensation A system in which the programmed tool path may be altered to compensate for cutter diameter differences.

Cutter Offset The distance from the part surface to the axial centre of a cutter (the radius of the cutter).

Cutter Path The path defined by the centre of the cutter.

Cutting Plane A tool that slices through the 3D data exhibiting characteristics of a plane in the data, including scalar and vector measurements.

Cycle A sequence of operations that is repeated. The time the repetition requires is cycle time.

Cylinder A solid primitive defined as a right-circular cylinder. The ends are circular and of equal radius. The axis is normal to the ends.

Data Exchange (In CADCAM} Data Exchange covers all the issues of exchanging models between different CAD systems. Standard like IGES and STEP are parts of this issue.

Data Management Coordinated management of all aspects of electronic manipulation of machine readable data in a computer environment; includes data capture, communication, storage, retrieval and associated processing.

Database A collection of data. The collection is organized in a logical structure for the primary purpose of automating information.

Datum Point or plane of reference.

DBMS (Database Management System} Software designed to manipulate the information in a database. It can create, sort, display selected information, search for specific information, and perform many other tasks of a database.

DDA Digital Differential Analyser.

Debug To trouble shoot, detect, locate and remove mistakes from a program.

Decompression Process of returning a compressed file to its full bitmap.

Dedicated Process or system that is available for only one function or use.

Default An automatic decision that is made by computer software and hardware programs. The decision will automatically be carried out unless the user changes the default settings.

Detail Drawing The drawing of a single object design including graphics, dimensions, and annotations, complete enough for manufacturing.

Digit A single character in any numbering system.

DMIS (Dimension Measuring Interface Specification} Standard programming language of CMM machines.

DNC Direct Numerical Control. A system in which machine coded programs are introduced into the CNC controller from a remote computer.

Documentation A generic term for a wide variety of hard-copy or on-line reports, drawings, and lists to be used by various departments involved in any aspect of design to fabrication of a part.

DOS Disk Operating System.

Dot Pitch Dot pitch is the space between pixels. The smaller the number, the sharper the image will appear.

Download To transfer to your computer a copy of a file that resides on another computer.

Downstream Process All subsequent operations or processes performed on or with a solid model. Downstream processes include (but are not limited to) analysis, NC code generation, and rapid prototyping.

Downtime Time during which equipment is inoperable because of faults.

DPI (dots per inch} A measurement of linear resolution for a printer or scanner. For example, a resolution of 600 dpi means that there are 600 dots across and 600 dots down. A higher number of dots create a finer resolution.

Dragging Dynamically moving the virtual image of graphical entity across the display screen to a new location using a puck, mouse, or stylus.

DRAM Dynamic Random Access Memory is the memory at any location in a computer that can be accessed immediately for reading and writing operations.

Drawing An engineering document or digital data file(s) that discloses (directly or by reference), by means of graphic or textual presentations, or a combination of both, the physical and functional requirements of an item.

Drawing Conversion The process of moving engineering data from hardcopy or raster format to CAD format.

Drive An internal or external assembly that can read and/or write electronic data using disk-storage media.

Driver A small software program that links together the computer and its components and peripherals: printers, scanners and the monitor.

Dwell Time A timed delay of programmed or established duration used in specific machining operations.

DXF (Data Exchange Format) A translation format developed by AutoCAD to transfer geometry data to and from AutoCAD.

EBCDIC Extended binary coded decimal interchange code.

Edge A bounded line or curve that forms the intersection of two faces on the surface of an object.

EDO DRAM A type of DRAM that has enhanced readability in the Extended-Data-Out mode.
EIA Electronics industries association

Electrostatic printing Printing large-format prints in a process similar to, but not the same as, colour photocopiers.

Emulate The use of software to allow one device to imitate another.

Encoder A transducer that produces digital pulses based on mechanical displacement.

Encryption A method of ensuring data secrecy. The message is coded using a key available only to the sender and the receiver. The coded message is sent to the receiver and then decoded upon receipt.

End of Program A miscellaneous function (M02) that represents the completion of a programmed cycle.

End-Of-Block Character A character representing the end of a programmed block of information.

Entity Any information that is being displayed on an interactive CRT, which can be identified. Entities can consist of geometry (points, lines, circular arcs, conics, splines, surfaces, solids, etc.), or as text items (notes, dimensions, lists, tables, etc.), or as "information-adding" things (coordinate systems, surface normal vectors, etc.).

EPD (Electronic Product Definition} To concurrently create, manage, share, and reuse electronic product information in a collaborative environment throughout a product's life cycle and across a distributed value chain.

EPS (Encapsulated Postscript} A file type that allows the carrying of different information between software programs.

ETHERNET Xerox trademark name for a network cable system that allows communication between the workstations and servers connected to the network.

Executive Program A set of programming instructions that allows a CNC lathe to have the capability to perform lathe functions or a CNC mill to perform milling functions. A set of instructions designed to output specific functions.

