




**COMPUTER NUMERICAL CONTROL**  
**From Programming to Networking**





# COMPUTER NUMERICAL CONTROL

## From Programming to Networking

S.C. Jonathan Lin



*A virtual one-volume encyclopedia on computer numerical control!*

NET  
\$49.95

# COMPUTER NUMERICAL CONTROL

From Programming  
to Networking

S.C. Jonathan Lin

Look no further! *Computer Numerical Control: From Programming to Networking* is a complete source for answering all of your computer numerical control questions.

**Don't miss these outstanding features:**

- covers all important aspects of CNC, including process planning, tooling, workholding, CAD/CAM NC programming, DNC networking, and programming milling and turning
- includes programming exercises and two complete CNC machining projects that offer "real world" practice
- reinforces text with numerous examples and over 800 illustrations and photos
- incorporates both English and metric measurement units throughout

Also available from Delmar Publishers Inc.™

*Fundamentals of Computer Numerical Control, 3E* • Luggen • Order # 0-8273-6496-2

*Learning Computer Numerical Control* • Janke • Order # 0-8273-4536-4



DELMAR PUBLISHERS INC.™

Page 3 of 90

03  
03  
66

COMPUTER NUMERICAL CONTROL

LIN S 3791 Autocad  
868312 SC 1 92834 VARA  
FF01 48 256 46

RA V AMS

Ex. 1011

# **Computer Numerical Control: Essentials in Programming and Networking**



# **Computer Numerical Control: Essentials in Programming and Networking**

**Su-Chen Jonathon Lin  
Professor of Computer-Aided Manufacturing  
Eastern Michigan University**

**I T P<sup>TM</sup>**



**Delmar Publishers Inc.<sup>TM</sup>**

## NOTICE TO THE READER

Publisher does not warrant or guarantee any of the products described herein or perform any independent analysis in connection with any of the product information contained herein. Publisher does not assume, and expressly disclaims, any obligation to obtain and include information other than that provided to it by the manufacturer.

The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instructions contained herein, the reader willingly assumes all risks in connection with such instructions.

The publisher makes no representations or warranties of any kind, including but not limited to, the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and the publisher takes no responsibility with respect to such material. The publisher shall not be liable for any special, consequential or exemplary damages resulting, in whole or in part, from readers' use of, or reliance upon, this material.

### Delmar Staff:

Senior Acquisitions Editor: Vernon R. Anthony

Editorial Assistant: Alison Foster

Project Editor: Elena M. Mauceri

Production Coordinator: Karen Smith

Art/Design Coordinator: Cheri Plasse

For information address Delmar Publishers Inc.

3 Columbia Circle Drive, Box 15-015

Albany, New York 12203-5015

Copyright © 1994

By Delmar Publishers Inc.

The trademark ITP is used under license.

All rights reserved. No part of this work covered by the copyright hereon may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, and information storage and retrieval systems—without written permission of the publisher.

Printed in the United States of America

Published simultaneously in Canada

by Nelson Canada,

a division of The Thomson Corporation

10 9 8 7 6 5 4 3 2 1 XX 99 98 97 96 95 94

Lin, Su-Chen Jonathon.

Computer numerical control : essentials in programming and networking / Su-Chen Jonathon Lin.

p. cm.

Includes index.

ISBN 0-8273-4715-4

1. Machine-tools—Numerical control. I. Title.

TJ1189.L56 1994

621.9'023—dc20

93-5254

CIP



## Additional Titles Available

*Learning Computer Numerical Control*/ Janke

ISBN: 0-8273-4536-4

*Fundamentals of Computer Numerical Control, 3E*/ Luggen

ISBN: 0-8273-6496-2

*Computer Numerical Control: Concepts and Programming, 2E*/ Seames

ISBN: 0-8273-3782-5

To request more information on these publications, contact your local bookstore, or call or write to:

Delmar Publishers Inc.

3 Columbia Circle

P.O. Box 15015

Albany, NY 12212-5015

Phone: 1-800-347-7707 · 1-518-464-3500 · Fax: 1-518-464-0301

# Contents

<b>Preface</b>	<b>xi</b>	<b>Chapter 4. Machine Axes and Coordinate Systems</b>	<b>75</b>
<b>Acknowledgments</b>	<b>xiii</b>	4.1 Machine Axis Identification	75
<b>PART I. Essentials of CNC Technology</b>	<b>1</b>	4.2 Coordinate Systems	80
<b>Chapter 1. Introduction to Numerical Control</b>	<b>2</b>	4.3 Positioning Systems	86
1.1 What Is Numerical Control?	2	4.4 Reference Coordinates	89
1.2 Historical Perspective of Numerical Control	3	Review Questions	92
1.3 Basic Elements of a CNC System	6	<b>Chapter 5. NC Code Systems</b>	<b>94</b>
1.4 Computer Numerical Control	8	5.1 Number Bases	94
1.5 Advantages of CNC	11	5.2 NC Program Storage Media	97
1.6 Limitations of CNC	12	5.3 Tape Coding Systems	99
1.7 Applications of Computer Numerical Control	12	5.4 NC Codes	102
1.8 Future of CNC Technology	20	5.5 NC Words	104
Review Questions	21	5.6 Tape Formats	109
<b>Chapter 2. NC Systems</b>	<b>23</b>	Review Questions	111
2.1 Mode of the NC Control Systems	23	<b>Chapter 6. Planning for NC Operations</b>	<b>113</b>
2.2 CNC Interpolation	28	6.1 NC Operation	113
2.3 Open-Loop and Closed-Control Systems	33	6.2 Process Planning for NC Operations	114
Review Questions	36	6.3 Machining Layout	117
<b>Chapter 3. NC/CNC Control Hardware and Operating Software</b>	<b>38</b>	6.4 Study of the Part Drawing	117
3.1 NC versus CNC Systems	38	6.5 Considerations for Raw Material	121
3.2 The CNC Machine Control Unit	40	6.6 Part Features Identification and Processes Selection	122
3.3 CNC Control Selection	48	6.7 Processes Sequencing	130
3.4 CNC Software	52	6.8 Tool Path Planning	132
3.5 Information Processing of the CNC System	56	6.9 Significant Points of the Workpiece	132
3.6 Feed Drives	56	6.10 Machining Parameters	136
Review Questions	73	6.11 Machining Parameters for Turning	139
		6.12 Machining Parameters for Milling	144
		6.13 Machining Parameters for Drilling	147
		6.14 NC Documentation	152
		Review Questions	162



<b>Chapter 7. Tooling for NC</b>	<b>165</b>	10.1 Work Coordinate Compensation (G54–G59)	260
7.1 Tooling System for Milling	165	10.2 Cutter Diameter (Radius) Compensation	264
7.2 Tooling System for Turning	172	10.3 Tool Offset	290
7.3 Tooling Strategies for CNC Turning Machines	172	Review Questions	294
7.4 Identification and Selection of Indexable Inserts	174	<b>Chapter 11. Fixed Cycles in Milling</b>	<b>298</b>
7.5 Identification and Selection of Toolholders	179	11.1 Standard Fixed Cycles	298
Review Questions	188	11.2 Special Fixed Cycles	314
<b>Chapter 8. Workholding for NC</b>	<b>189</b>	11.3 Point Pattern Cycles	317
8.1 Workholding Principles of NC	189	11.4 Bridgeport Special Fixed Cycles	323
8.2 Types of Workholders	190	11.5 User-Defined Fixed Cycles	326
8.3 Workpiece Locating Principle	190	Review Questions	329
8.4 Locating Devices	192	<b>Chapter 12. Repetitive Programming: Loops, Subprograms and Macros</b>	<b>330</b>
8.5 Clamping Principle	195	12.1 Loops	330
8.6 Manual Clamping Devices	196	12.2 Subprograms or Subroutines	333
8.7 Power Clamping	203	12.3 Macros	341
8.8 Clamping Methods	206	12.4 Macro Programming Examples	349
8.9 Modular Workholding	208	Review Questions	366
8.10 Workholding Devices for CNC Turning Machines	214	<b>Chapter 13. Turning Programming Fundamentals</b>	<b>368</b>
Review Questions	219	13.1 CNC Turning Machines and Their Controlled Axes	368
<b>PART II. Manual CNC Programming</b>	<b>221</b>	13.2 Turning Processes	370
<b>Chapter 9. Milling Programming Fundamentals</b>	<b>222</b>	13.3 Preparatory Functions for Turning	371
9.1 Tool Motion Commands	223	13.4 Tool Function	372
9.2 Plane Selection	240	13.5 Feed Function	373
9.3 Positioning Systems (G90 and G91)	242	13.6 Spindle Speed Function	373
9.4 Input Unit Selection (G70 and G71)	244	13.7 Miscellaneous Functions	375
9.5 Absolute Zero Setting (G92)	244	13.8 Reference Point and Coordinate System	376
9.6 Reference Point (Origin) Return (G28, G29, and G30)	247	13.9 Diameter and Radius Programming	379
9.7 Tool Selection and Change	251	13.10 Positioning Systems (Absolute versus Incremental)	380
9.8 Feed Selection and Input	252	13.11 Input Unit System Selection (Inch versus Metric)	381
9.9 Spindle Speed Selection and Control	252	13.12 Tool Motion Commands	382
9.10 Miscellaneous Functions	252	13.13 Thread Cutting	396
9.11 Programming Examples	253	13.14 Reference Point Return	407
Review Questions	257	13.15 Programming Examples	408
<b>Chapter 10. Compensation and Offset in Milling</b>	<b>260</b>	Review Questions	411