Export Sending a model to a file (IGES, DXF, STEP, TIFF, etc.) so that it can be read or imported into another program.

Extrusion A process used in geometric modelling to convert 2D shapes into 3D shapes. A 3D object is created by displacing a copy of the 2D shape, than linking the copy to the original to form a closed, solid object.

Face A type of element used in geometric modelling. A face can be a flat, planar polygon or a curved, bounded surface. Some systems define a face as the bounded portion of an infinite surface region.

Family of Parts A collection of parts with similar shape, but differing in physical measurement.

FEA (Finite Element Analysis} A method used in CAD for determining the structural integrity of a mechanical part or physical construction under design by mathematical simulation of the part and its loading conditions.

Feature Software abstraction of the mechanical concept of hole, rib, slot, and pocket used to describe the model in a more functional way.

Featured-Based Modelling Performs functions that were previously performed using primitive Boolean operations. Example, the through-hole feature understands the rule that it must pass completely through the part and will do so no matter how the part changes.

Feed A programmed or manually established rate of movement of the cutting tool into the work piece for the required machining operation.

FEM Finite Element Modelling. A type of software made to compute and simulate the reaction of CAD models in respect of stress, magnetism, plastic injection, etc.

File Format The specific organization of data within a file. There are multiple raster and vector file formats, including JPEG, STEP, IGES and DXF.

File Server The local area networks allow users to share the peripherals (printers, modems, scanners) and thereby conserve their software cost. The file server is a device on the LAN where shared software is stored.

File An organized collection of relevant, orderly data.

Fillet Surface The transition surface that blends together two surfaces.

Firewall A computer system that sits between the Internet and a company's LAN. It is a means of automatically limiting what a company's computer system will pass along to outside computer systems. It acts as an active gateway to keep non-company entities from accessing company confidential data.

Firmware Programs or controlled instructions that are not changeable by the user and that are often held in ROM, Read Only Memory.

Fixed Cycle A preset sequence of events initiated by a single G coded command.

Flat Shading The flat shading method is also called constant shading. For rendering, it assigns a uniform colour throughout an entire polygon. This shading results in the lowest quality, an object surface with a faceted appearance and a visible underlying geometry that looks 'blocky'.

Floating Zero A characteristic of a machine control unit that allows the zero reference point to be established at any point of travel along an axis.

Form Features In solid modelling, parts of solid objects that can be specified in familiar engineering terms (e.g. fillets, slots, and through-holes).

Format Arrangement of data

FORTRAN An acronym for FORmula TRANslation. A high-level mathematical source language developed for scientific and engineering applications.

Free-Form Surface Surfaces that are not limited to mathematically simple linear or quadric surfaces.

FTP File Transfer Protocol is a protocol that allows the transfer of files from one computer to another.

Full Duplex Allows simultaneous transmission of information in both directions.

Functionality Refers to a set of system capabilities in terms of the functions they provide.

G Code Preparatory Function. An NC word addressed by the letter G and followed by a numeric value**.**

Gateway A device used to connect networks with radically different communications architectures.

Gauge Height A predetermined *Z*-axis clear plane retraction point along the *Z*-axis to which the cuter retreats allowing safe *XY* axis travel.

Generative Process Planning Computer-based process planning method whereby new process plans are created based on part and product information, as well as manufacturing information.

Geometric Modelling Methods used to create the exact geometric model of a part in the computer system

Geometry Elements that make up a model, such as points, lines, surfaces, solids, etc.

GIF Graphic Interchange Format, a commonly used graphics format developed by CompuServe.

Gouge Damaging the part due to tool motion entering the bounds of the desired finished part.

Gouraud Shading Gouraud shading is a process by which colour information is interpolated across the face of the polygon to determine the colours at each pixel.

Grey Code Binary code in which the successive values differ by only one bit.

Group A collection of elements.

Hard Copy A readable output of data on paper.

Hard Wired Having logic circuits interconnected on a back plane to give a fixed pattern of events.

Hertz Cycles per second or some repeated action per second - Hz.

Hidden Line A wire frame display option that displays only the lines that should be visible from the current view.

Hidden Surface Removal Hidden Surface Removal or visible surface determination entails displaying only those surfaces that are visible to a viewer because objects are a collection of surfaces or solids.

High-Level Language A problem-oriented programming language using words, symbols, and command statements, which closely resemble English language statements.

Home Position A fixed location in the basic coordinate axis of the machine tool.

HTML An acronym for Hypertext Mark-up Language, HTML codes are interpreted by the web browser to format documents in a particular way.

HTTP The abbreviation for Hypertext Transfer Protocol, a protocol used to transfer documents on the World-Wide Web.

Hybrid Solid Modeller A solid modelling database that actively maintains two or more substantially different representations of solid objects such as CSG and B-Rep.

Hypertext This term describes the system that allows documents to be cross-linked in such a way that the reader can explore related documents by clicking on a highlighted word or symbol.

IC Integrated Circuit.

IEEE Institute of Electrical and Electronics Engineers.

IGES (Initial Graphics Exchange Specification) A standard for exchanging mechanical design data between CAD systems.

Image Electronic representation of a document stored and displayed as a bitrnap.

Implicitly Defined Information that is defined by a situation rather than by explicit definition. In solid modelling, an edge defined as the intersection of two surfaces is implicitly defined.

Incremental Dimensioning A method of expressing a dimension with respect to previous point.

Independent Demand Demand for an item that occurs separately.

Inkjet Printer A type of printer that sprays tiny streams of quick-drying ink onto the paper. An inkjet printer produces high quality printing like that of a laser printer.

Intelligent Workstation A workstation in a system that can perform certain data processing functions in a stand-alone mode, independent of another computer.

Interface The communication that takes place between various elements in a system.

Interference Checking A CAD/CAM capability that enables mechanical designers to automatically examine intersection of objects within a 3D model.

Interoperability Related to the examination of the information exchange between two specific CAD systems, and the ability of each CAD system to use such information.

Interpolation A function of a control whereby data points are generated between given coordinate positions.

Inventory Stores or goods including raw materials, WIP, finished products, etc.

IP The abbreviation for Internet Protocol, IP refers to the set of communication standards that control communications activity on the Internet.

ISDN Integrated Services Digital Network: a high-speed digital phone system.

ISO 9000 A series of international standards that provides quality management guidance and identifies quality system elements that are necessary for quality assurance.

ISO International Organisation of Standardisation.

Jig A fixturing device used most often for drilling operations.

Jog A control function that momentarily operates a drive into the machine.

JPEG (JOINT PHOTOGRAPHIC EXPERTS GROUP} A widely accepted, international standard for compression of colour images.

Kaizen Japanese word for continuous improvement

Laminated Object Manufacturing (LOM) A patented process by Helisys of producing a physical prototype directly from a 3D surface or solid model.

LAN (Local Area Network) A network designed to connect devices over short distances. A data communications system that offers high-speed communication channels optimised for connecting information processing equipment over short distances. Consists of protocols and software to drive networks.

Layer A logical separation of data to be viewed individually or in combination.

Lead Time The time between ordering and receiving goods.

Leading Zeros Redundant zeros to the left of a number.

Legacy Data Existing data that has been acquired by an organization.

Linear Interpolation A control function whereby data points are generated between given coordinate positions to allow simultaneous movement one, two or more axes of motion in a linear path.

Lofting The process of fitting a surface between two or more profiles.

Loop Repetitive operations can be programmed in a continuous mode until the desired functions have been completed.

LSI Large scale integration used in connection with the integrated circuits.

Machining Centre Machine tools, normally numerically controlled, capable of automatically repeating many operations such as drilling, reaming, tapping, milling, and boring multiple faces on a work piece.

Macro A group of instructions that can be stored and recalled to solve a recurring problem.

Manual Data Input A mode of control that allows the operator to input data into the control system, the data input is identical to the data that can be input by other means such as tape or DNC.

Manual Part Programming The preparation of a manuscript in machine control language and format to define a sequence of commands for processing by a CNC machine.

MDI A mode of control that allows the operator to input data into the control system, the data input is identical to the data that can be input by other means such as tape or DNC.

Memory An organised collection of storage elements into which a unit of information consisting of binary digits can be stored and later be retrieved.

Meshing Action of computing a set of simple elements giving a good approximation of the designed part. A good meshing must be precise where the computation must be precise but with a minimum number of elements.

mil One thousandth of an inch.

MIPS Million Instructions Per Second.

Mirror Image The reversal of plus and minus values along an axis. Mirror imaging is used to make a left-handed part from a right-handed tool path.

Modal Pertaining to information that is retained by the system until new information that replaces it is obtained.

Model (in CAD) Precise term for the data managed by a CAD system to represent the parts. It is the virtual model of the designed part.

Model Space The geometric space defined in terms of three-dimensional coordinates where 3D modelling takes place.

Modem Modem stand for modulator/demodulator. It is a device that encodes data for transmission over a particular medium, such as telephone lines.

MPEG Motion Picture Experts Group the group that has defined the standards for compressed video transmission.

MRP (Material Requirements Planning) The computerized method for planning the utilization of a company's resources in manufacturing, including scheduling, vendor selection, material alternatives.

MSI Medium scale integration used in connection with the integrated circuits

Multimedia The discipline of integrating audio and pictorial data, in Information Technology, often for education and training applications.

NC Numerical Control. A technique of operating machine tools by software commands.

Nesting The arrangement of shapes within the stock area to maximize material use and to minimize scrap.

Network A system of computers, terminals and databases connected by communications lines, which allows the exchange of information and files.

Network Architecture The organizational concept enabling communications between data processing equipment at multiple locations. The network architecture specifies the processors and terminals and defines the protocols and software that must be used to accomplish accurate data communications.