<b>Chapter 14. Compensation and Offset in Turning</b>	<b>415</b>	18.2 Controlling Surfaces	553
14.1 Tool Offset	415	18.3 Check Surface Modifiers	555
14.2 Tool Nose Radius Compensation	418	18.4 Start-Up Statements	555
14.3 Direction of Imaginary Tool Nose	422	18.5 Continuous-Path Programming	560
14.4 Tool Nose Radius Compensation Commands (G40, G41, and G42)	423	18.6 Tolerance Statements	564
14.5 Start-Up of TNR Compensation	425	18.7 Tool Motion Programming	
14.6 Tool Path during TNR Compensation Mode	426	Examples	566
14.7 TNR Compensation Cancel	426	Review Questions	571
14.8 Notes on TNR Compensation	429	<b>Chapter 19. Postprocessor Statements</b>	<b>574</b>
14.9 Programming Examples	430	19.1 Postprocessor Statements	574
Review Questions	433	19.2 Auxiliary Statements	582
<b>Chapter 15. Fixed Cycles in Turning</b>	<b>434</b>	Review Questions	584
15.1 Single Fixed Cycles	434	<b>Chapter 20. Generation and Execution of APT Programs</b>	<b>586</b>
15.2 Multiple Repetitive Cycles	444	20.1 Structure of an APT Program	586
15.3 Hole Operating Cycles (G80-G89)	455	20.2 Examples of APT Programs	587
15.4 Programming Examples	455	20.3 The APT Processor	595
Review Questions	458	20.4 Processing of APT Source Programs	597
<b>Chapter 16. Integration of NC Planning and Programming</b>	<b>461</b>	Review Questions	602
16.1 Turning Project	461	<b>Chapter 21. Advanced Geometry Definitions</b>	<b>603</b>
16.2 Milling Project	477	21.1 General Conic Definitions	603
<b>PART III. Computer-Assisted Programming in APT</b>	<b>501</b>	21.2 Loft Conics	611
<b>Chapter 17. APT: Basic Geometry Definition</b>	<b>502</b>	21.3 Tabulated Cylinder Definitions	613
17.1 Geometry Definition Statements	502	21.4 Ruled Surface Definitions	618
17.2 Point Definitions	504	21.5 Quadric Surface Definitions	622
17.3 Line Definitions	511	21.6 Circular Cylinder Definitions	625
17.4 Circle Definitions	520	21.7 Cone Definitions	628
17.5 Plane Definitions	530	21.8 Sphere Definitions	629
17.6 Geometry Definition Examples	540	21.9 Vector Definitions	632
Review Questions	545	21.10 Pattern Definitions	638
<b>Chapter 18. APT: Cutter Motion Definition</b>	<b>550</b>	21.11 Programming Examples	647
18.1 Point-to-Point Motion Programming	550	Review Questions	649
		<b>Chapter 22. Advanced Tool Motion Statements</b>	<b>652</b>
		22.1 Start-Up Statements	652
		22.2 Part Surface Statements	659
		22.3 Cutter-to-Surface Relationships	661
		22.4 Multiple Intersections	663



22.5	Multiple Check Surfaces	666	25.6	Programming Example for Turning	799
22.6	THICK Statement	668		Review Questions	801
22.7	POCKET Statement	671			
22.8	CYCLE Statements	674	<b>Chapter 26. 3-D Surface Tool Path</b>		
22.9	Macro Programming	679	<b>Fundamentals</b>	<b>805</b>	
22.10	Programming Examples	683	26.1	3-D Computer Models	805
	Review Questions	685	26.2	Profile Elements of Surfaces	807
<b>Chapter 23. Computation, Repetitive Programming, and Matrix Transformation Programming in APT</b>	<b>686</b>		26.3	Primitive Surfaces	807
23.1	Computation	686	26.4	Swept Surfaces	808
23.2	Repetitive Programming	692	26.5	Sculptured Surfaces	811
23.3	Looping Programming Using LOOPST and LOOPND Statements	701	26.6	Surface Editing Modules	813
23.4	Matrix Transformations	702	26.7	NC Parameters for 3-D Surfacing	816
23.5	Matrix Definitions	706		Review Questions	821
23.6	Geometry Transformation	715	<b>PART V. CNC Networking</b>	<b>823</b>	
23.7	TRACUT Statement	717	<b>Chapter 27. Communications and Computer Networking Fundamentals for DNC</b>	<b>824</b>	
23.8	COPY Statement	718	27.1	Data Communications	824
23.9	Programming Examples	724	27.2	Local Area Networks	829
	Review Questions	730	27.3	Network Hardware	837
<b>PART IV. CAD/CAM NC Programming</b>	<b>735</b>		27.4	Network Operating Systems	837
<b>Chapter 24. Introduction to Computer-Aided Design and Its Industrial Applications</b>	<b>736</b>		27.5	IBM and Digital Network Architecture	838
24.1	CAD Systems, Peripherals and Accessories	737	27.6	Computer Network Interconnectivity	841
24.2	Basic Geometrical Primitives for CAD	744		Review Questions	841
24.3	Three-Dimensional Modeling	748	<b>Chapter 28. DNC Implementation</b>	<b>843</b>	
24.4	Drawing Translations	753	28.1	NC Programs Generation and Transfer Methods	843
24.5	Drawing Conversion	758	28.2	Direct Numerical Control	844
24.6	Industrial Applications	759	28.3	Distributive Numerical Control	845
	Review Questions	766	28.4	DNC Configurations	847
<b>Chapter 25. 2-D Tool Path Fundamentals</b>	<b>768</b>		28.5	Connecting NC/CNC Machines to DNC Systems	851
25.1	NC Parameters	768	28.6	DNC System Hardware Components	853
25.2	Boundary Definition	776	28.7	DNC Software	853
25.3	NC Modules for Milling	778	28.8	DNC Selection Criteria	855
25.4	Programming Procedure and Example for Milling	787		Review Questions	856
25.5	NC Modules for Turning	792	<b>References</b>	<b>857</b>	
			<b>Index</b>	<b>859</b>	

# Preface

The advent of modern computer technology and a new generation of manufacturing equipment, particularly the computer numerical control (CNC) machine, have brought enormous changes to the manufacturing industry. To compete in the global market, most manufacturers have adopted the approach of computer-aided manufacturing (CAM), computer-integrated manufacturing (CIM), or flexible manufacturing systems (FMS) to efficiently and economically produce the product. These approaches use all kinds of programmable devices and systems, including CNC machines, CAD/CAM systems, robots, computers, and programmable logic controllers. CNC machines are the nucleus of CAM, CIM, and FMS. The global evidence shows that the population of CNC machines has been expanding at an amazing rate.

The ever-increasing use of CNC machines in the manufacturing industry has created a demand for personnel who are knowledgeable about CNC technology, capable of preparing part programs, and competent in managing CNC operations. Responding to this urgent need, dedicated CNC courses and related courses on subjects such as CAM and automation are being established in colleges, universities, and industrial organizations.

This book has been prepared as a comprehensive text and reference for the study of computer numerical control, with emphasis on the essentials of CNC technology and programming. The book is virtually a one-volume encyclopedia on modern CNC technology. It is written for students of industrial technology, engineering technology, and engineering, as well as for persons with a technical background, such as manufacturing engineers and technicians in industry. The book is written at the undergraduate level and is suitable for a two-semester course. It is appropriate for industrial technology, engineering technology, and engineering courses in universities and colleges, junior colleges, and trade schools, as well as for courses offered by CNC machine tool manufacturers.

The book is organized into the following five parts:

Part I: Essentials of CNC Technology (Chapters 1 through 8)

Part II: Manual CNC Programming (Chapters 9 through 16)

Part III: Computer-Assisted Programming in APT (Chapters 17 through 23)

Part IV: CAD/CAM NC Programming (Chapters 24 through 26)

Part V: CNC Networking (Chapters 27 and 28)

A two-semester course is adequate to cover most of the material. The depth and choice of topic coverage and projects may vary based on the particular curriculum. The following is a list of suggested topics to be covered in two semesters:

First semester      Part I (Chapters 1 through 8)

                                 Part II (Chapters 9, 10, 11, 13, 14, and 16)

- Part III (Chapters 17 through 20)
- Part IV (Chapters 24 and 25)
- Second semester
  - Part II (Chapters 12 and 15)
  - Part III (Chapters 21 through 23)
  - Part IV (Chapter 26)
  - Part V (Chapters 27 and 28)

The chapters covered in the first semester should be adequate for curricula in which the CNC course is offered only in one semester.

Although Fanuc controller systems are used to develop the chapters on manual programming in Part II, the programming principles can be applied to other systems with very little effort.

Both metric and English units are used in the book.

Chapter 24 was written by Dr. Fuh-Cwo (Tony) Shiue, a good friend and colleague of many years.

Su-Chen Jonathon Lin

# Acknowledgments

The author is indebted to many persons and organizations for their help and contributions in making this book idea a reality. I wish to express my sincere appreciation to the many students, teachers, and industrial personnel who reviewed all or part of the book and made many valuable suggestions for its improvement.

Special thanks are due to Mr. Jia Fei and Mr. Lianggen (Jeff) Zheng, who prepared several hundred illustrations in AutoCAD. Without their great skill, care, and amazing speed, the book would never have been finished.

Thanks are also extended to the Delmar staff for their patience and professional help. I owe a special debt of gratitude to Michael McDermott and Vernon Anthony for their effort in motivating me to start and complete this project.

I am greatly indebted to my dear wife, Grace, for her inexhaustible support, patience, and encouragement. I am also grateful to my son, Andrew, and daughter, Carol, for permitting me to spend numerous days and nights in the office preparing the manuscript.

I also greatly appreciate the assistance of the following firms and their personnel, who provided CAD/CAM NC software, technical information, and illustrations for this text.

ACMA Computer Inc.  
American SIP Corp.  
Becker Publishing Company, Inc.  
Bridgeport Machines, Inc.  
Carboloy, Inc.  
Carr Lane Manufacturing Co.  
Cassell PLC  
Cincinnati Milacron  
CNC Software, Inc.  
Collins Tool Corp.  
Forkardt Inc.  
Hardinge Brothers, Inc.  
Hewlett-Packard Company  
*Hydraulics and Pneumatics* magazine  
Industrial Text Company  
Ingersoll GmbH, Inc.  
Leadwell Manufacturing, Inc.  
Macmillan Publishing Company

xiii

McGraw-Hill Publishing Company  
Mid-State Machine Products  
Mitsubishi Electric Industrial Control, Inc.  
Mori Seiki Co., Ltd.  
Numeridex CAM Division, Automation Intelligence, Inc.  
Peddinghaus Corp.  
Pitman Publishing Limited  
Point Control, Inc.  
Royal Machine and Tool Corp.  
3Refix Technology  
Valenite Corp.

I also wish to thank the following individuals, who reviewed the book in preparation for this new edition. Their insights and criticisms were extremely helpful.