Node A generic term for any device attached to a network. A node uses the network as a means of communication and has an address on the network.

NURBS (Non-Uniform Rational B-Splines) A mathematical description of a surface created by two (or more) B-splines.

Object An item that shares certain characteristics with other items. These are classes as an object and rules are then applied to the class.

Object Oriented A term frequently used to describe a philosophy that examines data objects rather than function when designing systems.

Offset A displacement in the axial direction of the tool equal to the difference between the actual tool length and the programmed tool length.

Open Order A customer order that has been launched into production and is in process.

Optional Stop A miscellaneous function similar to Program Stop (*M*01) except that the control ignores the command unless the operator has previously pushed a button to validate the command.

Overshoot The amount by which axis motion exceeds the target value.

PAN The process of moving the display window to view different areas of a drawing. This is done most commonly by depressing the pan button and moving the mouse.

Parabolic Interpolation Control of a cutter path by interpolation between three fixed points, with the assumption that the intermediate points are on a parabola.

Parametric CAD System A type of modem CAD system that lets you relate the geometry of different elements of a product. Parametric capability of some CAD systems to keep a directed set of relationship so that changes can be propagated to following constructions. In some cases these relationship will correspond to the design intended and some mechanical logic of the design.

PCB Printed circuit board.

PDES Acronym for Product Data Exchange using STEP.

PDES/STEP A set of standards under development for communicating a complete product model with sufficient information content that advanced CAD/CAM applications can interpret.

PDM Product data Management. A complete set of software to manage all data and files related to a product.

Pegging The process of tracing through the MRP records to identify how the change in one component record affects others.

Peripheral Equipment This term refers to external input or export devices that are physically not part of a computer's housing. Examples include printers, scanners, external drives, modems, monitors, etc.

PHIGS Programmer's Hierarchical Interface Graphics Standard.

Phong Shading Phong shading is a sophisticated smooth shading method, and is best known for its ability to render precise, realistic specula highlights.

Piezoelectric An inkjet printing technology that uses a mechanical-electric charge instead of heat to drive micro droplets through the nozzle.

Pixel The smallest unit of data in a digital image.

Planned Order A customer order that is planned for production but not yet released.

Platform A combination of computer hardware and an operating system.

Plotter A term that refers to the CAD origins of wide-format printers. A printer, that graphs computer output.

Port An outlet or connection location on a computer, which allows a peripheral device to operate. A communications port (COM port) allows the modem to operate, and a local port (LPT) enables the printer to operate.

Postprocessor A computer procedure, that takes CL file information and translates into NC machine specific programming terms.

PostScript® An Adobe programming language that enables text and graphic images to be output from different devices with consistent and predictable results.

Primitives A solid or surface that is not derived from other elements. A solid volume defined by simple standard geometrical shapes, such as a box, cone, and cylinder.

Process Planning The set of instructions for product manufacturing.

Product Cycle The total of all steps leading from concept of a product to its manufacture.

Product Data All engineering data, in processable form, necessary to define the geometry, the function, and the behaviour of an item over its entire life span.

Product Model A data model that contains the functions and physical characteristics of each unit of a product throughout its complete life cycle.

Product Structure The definition and organization of the objects that is appropriate for the design.

Productivity Efficiency as a ratio of output divided by input.

Projection The process of reducing three dimensions to two dimensions for display is called Projection.

Protocol A defined communication format that contains the control procedures required to facilitate data transfer across the link interfaces, and to and from the user's application programs.

Query A request for information entered while the computer system is processing.

Queue A waiting line.

RAM Random Access Memory needed for running programs such as Netscape.

Rapid Prototyping The process of producing a physical prototype directly for CAD 3D surface or solid modelling data by a number of patented processes such as SLA, LOM, FDM, SGC, or SLS.

Raster A two-dimensional array of elements, called pixels or picture elements, which when displayed on a screen or paper, form an image or representation of an original document.

Rasterization Translating an image into pixels.

Real-Time The description for an operating system that responds to an external event within a short and predictable time frame.

Reboot The process of turning a computer system or printer off and then back on again, to reload the software.

Relational The term used to describe a database system that models relationships between objects.

Relational Database A software program, which allows users to obtain information drawn from two or more databases that are made up of two-dimensional arrays of data.

Rendering Process of adding shading, colours, reflectivity, textures, etc. to a model to make it appear realistic.

Retract Machining move which removes the tool from the cut.

RGB (Red, Green, Blue) RGB is an additive colour model used in colour monitors, conventional photo film and paper to create full colour.

RISC Reduced Instruction Set Computing

Routing The processing steps needed to create a product.

Ruled Surface A surface generated by linear interpolation between two lines or curves, or a point and a line or curve.

SCSI (Small Computer System Interface) SCSI is a standard method of connecting devices to computers.

Sculptured Surface A free-form surface that is curved in more than one direction, typically by NURBS, Bezier, or other mathematical definitions.