Samuel Owusu-Ofori  
Dept. of Mechanical Engineering  
North Carolina A&T State University  
Greensboro, North Carolina

Brad C. Harriger  
Purdue University  
W. Lafayette, Indiana

Henry R. Putz  
Center for CNC Education  
Worthington, Ohio

Charles R. Mabbott, Sr.  
Canton, Michigan

Larry T. Williams  
Machine Tool Dept.  
Linn Technical College  
Linn, Missouri

James C. Meide  
Hennepin Technical College  
Brooklyn Park, Minnesota

Gerald R. Brown  
Portsmouth, New Hampshire

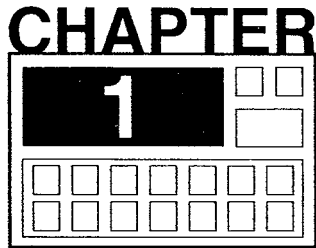
Jeff Jackson  
Industrial Engineer  
Howell, Michigan

Dr. Chin-Sheng Chen  
Dept. Of Industrial Engineering  
Florida International University  
Miami, Florida



**PART I**

# **Essentials of CNC Technology**



# Introduction to Numerical Control

---

Numerical control (NC) is one of the most important single elements in the transformation of the manufacturing industry since the first industrial revolution. Since its inception in the early 1950s, NC technology has undergone several major developments in both hardware and software. Today over one hundred and sixty thousand NC machine tools are used in twenty thousand manufacturing plants in the United States. NC machine tools represent about 6 percent of the total U.S. machine tool population, and the percentage has steadily increased in recent years. In NC user plants, NC machines account for about 20 percent of the machine tools on plant floors. However, they account for approximately 60 percent of all machining activities. Today's computer numerical control (CNC) machine tools have become the major driving force for American manufacturers to improve productivity in order to survive enormous foreign competition.

This chapter presents an overview of CNC technology, including definition, historical perspective, basic elements, features of CNC, advantages and limitations, CNC applications, and the future of CNC.

---

## 1.1 WHAT IS NUMERICAL CONTROL?

Numerical control has been defined by the Electronic Industries Association (EIA) as “a system in which actions are controlled by the direct insertion of numerical data at some point. The system must automatically interpret at least some portion of this data.” More precisely, numerical control can be considered as a versatile form of programmable automation in which the machine tool is controlled by a series of coded instructions consisting of letters, numbers, punctuation marks, and other symbols. These coded instructions are converted into two types of control signals: pulses of electric output signals and on/off control signals. Pulses of output signals implement the positioning of the machine spindle relative to the workpiece and its speed. Functions of the on/off signals include (1) control of the speed and direction of the spindle rotation, (2) control of coolant supply, (3) selection of the cutting tool, and (4) others, such as stop, optional stop, and automatic clamping and unclamping.

## 1.2 HISTORICAL PERSPECTIVE OF NUMERICAL CONTROL

The idea of using programmed information for controlling a device can be traced back to the fourteenth century, when pegged cylinders were used to control the movement of ornamental figures on church clocks (Kief and Olling, 1985). Many creative inventions have made significant contributions to CNC technology. Here we briefly outline some milestones in the course of numerical control development.

### 1.2.1 Events before the Dawn of the NC Era

The actual concept of numerical control might have originated as early as 1725 in England, where systematized punched cards were used to control the cloth pattern of knitting machine operations. In 1808 Joseph M. Jacquard employed punched holes in sheet metal cards to automatically control the patterns made by weaving machines. The activation of a needle was determined by the presence or absence of its associated hole. Around 1863 M. Fourneaux patented the automatic player piano, which used air passing through holes in a perforated paper drum approximately 4 1/2 in. wide to control the keyboard mechanism in a preprogrammed sequence. In 1912 Emmanuel Scheyer of New York invented the Kin-autograph machine, which used punched paper tape to cut cloth. He proclaimed that his device could easily be used to control various types of machinery, including multiple-axis machine tools. In 1926 Max Schenker of Switzerland patented the automatic lathe, which used punched cards for the storage of information. His device was able to mathematically program the lathe to control both direction and speed.

### 1.2.2 The Birth of the NC Machine

In 1947 John Parsons of the Parsons Corporation, based in Traverse City, Michigan, began experimenting with the idea of using three-axis curvature data to control machine tool motion for the complex machining of aircraft components. He set up a successful demonstration of the mathematical approach to contour cutting for the Air Force in December 1948. In June 1949 Parsons was awarded an Air Force research contract of \$200,000 for a 21-month project to develop an automatic contour cutting machine. Shortly after receiving the contract, Parsons was joined by the Servomechanisms Laboratory of the Massachusetts Institute of Technology (MIT) as a subcontractor. In 1951 MIT was awarded the major contract from the Air Force that resulted in the building of the first numerically controlled machine tool, a Cincinnati Hydrotel vertical-spindle milling machine, in 1952. The machine control unit (MCU), built with electron vacuum tubes, took up more floor space than the machine. It received machining data from binary-coded punch tapes to generate movements of three axes in a linear interpolation manner. The effort over the following three years was mainly devoted to hardware refinements and mathematical techniques for tape preparation.

In 1955 the Subcommittee on Numerical Control was organized by the Airframe Manufacturing Equipment Committee of the Aerospace Industries Association (AIA) to investigate the NC systems under development. The subcommittee made two recommendations to the Air Force: that forthcoming machines be equipped with numerical control and

that a standardization of NC systems be established. The Air Force subsequently granted \$35 million for the manufacture of 100 NC milling machines, which were distributed to manufacturers for producing aircraft and missile components. Standardization activities centered around hardware and software in three aspects: type and format of control input medium, system configuration for interpolation, and part-programming languages.

### **1.2.3 Programming Languages Development**

Following the successful development of the first NC machine, MIT began work on a computerized NC programming language called Automatically Programmed Tool (APT) in 1954. It was a symbol language capable of defining part geometries and cutter paths for simple workpieces on a computer. APT II was released from MIT in 1958. This was a more advanced version that ran on the IBM 704 computer. In 1961 another, more powerful programming language, APT III, was released. AIA awarded a project to the Illinois Institute of Technology Research Institute (IITRI) to further develop and maintain the APT III language. Expansion of the APT III system resulted in the release of APT IV, which includes more complex surface definitions for machining. APT has become the standard NC programming language for machining complex parts. The majority of today's computer-aided design/computer-aided manufacturing (CAD/CAM) NC systems are APT-based.

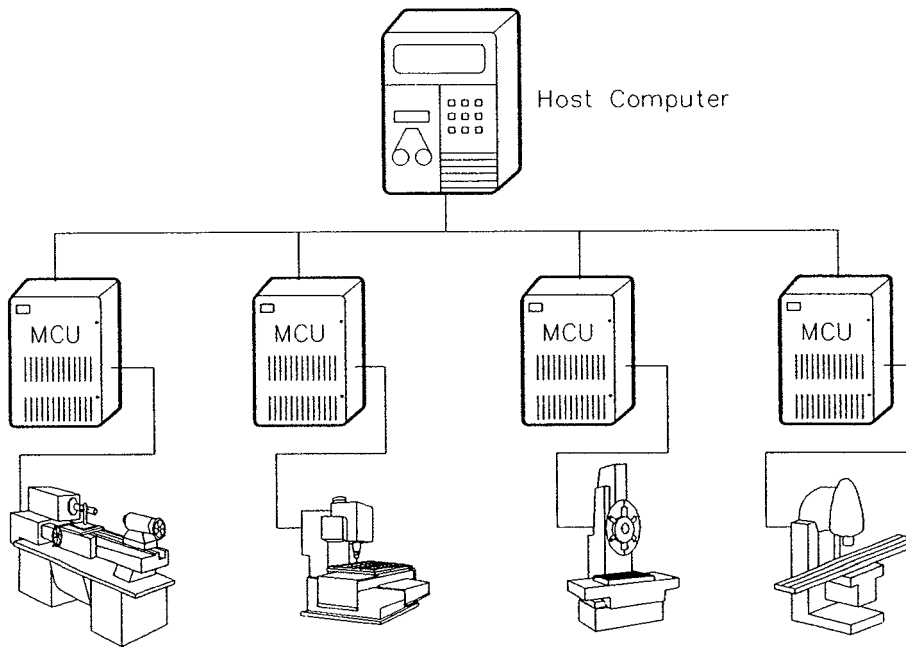
The operation of the APT system required a large computer with a minimum of 256-kilobyte storage capability. This requirement seemed costly for parts that did not require complex machining. For this reason, several language systems derived from APT were developed for smaller computers, including ADAPT (IBM), IFAPT (France), MINIAPT (Germany), and FAPT (Japan). Other programming languages that were not APT-related, such as COMPACT II, AUTOSPOT, AUTOPROMPT, CAMP I, and SPLIT, were also developed concurrently with APT.

### **1.2.4 Direct Numerical Control**

In the mid-1960s Cincinnati Milacron and General Electric independently proved the feasibility of the concept of direct numerical control (DNC), in which NC machines are operated directly from a remote computer. What may have been the first successful DNC demonstration was given in 1966. A computer located about 3000 feet from a hard-wired NC milling machine successfully transmitted instructions to the MCU via a telephone line. Many DNC systems were soon developed, and a half-dozen exhibitors displayed DNC systems at the 1970 National Machine Tool Builders Association (NMTBA) show. Figure 1.1 illustrates the concept of direct numerical control, in which a shared computer is used to program, service, control, and execute machining processes for a group of NC machine tools. The DNC systems were primarily used to download part programs to NC machines. Their uses and functions were very limited.

### **1.2.5 Computer Numerical Control**

In contrast to the remote-controlled concept of DNC, the idea of computer numerical control is to position a computer right at the machine. The emergence of integrated-circuit minicomputers and lower-cost CRTs (cathode ray tubes) brought the CNC concept into reality. A dedicated computer is built into the MCU to control one machine tool. Nearly all



**FIGURE 1.1**  
Direct numerical control.

numerically controlled machine tools are of the CNC type. Today's computer technology provides CNC systems with a higher level of stored intelligence right at the machine, with convenient editing capability and memory capacity for part programs storage.

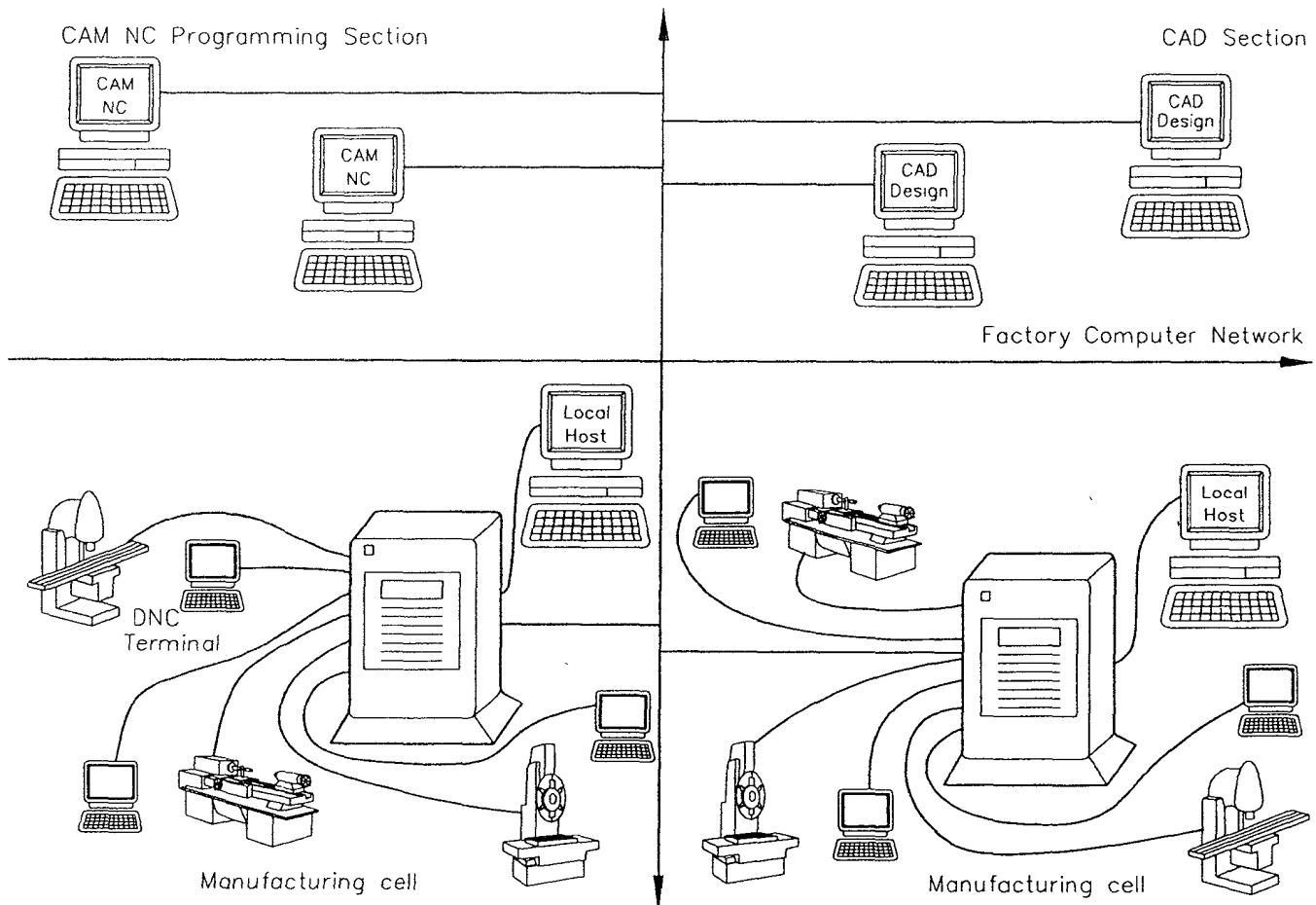
### 1.2.6 Distributive Numerical Control

Two desirable features of CNC—the ability to run a part program from memory and the capability of storing multiple part programs in the memory—make it possible to operate a CNC machine independent of the host computer. This frees the host computer's time and capacity for performing other system tasks. With these CNC capabilities and other technological advancements in computers, programmable controllers, and data communication network systems, another form of numerical control—distributive numerical control—began to evolve in the early 1980s. In distributive numerical control systems, a network of computers is used to coordinate the operation of a number of CNC machines (Figure 1.2). In addition to downloading part programs to NC machines, these systems can handle several other important functions such as line balancing and scheduling, monitoring of machine and control status data, and generation of management information.

### 1.2.7 CAD/CAM NC Programming

Modern computer technology has had a significant influence on engineering design and manufacturing in terms of reducing design and manufacturing cost, inventory, and lead time, as well as increasing productivity and product quality. The idea behind CAD/CAM is the use of a single technical database by both design and manufacturing personnel during the product development and production stages.



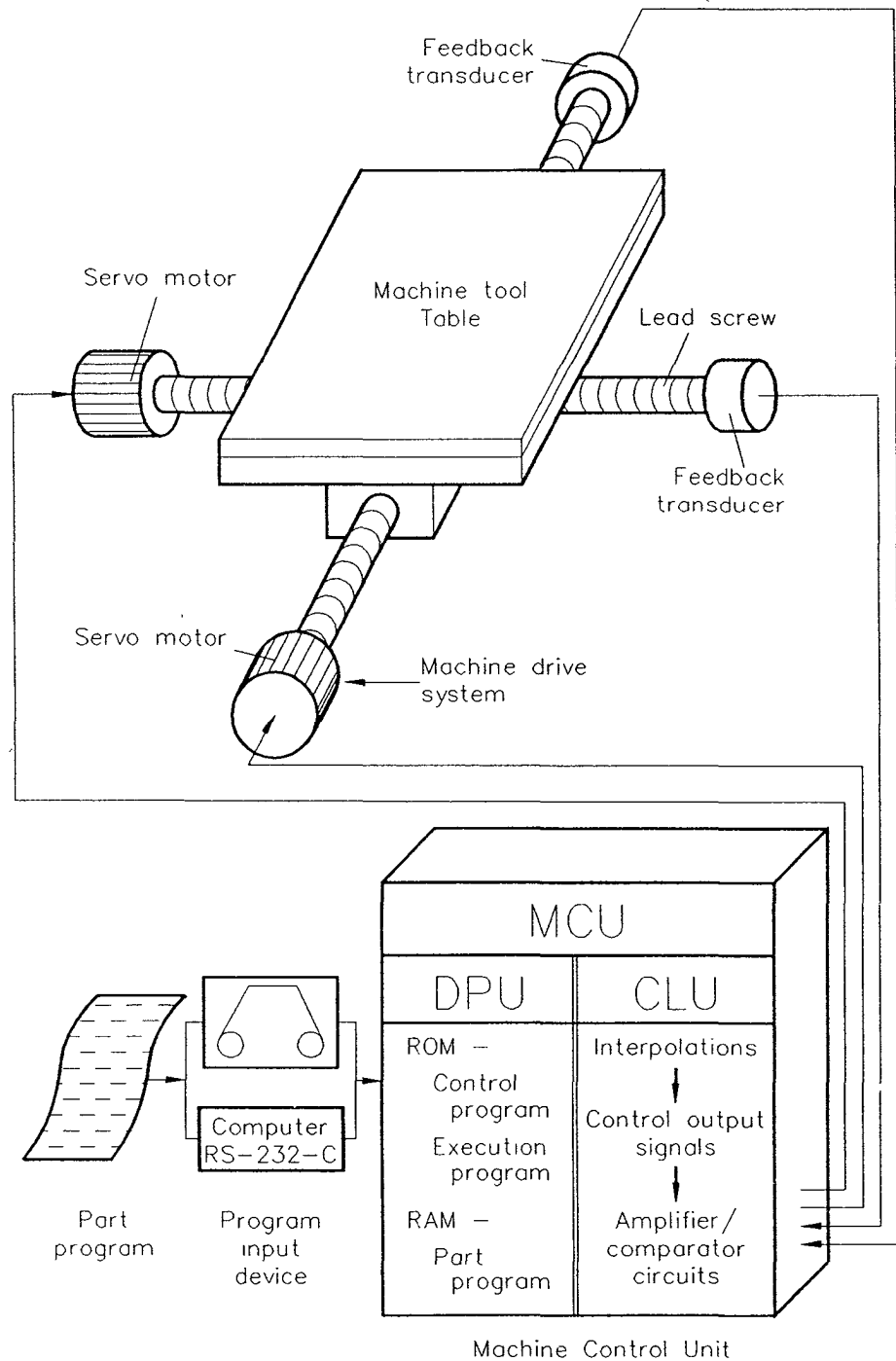


**FIGURE 1.2**  
Distributive numerical control.

In numerical control applications, CAD systems are used to define part geometries, and CAM systems use these geometrical data to generate the cutter paths that are the cutter centerline data. Other features of CAM may include process planning, group technology, production planning, production control, and materials requirement planning. The implementation of CAD/CAM in NC programming has gained in popularity in recent years. Around 60 CAD/CAM NC software systems have been developed. The majority of these systems run on microcomputers, and some operate on minicomputers or even on mainframes.

### 1.3 BASIC ELEMENTS OF A CNC SYSTEM

An in-depth discussion of CNC hardware and software is given in Chapter 3, but it would be very beneficial at this point to briefly explain the major elements of an NC system and their basic functions. A numerically controlled system consists of the following six major components (Figure 1.3):



**FIGURE 1.3**  
Basic elements of CNC systems.

- Part program
- Program input device
- Machine control unit
- Drive system
- Machine tool
- Feedback system

A part program consists of the coded instructions required to control the movement of a machine tool for producing a part. When loaded to a CNC system, the MCU converts these instructions into electric signals that activate machine tool functions and generate cutting tool movements.

The function of the MCU is to read and interpret the coded instructions for machining a particular workpiece and then generating electric output signals. These control signals are fed into the servo amplifiers for regulating the servo mechanism, which could be electric motors or hydraulic motors. Actuation of the servo mechanism results in the precise positioning of the machine table or spindle via the drive system, usually the ball lead screws. As the lead screws rotate, the feedback devices, which may be attached to the lead screws, detect the direction, position, and speed of the motion and send these signals back to the MCU. The MCU compares these signals with the reference signals given by the coded instructions and constantly feeds the error signals (the difference between reference signals and feedback signals) to the servo amplifiers until the prescribed position has been reached.

The MCU can be further divided into two sections: the data-processing unit (DPU) and the control loop unit (CLU) (Koren, 1983; Chang et al., 1991). A DPU typically performs the following functions:

- Reads coded instructions (part program) from input devices such as paper tape reader, magnetic tape reader, or other media via RS-232-C port
- Processes (decodes) coded instructions
- Transmits data pertaining to the position of each axis, speed, and other auxiliary functions to the CLU

A CLU implements the following tasks:

- Executes linear or circular interpolations based on the position and speed signals from the DPU and generates control output signals
- Feeds the control output signals to amplifier circuits for driving servomechanisms
- Receives position and speed feedback signals for each of the axes
- Implements auxiliary control functions such as coolant on/off, spindle on/off, and tool change

An NC machine tool is a machine designed to be controlled by numerical data. NC machine tools appear in many forms depending on the type of application. The drive system determines the accuracy and power capacity of the machine tool. It usually consists of servo amplifiers, servo mechanisms, ball lead screws, and a slide table.