SDRAM Synchronous DRAM is a type of DRAM to which reads or writes can be performed synchronously with the memory clock.

SERVER A computer dedicated to a single purpose for multiple users on a network. An example is the print server, which is dedicated to the handling of print requests.

SGC (Solid Ground Curing) A patented process by Cubital for producing a physical prototype directly from a 3D surface of solid model.

SGRAM Synchronous Graphics Random Access memory (SGRAM) is a type of memory that is optimised for graphics use.

SIMM Single in-line memory module.

SLA (Stereo Lithography) A patented process by 3D Systems for producing a physical prototype directly from a 3D surface of solid model.

SLS (Selective Laser Sintering) A patented process by DTM Corporation for producing a physical prototype directly from a 3D surface of solid model.

Software Programs, data files, procedures, rules, and any associated documentation pertaining to the operation of a computer system or of a computer application.

Solid Modelling Software capability of representing the sense of material with its familiar operations like drilling a hole or adding a slot. Solid modellers will be able to produce automatically cross sections and display with hidden lines removed

Spiral Cutting Type of milling cut where the tool begins in the geometric centre of the area being machined and spirals its way out to the outside bounds.

Spline Mathematical interpolation routine for describing curves or surfaces.

SQL (Structured Query Language) A language typically used to create, update and query relational databases.

STEP (Standard for the Exchange of Product Model Data) ISO standard 10303. An international standard under development, which will be used to describe a product in a neutral format over its complete life cycle in a hardware independent way.

SURFACE A boundary defining an exterior face of a solid model.

Surface Modelling Geometric modelling method that describes a part by its surfaces.

Tables A representation of data in a relational database. Information is arranged in columns and rows.

Task The smallest group of work that can be assigned to a workstation.

TCP-IP The basic protocols controlling applications on the Internet.

Thermal-transfer Printer A machine that digitally prints by transferring inks from a foil ribbon onto media such as paper or vinyl.

Throughput The total items going through a conversion process.

TIFF Tagged Image File Format, a graphic file format.

Time Fence The length of time that must pass through without changing the MPS.

Token Ring A network topology originated and promoted by IBM Corporation. At the conceptual level, a token is created for the transfer of data along the network. The token is passed from one networked device to another until the matching device is located. The data is then delivered.

Tolerance Stack-Up The accumulative tolerance of mating parts in an assembly.

Tool Path In numerical control, the path of a cutting tool as it passes over stock of material to produce desired shape.

Topological Data Data that includes the connectivity relationships among geometric components.

TORUS A solid primitive defined by the revolution of a circle about an axis in the plane of the circle. The axis must not pass through the centre of the circle, and must lie outside the circle.

Transformation Change of coordinates; a series of mathematical operations that act on output primitives and geometric attributes to convert them from modelling coordinates to device coordinates.

Tree A method of file storage. The structure comprises a top-level, and one or more sublevels which may in turn contain sublevels, etc.

Trimmed Surface A surface that has a distinct boundary. The boundary of a trimmed surface is typically where the surface intersects other surfaces.

UNIX An open, multiple-user operating system supported on a wide range of hardware and software. It is an operating system that supports multi-users and multi-tasking.

Variational Geometry A method of representing a solid model as a set of interrelated equations defining its shape and dimensions.

VDA A data exchange standard, developed for the German automotive industry to exchange complex curve and surface data between auto suppliers and manufacturers.

Vector An image plotted by lines on an *XY* axis. This image is different from a bitmap, which is composed of dots.

Vertex An *X*-*Y*-*Z* location (a point in space) used to define an element. All types of elements consist of one or more vertices.

Virtual Reality Class of software made to build a high-end representation of reality. It includes special effect like fire, animation of characters, high-end rendering.

VLSI Very Large Scale Integration.

WAN Wide Area Network. Private network facilities, usually offered by public telephone companies that link business network nodes.

WIP (Work In Process) Raw material undergoing change in the manufacturing process before it becomes finished inventory.

Wireframe Modelling A method of modelling geometry by using "wire." The geometry is described with lines, arcs, splines, etc., in 3D space.

Word Address Often called NC Words, such as *X*, *Y*, *Z*, *F*, *G*, *M*.

Work Centre A facility or a set of machines where a service is offered on a job.

Workflow The sequential management of document images through work queues and various application processes.

Workstation A computer that can serve only one operator at a time that commonly uses specialized software designed for engineering or other scientific applications.

WORM (Write Once, Read Many) An optical disk technology that permits one time creation of data but multiple read access.

WYSIWYG (What You See Is What You Get) An acronym meaning that a computer file's output is actually what is seen on the monitor.

Z-buffering A process of removing hidden surfaces using the depth value stored in the *Z*-buffer.

Zoom The process of magnifying the display of an image in a window to more closely inspect areas of a drawing.

Z-sorting A process of removing hidden surfaces by sorting polygons in back-to-front order prior to rendering.