---

## 1.4 COMPUTER NUMERICAL CONTROL

Computer numerical control technology evolved from the direct numerical control applications of the late 1960s and early 1970s. Advances in microelectronics and microprocessors have allowed the computer to be built into the machine control unit of NC machine tools. We can define CNC as a self-contained NC system for a single machine tool that uses a dedicated computer controlled by stored instructions in the memory to implement some or all of the basic NC functions. Almost all NC machine control units built today are of CNC type because of its flexibility and relatively low cost.

Conventional NC controllers are commonly referred to as hard-wired controllers, in which all functions such as interpolations, tape format recognition, coded instruction recognition, and absolute or incremental positioning are performed by electronic circuitry. With the power of the microprocessor, CNC controllers attempt to implement as many of the conventional MCU functions as possible with computer software. This simplifies the CNC hardware circuitry and consequently lowers costs and improves reliability. CNC controllers are often referred to as soft-wired controllers.

CNC systems offer many features not found in their conventional NC (hard-wired) counterparts. They include:

1. **CRT display and graphic representation of the part program.** A cathode ray tube (CRT) provides a multifunction screen that displays information on positioning, feed rate, spindle speed, and other operational variables, as well as parametric data when a part program is being executed. Graphic capability has also been built into modern controllers to visualize workpieces and cutters and simulate cutter paths.
2. **Data entry.** Most CNC controllers furnish an on-board full alphanumeric keyboard to facilitate manual data input (MDI) to controllers. In addition, a tape reader and RS-232-C or 20 mA current loop connector are included to allow the part programs written at external devices to be downloaded to the CNC controllers. These programs may also be uploaded to a computer and stored in a memory disk, or put on punched tape for future use.
3. **Controller memory.** Two types of memory are available in CNC controllers: read-only memory (ROM) and random-access memory (RAM). The execution program resides in ROM, which cannot be erased without special equipment, and the part programs are stored in RAM. The information in RAM would be lost with the removal of power; however, with the use of battery backup power, the programs in RAM can be retained even when the controller is turned off. Many modern CNC controllers provide the memory capacity for 200 feet of tape programming, and some even extend to 2000 feet.
4. **Program editing.** After part programs are downloaded to the CNC memory, changes or updates can be made right at the machine. Such edits are then stored in RAM and override the information that is read in via a tape reader or other external device. This feature eliminates the need for making another tape, as required in conventional NC systems when changes are made.
5. **Program verification.** Some modern CNC systems furnish simulation programs that allow the cutter paths and other auxiliary functions of the stored part programs to be displayed and verified graphically on a display unit before the part program is run on the machine.
6. **Error diagnostics.** The built-in diagnostic software programs monitor all operating conditions and functions of a CNC system. If a malfunction such as a program error, operation error, or servo error is detected, an alarm signal and appropriate message will be displayed on the CRT.
7. **Communication facility.** CNC control units have the capability of communicating with other microprocessor-based devices, such as computers, robotics controllers, and programmable controllers. This feature allows part programs to be downloaded

to the CNC machine from a host computer or a computer network. It enables CNC systems to be linked with computer-like devices in distributive numerical control systems and flexible manufacturing systems.

8. **Management information.** Because the CNC system controls almost all machine tool functions via the computer incorporated in the MCU, it is possible to generate useful information on parameters such as spindle on time, part run time, machine down time, and number of parts produced for further machine utilization analysis.
9. **Coordinate system (absolute vs. incremental).** Most CNC systems allow both absolute and incremental coordinate systems to be used in the same part program.
10. **Unit system (inch vs. metric).** Most CNC systems allow both inch and metric unit systems to be used.
11. **Code format (EIA vs. ASCII).** Most modern CNC systems can handle both EIA and ASCII code formats.
12. **Macro and computing capability.** The power of computer programming relies mainly on the ability to perform arithmetic operations, control operations (branching and repetition), logic functions, and variable definition and manipulation. These desirable features, found in most modern CNC systems, make macro programming possible. A macro is a group of instructions stored in memory as a separate program that can be called by a single command in a main program to generate a series of cutter paths. It is extremely useful for repetitive machining sequences such as facing, rectangular pocketing, and cutting ellipses and other curves that can be described mathematically. Macro programming significantly shortens part programs by eliminating the need to repeat program instructions for similar machining patterns.
13. **Cutter compensation for size and length.** The features of cutter compensation allow the programmer to write part programs based on the dimensions given in the part drawings without considering the size and length of the cutters used. Other tasks that can be achieved with the cutter compensation function include the use of different cutter sizes, facilitation of the roughing cut and finishing cut, and regrinding of the cutter without the need to change the part programs.
14. **Interpolation functions.** In conventional NC systems interpolation functions are implemented in hard-wired electronic circuits, whereas CNC systems use the power of the microprocessor to execute interpolations through software. The two commonly used interpolation functions in most NC and CNC systems are the linear and circular types. Some modern CNC controllers offer parabolic, helical, and cubic interpolators.
15. **Advanced programming features.** CNC systems take advantage of the computing power offered by the microprocessor to implement several advanced programming features, such as scaling (enlargement and reduction of part geometry), mirror imaging (axis inversion), and polar rotation.
16. **Postprocessing capability.** Some modern CNC systems have the embedded postprocessing capability to accept BCL (binary cutter location) data directly. With each machine having its postprocessor stored in the controller memory, the need for off-line postprocessors is eliminated. This type of postprocessing is achieved in real time as the part program is run. It requires considerable space in terms of CNC memory and logic sections.



## 1.5 ADVANTAGES OF CNC

Since the successful introduction of the first numerically controlled milling machine by the Massachusetts Institute of Technology Servomechanism Laboratory in 1952, NC/CNC technology has revolutionized the manufacturing industry. Today's CNC systems are very reliable, highly accurate, and less costly. They are widely used in almost every aspect of manufacturing processes. The advantages of CNC technology are summarized below.

1. **Increased productivity.** The productivity of a CNC machine is about three to four times that of its conventional counterpart because it requires less time for setup and operator adjustments, parts are produced faster, and it requires less downtime and inspection time.
2. **High accuracy and repeatability.** Accuracy is the measure of how close a machine tool positions cutting tools to an indicated location. Repeatability is the ability of a machine tool to repeat its accuracy. CNC machines with a closed-loop control system have the capability of producing workpieces with repeatable accuracy in both form and dimensions. These features facilitate the interchangeability of parts and keep the amount of lost material to a minimum.
3. **Reduced production costs.** Factors contributing to the lower production cost of workpieces made with CNC machines include
  - Saving on tooling, jigs, and fixtures
  - Extended tool life due to optimal cutting conditions
  - Savings from less scrap and rework
  - Saving on labor costs due to the lower skill required and greater operator productivity
  - Savings from the repeated use of part programs
  - Reduction of overall production time
  - Higher machine utilization due to downtime reduction
  - Greater operator efficiency
  - Reduction in manual part handling
  - Reduction of human error
4. **Reduced indirect operating costs.** Indirect operating costs can be greatly reduced by the use of NC through
  - Reduced part inventory
  - Reduced lead time for production
  - Shorter delivery schedules
  - Greater machine tool safety due to less need for operator intervention
  - Better machine utilization because the machine spends more of its time in actual machining
  - Reduced inspection time since CNC machines produce workpieces of homogeneous quality

5. **Facilitation of complex machining operations.** CNC machines may be the only equipment that can provide quick and accurate machining operations for workpieces that involve complex shapes such as three-dimensional surfaces.
6. **Greater flexibility.** CNC machines facilitate quick changes from machining one type of workpiece to another, with minimum setup time and tooling and fixture changes.
7. **Improved production planning and control.** Since CNC machines have complete control over machining and control operations, records on the machining time, downtime, and other information that may be very useful in predicting operating costs and scheduling can be easily gathered for facilitating production planning.
8. **Lower operator skill requirement.** The skills required of the CNC operator mainly involve the loading and unloading of the workpieces and cutting tools, manipulating the control console, and monitoring the operations in progress. These tasks do not require the same level of skills as needed for the operation of conventional machine tools.
9. **Facilitation of flexible automation.** The CNC makes many automated manufacturing systems, such as automated work cells and flexible manufacturing systems, possible and practical.

---

## 1.6 LIMITATIONS OF CNC

Entering its fifth decade of existence, CNC technology has matured with a respectable reliability and has been widely adopted in industry. However, it is not free of drawbacks and limitations. Some of these limitations are described below.

1. **High initial investment.** The biggest single limiting factor in the use of CNC machines is their high initial capital cost and subsequent installation expenditure. CNC machines are expensive. The initial investment may range from \$25,000 or \$50,000 for a small, simple CNC machine up to millions of dollars for a complex machining center. To be cost-effective, CNC machines must be utilized to their full capacity.
2. **High maintenance requirement.** CNC machines are high-technology equipment and can be very complex in their mechanical and electrical systems. To preserve their inherent advantages of accuracy and repeatability, the machines must be kept in top condition and the controllers must be regularly maintained. Maintenance personnel must have expertise in both mechanical and electrical/electronic areas.
3. **Not cost-effective for low production levels.** Using a CNC machine to make one or few parts of a not-too-complex geometry is not cost-effective. A skilled machinist could accomplish the same task in less time and at lower cost. However, as the geometric complexity of parts increases, a CNC machine becomes more economical.

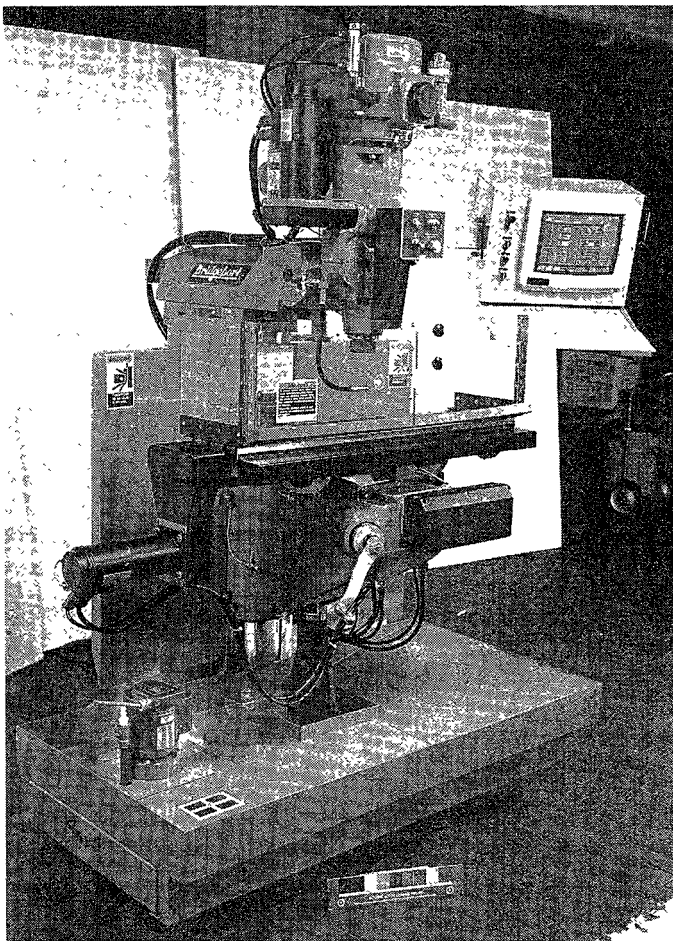
---

## 1.7 APPLICATIONS OF COMPUTER NUMERICAL CONTROL

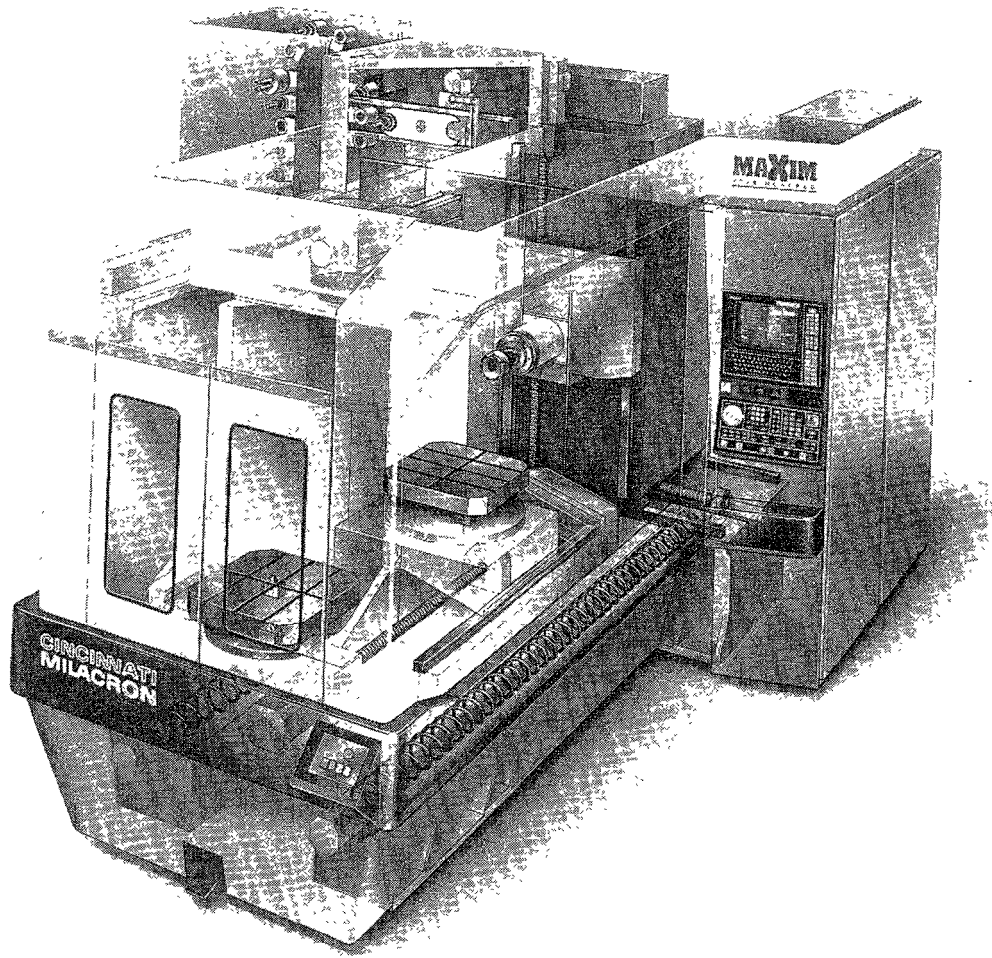
Applications of computer numerical control technology can be classified into five categories:

- Chip removal (metal cutting)
- Grinding
- Unconventional machining
- Fabrication
- Special-purpose applications

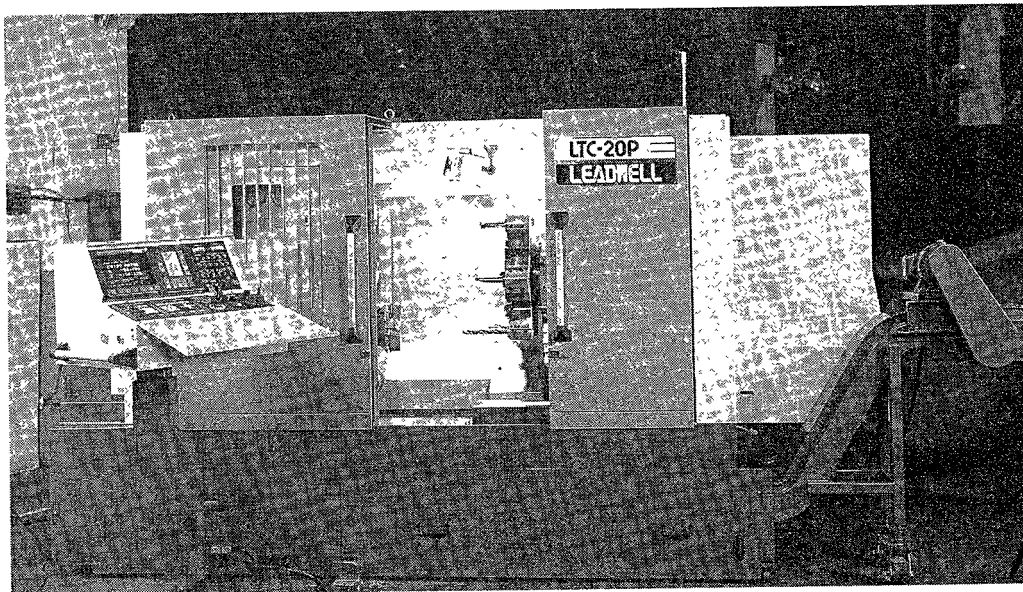
Chip removal applications consist of milling, turning, drilling, boring, and tapping. NC machine tools that fall in this category include machining centers, CNC mill/drill machines, CNC turning centers, and mill-turn machines. About 75 percent of all NC machine tools are of the chip removal type. A recent survey published in the *Modern Machine Shop 1992 Guidebook to CNC Technology* indicates that 25 percent of CNC machine tools used in the United States are machining centers, 27 percent are turning centers, 12 percent are CNC mills, and 10 percent are mill-turn machines. Figures 1.4–1.9 show a variety of CNC chip removal machines.



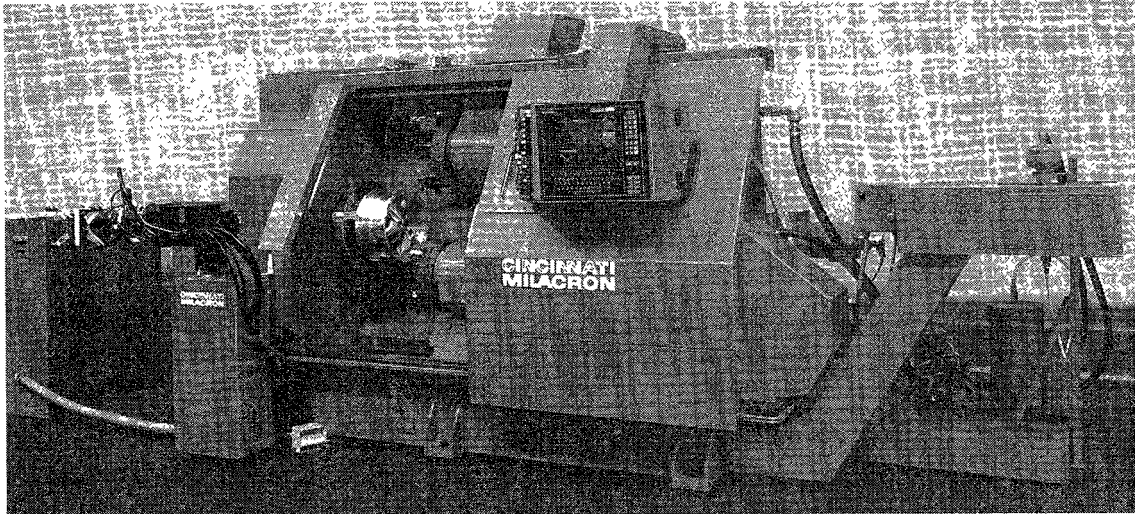
**FIGURE 1.4**  
CNC mill. (Courtesy of Bridgeport Machines, Inc.)