STANDARDS PERTAINING TO CAM

1. ISO 10218-1:2006 - Robots for industrial environments—Safety requirements—Part 1: Robot
2. ISO 10303-108:2005 - Industrial automation systems and integration—Product data representation and exchange—Part 108: Integrated application resource: Parameterization and constraints for explicit geometric product models
3. ISO 10303-111:2007 - Industrial automation systems and integration—Product data representation and exchange—Part 111: Integrated application resource : Elements for the procedural modelling of solid shapes
4. ISO 10303-112:2006 - Industrial automation systems and integration—Product data representation and exchange—Part 112: Integrated application resource: Modelling commands for the exchange of procedurally represented 2D CAD models
5. ISO 10303-204:2002 - Industrial automation systems and integration—Product data representation and exchange—Part 204: Application protocol: Mechanical design using boundary representation
6. ISO 10303-240:2005 - Industrial automation systems and integration—Product data representation and exchange—Part 240: Application protocol: Process plans for machined products
7. ISO 11593:1996 - Manipulating industrial robots—Automatic end effector exchange systems—Vocabulary and presentation of characteristics
8. ISO 128-1:2003 - Technical drawings—General principles of presentation—Part 1: Introduction and index
9. ISO 128-21:1997 - Technical drawings—General principles of presentation—Part 21: Preparation of lines by CAD systems
10. ISO 13567-1:1998 - Technical product documentation—Organization and naming of layers for CAD—Part 1: Overview and principles
11. ISO 13567-2:1998 - Technical product documentation—Organization and naming of layers for CAD—Part 2: Concepts, format and codes used in construction documentation
12. ISO 14539:2000 - Manipulating industrial robots—Object handling with grasp-type grippers—Vocabulary and presentation of characteristics
13. ISO 14649-1:2003 - Industrial automation systems and integration—Physical device control—Data model for computerized numerical controllers—Part 1: Overview and fundamental principles

14. ISO 14649-10:2004 - Industrial automation systems and integration—Physical device control—Data model for computerized numerical controllers—Part 10: General process data
15. ISO 14649-11:2004 - Industrial automation systems and integration—Physical device control—Data model for computerized numerical controllers—Part 11: Process data for milling
16. ISO 14649-12:2005 - Industrial automation systems and integration—Physical device control—Data model for computerized numerical controllers—Part 12: Process data for turning
17. ISO 14649-121:2005 - Industrial automation systems and integration—Physical device control—Data model for computerized numerical controllers—Part 121: Tools for turning machines
18. ISO 1503:2008 - Spatial orientation and direction of movement—Ergonomic requirements
19. ISO 19439:2006 - Enterprise integration—Framework for enterprise modelling
20. ISO 20176:2006 - Road vehicles—H-point machine (HPM II)—Specifications and procedure for H-point determination
21. ISO 2806:1994 - Industrial automation systems—Numerical control of machines—Vocabulary
22. ISO 2972:1979 - Numerical control of machines—Symbols
23. ISO 3070-1:2007 - Machine tools—Test conditions for testing the accuracy of boring and milling machines with horizontal spindle—Part 1: Machines with fixed column and movable table
24. ISO 3070-2:2007 - Machine tools—Test conditions for testing the accuracy of boring and milling machines with horizontal spindle—Part 2: Machines with movable column and fixed table
25. ISO 3070-3:2007 - Machine tools—Test conditions for testing the accuracy of boring and milling machines with horizontal spindle—Part 3: Machines with movable column and movable table
26. ISO 3098-5:1997 - Technical product documentation—Lettering—Part 5: CAD lettering of the Latin alphabet, numerals and marks
27. ISO 3592:2000 - Industrial automation systems—Numerical control of machines—NC processor output—File structure and language format
28. ISO 4342:1985 - Numerical control of machines—NC processor input—Basic part program reference language
29. ISO 4343:2000 - Industrial automation systems—Numerical control of machines—NC processor output—Post processor commands
30. ISO 6582:1983 - Shipbuilding—Numerical control of machines—ESSI format
31. ISO 6983-1:1982 - Numerical control of machines—Program format and definition of address words—Part 1: Data format for positioning, line motion and contouring control systems
32. ISO 82045-5:2005 - Document management—Part 5: Application of metadata for the construction and facility management sector
33. ISO 8373:1994 - Manipulating industrial robots—Vocabulary
34. ISO 841:2001 - Industrial automation systems and integration—Numerical control of machines—Coordinate system and motion nomenclature
35. ISO 9283:1998 - Manipulating industrial robots—Performance criteria and related test methods
36. ISO 9409-1:2004 - Manipulating industrial robots—Mechanical interfaces—Part 1: Plates
37. ISO 9409-2:2002 - Manipulating industrial robots—Mechanical interfaces—Part 2: Shafts
38. ISO 9735-8:2002 - Electronic data interchange for administration, commerce and transport (EDIFACT)—Application level syntax rules (Syntax version number: 4, Syntax release number: 1)—Part 8: Associated data in EDI
39. ISO 9787:1999 - Manipulating industrial robots—Coordinate systems and motion nomenclatures
40. ISO 9946:1999 - Manipulating industrial robots—Presentation of characteristics
41. ISO/IEC 22091:2002 - Information technology—Streaming Lossless Data Compression algorithm (SLDC)