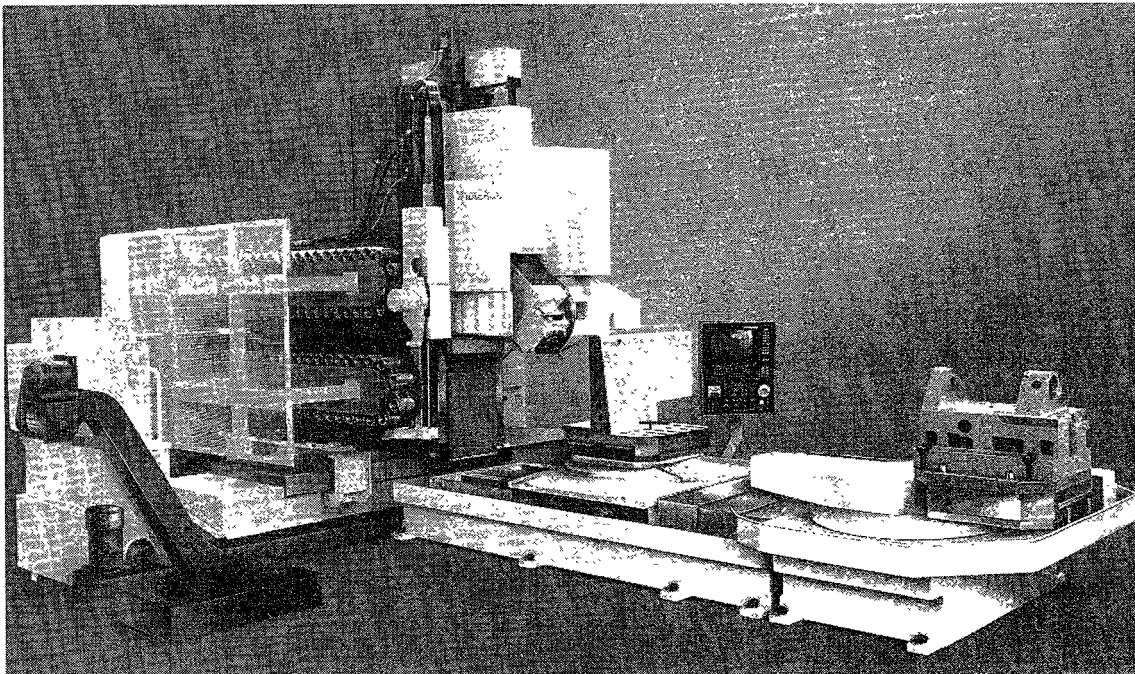
**FIGURE 1.5**  
Three-axis horizontal machining center. (Courtesy of Cincinnati Milacron.)