42. ISO/IEC 2382-24:1995 - Information technology—Vocabulary—Part 24: Computer-integrated manufacturing
43. ISO/IEC 24771:2009 - Information technology—Telecommunications and information exchange between systems—MAC/PHY standard for ad hoc wireless network to support QoS in an industrial work environment
44. ISO/IEC 9995-4:2009 - Information technology—Keyboard layouts for text and office systems—Part 4: Numeric section
45. ISO/PAS 26183:2006 - SASIG Product data quality guidelines for the global automotive industry
46. ISO/TR 13309:1995 - Manipulating industrial robots—Informative guide on test equipment and metrology methods of operation for robot performance evaluation in accordance with ISO 9283
47. ISO/TR 13567-3:1999 - Technical product documentation—Organization and naming of layers for CAD—Part 3: Application of ISO 13567-1 and ISO 13567-2
48. ISO/TR 6132:1981 - Numerical control of machines—Operational command and data format
49. ISO/TS 10303-1735:2006 - Industrial automation systems and integration—Product data representation and exchange—Part 1735: Application module: Pre defined datum 2D symbol
50. ISO/TS 10303-1736:2006 - Industrial automation systems and integration—Product data representation and exchange—Part 1736: Application module: Pre defined datum 3D symbol

INTERNET ADDRESSES

The following are some of the important internet addresses that will be useful for the readers to get further information on matters pertaining to those that are discussed in the book.

CNC Turning and Machining Centres

1. Ace Designers Products: www.acedesigners.co.in
2. Boehringer http://www.boehringer-werkzeugmaschinen.de/mag-boehringer.html
3. Bridgeport http://www.bpt.com/
4. Chiron America http://www.chironamerica.com/
5. Cincinnati Milacron http://cinmach.mag-ias.com/
6. Daewoo http://www.daewoomc.com/
7. Deckel Maho Gildemeister (http://www.gildemeister.com/)
8. Denford http://www.denford.ltd.uk/
9. Excell-o http://www.ex-cell-o.de/
10. Fadal http://www.fadal.com/
11. Giddings http://www.giddings.com/
12. Haas http://www.haascnc.com/
13. Hardinge http://www.hardingeus.com/
14. HMT Hindustan Machine Tools - Machine Tools & Industrial Machinery: http://www.hmtindia.com/
15. Hurco http://www.hurco.com/
16. Ingersoll http://www.ingersoll.com/products.htm
17. Kitamura http://www.kitamura-machinery.com/
18. KOMATSU http://www.komatsu-machinery.co.jp/HP/english/top_e.html
19. Makino http://www.makino.com/; http://www.makino.co.jp/en/index.html
20. Mandelli http://www.riellosistemi.it/riellosistemi/frontend/
21. Marubeni Citizen http://www.marucit.com/

22. Mikron http://www.gfac.com/
23. Mitsubishi Machine Tools http://www.mhimmt.com/
24. Monarch http://www.monarchmt.com/
25. Moriseiki http://www.moriseiki.com/english/index.html
26. OKK USA http://www.okkcorp.com/
27. Okuma America Corporation Home Page http://www.okuma.com/home.html
28. Star CNC Machine Tool Corp. http://www.starcnc.com/
29. Tornos http://www.tornos.us/
30. TOS Varnsdorf - http://www.tosvarnsdorf.cz/en/
31. Wasino http://www.amadawasino.com/
32. Yamazaki Mazak http://www.mazak.com/

CNC Controls

1. ANILAM http://www.anilam.com/
2. Autocon (Dynapath control) http://www.autocontech.com/
3. Fagor http://www.fagorautomation.mcc.es/
4. Fidia http://www.fidia.it/
5. GE Fanuc Automation http://www.gefanuc.com/
6. Heidenhain http://www.heidenhain.de/
7. MDSI http://www.mdsi2.com/index.html
8. Siemens

Other Machine Tools

1. AGIE EDM Machine tools http://www.gfac.com/
2. Amada's Punch presses and Laser cutting http://www.amada.com/site/default.asp?format=html&page=stamping.htm
3. ANCA - Grinder http://www.anca.com/
4. Brown & Sharpe http://brownandsharpe.com/
5. Charmilles EDM http://www.charmillesus.com/
6. Mitsubishi EDM http://www.mitsubishi-world.com/
7. Mitutoyo CMM: http://www.mitutoyo.com/
8. Renishaw Touch trigger probes http://www.renishaw.com/en/1030.aspx
9. Sodick EDM http://www.sodick.com/
10. Strippit Punch presses http://www.lvdgroup.com/
11. TRIMOS S.A.- Fabrique d'instruments de mesures: http://www.trimos.ch/
12. Trumpf Punch presses http://www.trumpf.com/
13. Welcome to THK America: http://www.thk.com/
14. Zeiss CMM http://www.zeiss.com/imt

Cutting Tool Manufacturers

1. Dormer Tools manufacturer of engineers cutting tools. http://www.dormertools.com/
2. Ingersoll International http://www.ingersollcuttingtools.com/en/index.htm
3. Iscar http://www.iscar.com/
4. Kennametal Inc http://www.kennametal.com/
5. Sandvik Tooling http://www.sandvik.com/
6. Seco Tools! http://www.secotools.com/wps/portal/corp
7. Sumitomo http://www.sumicarbide.com/
8. Taylor Hobson http://www.taylor-hobson.com/indexe.htm
9. Widia

CAD Software Links

1. ANVIL 5000 (MCS); http://www.mcsaz.com/support/a5k60updates.htm
2. AutoCAD (Autodesk); http://www.autodesk.com/
3. CADKEY (Baystate); http://www.cadkey.com/
4. CATIA (Dassault Système) http://www.dsweb.com/
5. CATIA-CADAM (Dassault/IBM); http://www-01.ibm.com/software/applications/plm/catiav4/
6. DesignWorkshop (Artifice Inc) http://www.artifice.com/
7. IntelliCAD http://www.intellicad.org/
8. Intergraph: http://www.intergraph.com/
9. IronCAD; http://www.ironcad.com/product/overview.html
10. Microstation (Bentley); http://www.bentley.com/
11. Pro engineer (Parametric Technology Corporation); http://www.ptc.com/
12. Rhino 3D (Robert McNeel & Associates) http://rhino3d.com/
13. SolidEdge (Unigraphics) http://www.plm.automation.siemens.com/en_us/products/velocity/solidedge/index.shtml
14. solidWorks http://www.solidworks.com/
15. TurboCAD (IMSI) http://www.turbocad.com/
16. Unigraphics http://www.plm.automation.siemens.com/en_us/products/nx/
17. Vdraft http://www.vdraft.com/
18. Vellum 3D (Ashlar Inc.) http://www.ashlar.com/
19. VX Vision http://www.vx.com/

CAM Software Links

1. AlphaCAM; http://www.alphacam.com/
2. ANVIL (MCS); http://www.mcsaz.com/
3. AS3000 (CAMSOFT);http://www.camsoftcorp.com/
4. ESPIRIT CAM (Auton);http://www.auton.it/
5. AutoPRO/CIMpro (INTERCIM);http://www.austinnc.com/
6. BobCADCAM;http://www.bobcad.com/

7. CAM Expert (RibbonSoft);http://www.ribbonsoft.com/
8. CAMWorks (Teksoft);Works inside SolidWorks; http://www.camworks.com/
9. CAPSMill and Turn (Cadem);http://www.cadem.com/
10. Cimatron;http://www.cimatron.com/
11. EdgeCAM; http://www.edgecam.com/
12. EXAPT ; http://www.exapt.de/en/products/NC-Planning/NC-Editor.htm
13. EZ-CAM (Bridgeport Machines) ;Feature based manufacturing http://www.bpt.com/
14. GeoPath (Solution Ware); http://www.solution-ware.com/
15. GIBBS (Gibbs); http://www.gibbsnc.com/
16. HI-MILL (Fidia); http://www.fidia.it/
17. HyperMILL-HyperCAM (Open Mind); http://www.openmind-tech.com/en/home/
18. INCAD (MTS); http://www.mts-cnc.com/
19. MASTERCAM (CNC Software);http://www.mastercam.com/
20. Mold flow Injection moulding design; http://www.moldflow.com/
21. PEPSCUT (CAMTEK); http://www.vero-software.com/
22. Predator Software (Editor, DNC & Verification); http://www.predator-software.com
23. Radpunch (Radan); http://www.radan.com/
24. Surfcam (Surfware);http://www.surfware.com/
25. Varimetrix;http://www.vx.com/
26. VisualMill (Mecsoft);http://www.mecsoft.com/

Other Useful Sites

1. Adams (http://www.mscsoftware.com/) Software for mechanism analysis
2. Algor FEM software: (http://www.algor.com/)
3. CNC Concepts THE site to visit. (http://cncci.com/)
4. Computer Aided Engineering: http://kernow.curtin.edu.au/cae.html
5. Free CAD links; http://www.cad-forum.com/
6. Metlfax Magazine On-line (http://www.metlfax.com/)
7. Modern Machine Shop Magazine; http://www.mmsonline.com/
8. Shareware on CNC: (http://www.ksc.com.hk/sharew/index.html)
9. STEP Tools; http://www.steptools.com/
10. Technical Education – CNC; http://www.tecedu.com/
11. The Society Of Manufacturing Engineers (http://www.sme.org/)
12. Thomas Register - (http://www.thomasregister.com/)

INDEX