**FIGURE 1.6**  
Two-axis turning center. (Courtesy of Leadwell Manufacturing, Inc.)

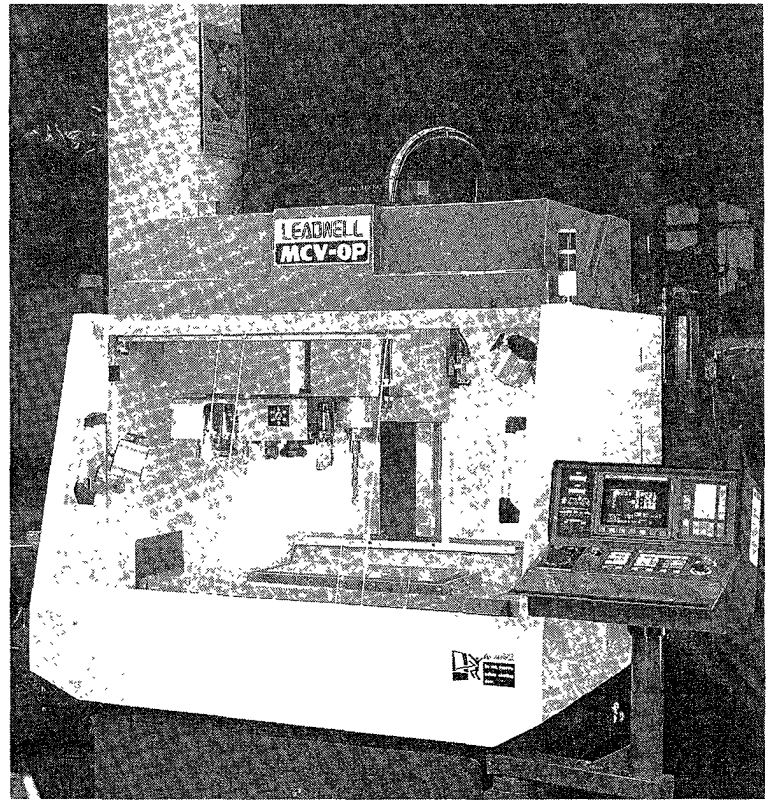


**FIGURE 1.7**  
Four-axis turning center. (Courtesy of Cincinnati Milacron.)

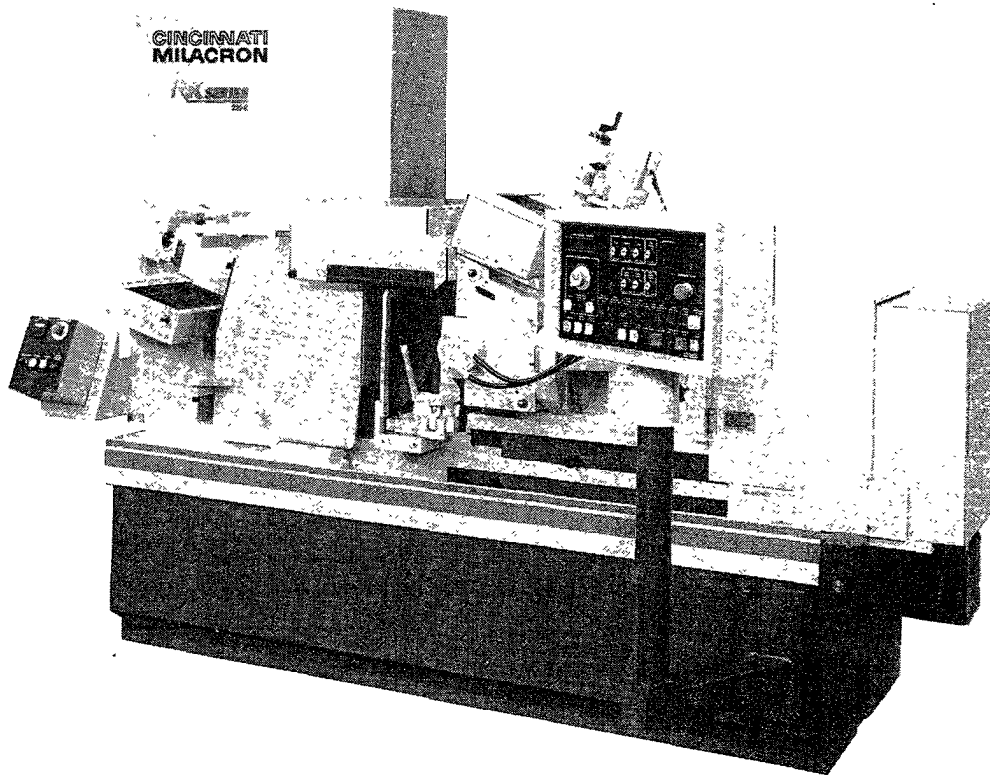


**FIGURE 1.8**  
CNC profiler. (Courtesy of Cincinnati Milacron.)

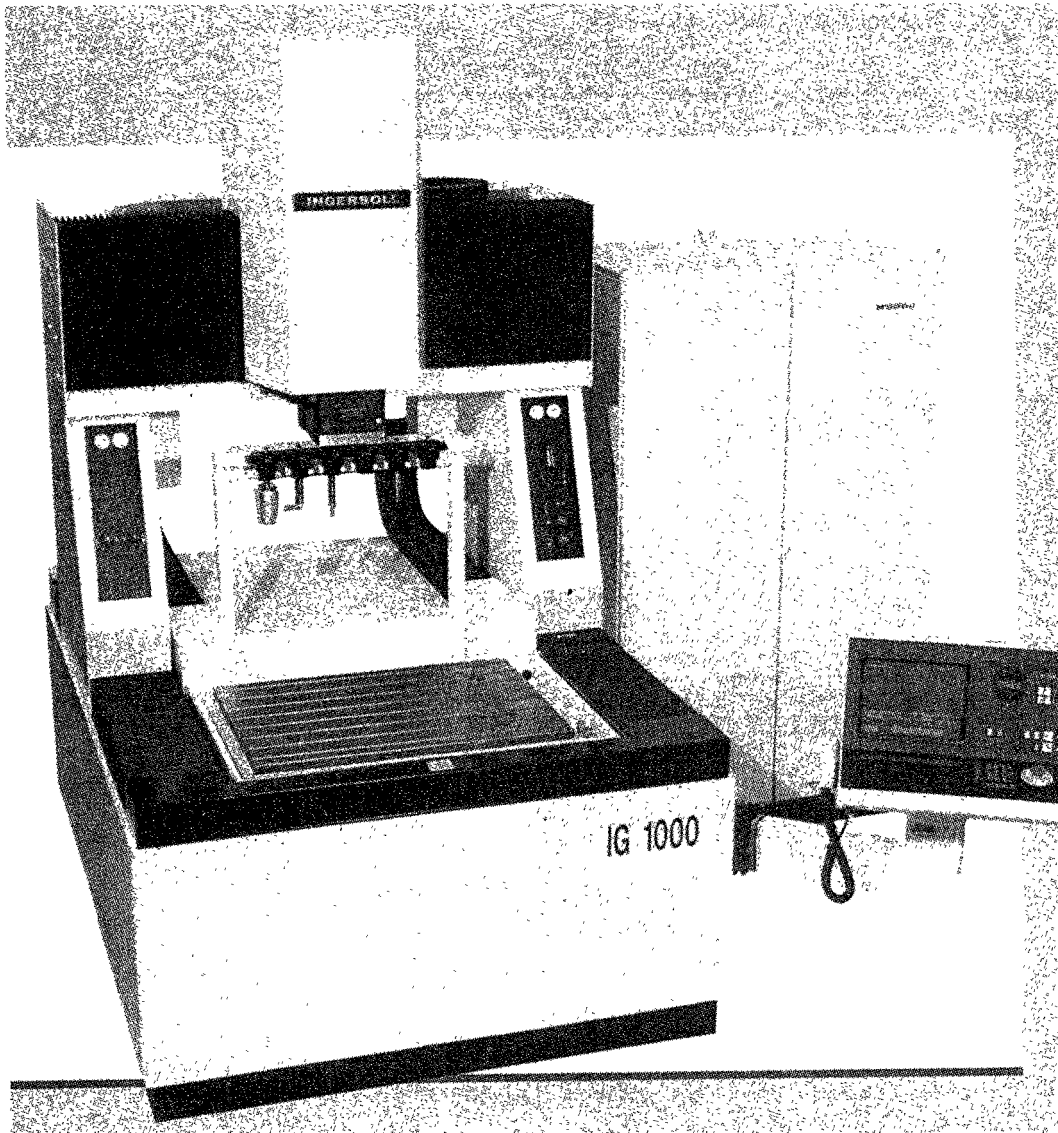




**FIGURE 1.9**  
Three-axis vertical machining center. (Courtesy of Leadwell Manufacturing, Inc.)



**FIGURE 1.10**  
CNC centerless grinding machine. (Courtesy of Cincinnati Milacron.)



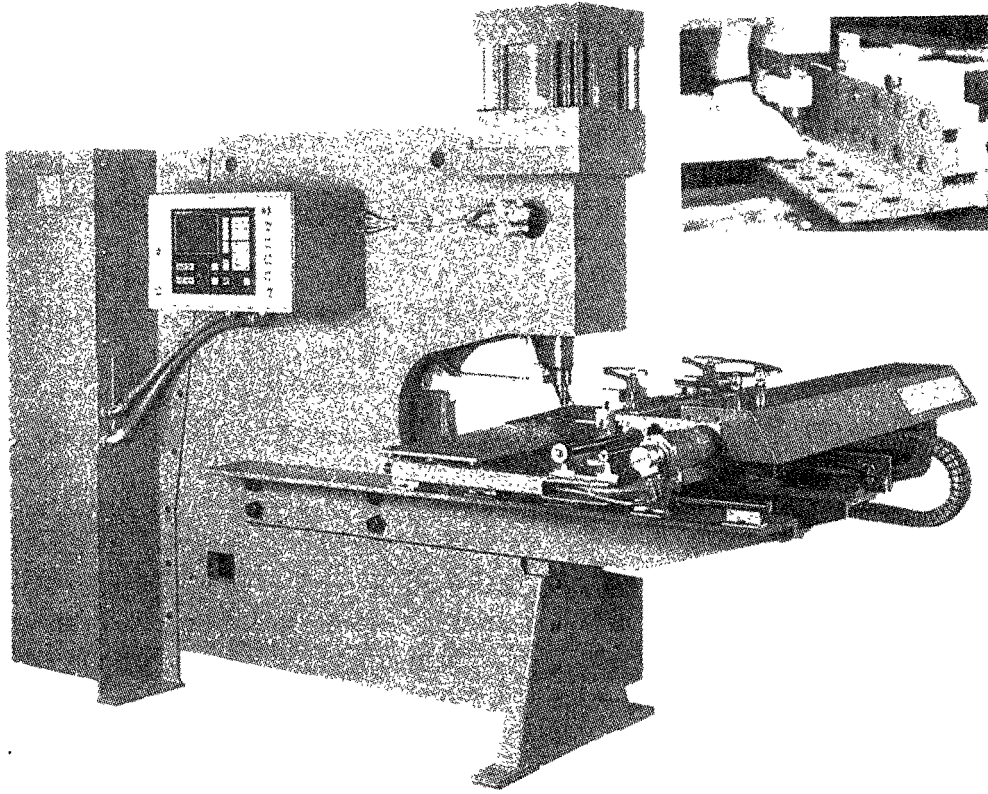
**FIGURE 1.11**  
CNC EDM. (Courtesy of Ingersoll GmbH, Inc.)

Grinding is the second major application of CNC machine tools. About 7 percent of all CNC machines are grinders. Figure 1.10 shows a CNC grinding machine.

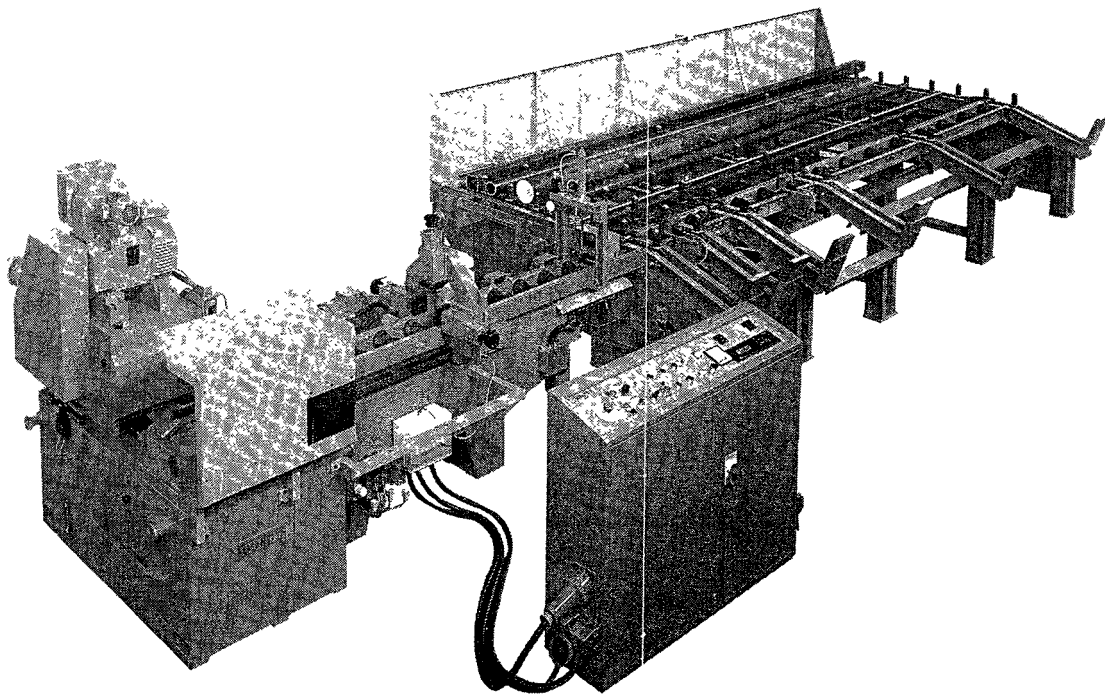
Unconventional machining includes electrical discharge machining (EDM) (Figure 1.11), electron beam machining, and laser machining. Nearly 7 percent of CNC machines are of this category. Among them, EDM is coming on strong and gaining ground.

NC has been applied to such fabrication processes as sheet metal punching, bending, pressing, and shearing. NC controlled fabrication machines account for 5 percent of all NC units. Figure 1.12 shows a CNC punch press. Figure 1.13 shows a CNC sawing center. NC units are also used to control such special-purpose applications as coordinate measurement machines, assembly, and material handling. Figure 1.14 shows a CNC coordinate measuring machine.

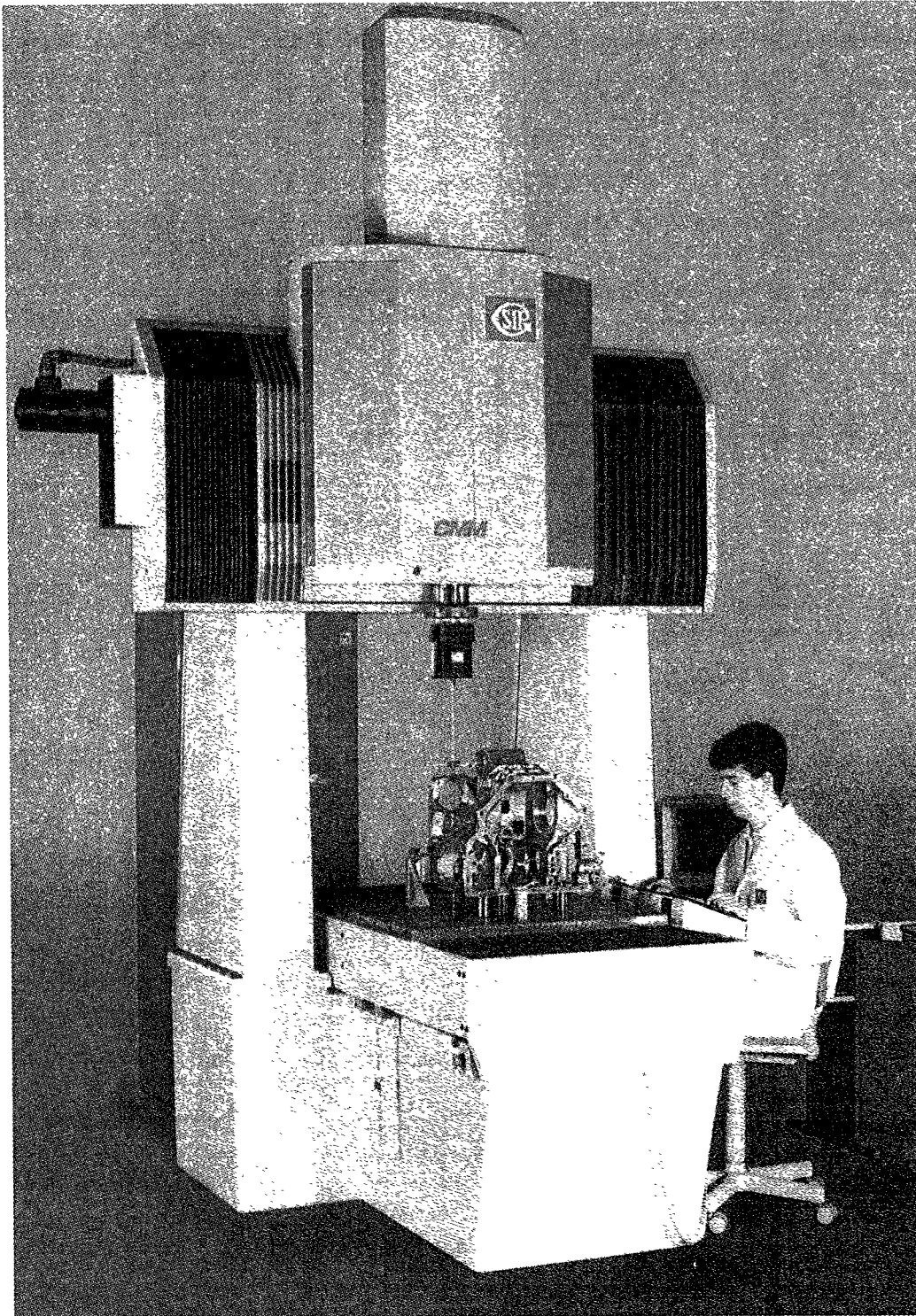




**FIGURE 1.12**  
CNC punch press. (Courtesy of Peddinghaus Corp.)



**FIGURE 1.13**  
CNC sawing center. (Courtesy of Peddinghaus Corp.)



**FIGURE 1.14**  
CNC coordinate measuring machine. (Courtesy of American SIP Corp.)

## 1.8 FUTURE OF CNC TECHNOLOGY

Compared with their predecessors of the last four decades, today's CNCs are more reliable, are more accurate, are faster, have more features, and are capable of executing many more functions. CNC technology is the best answer to the calls for productivity, product quality, and cost savings that are the survival elements in today's competitive manufacturing world. Until recently, most manufacturing people held the view that CNC machines are applicable only for batch-type jobs, not for production jobs. This viewpoint has changed drastically. Almost all machine tools used in new production lines in the automobile industry are CNC-controlled as opposed to the single-purpose type. This trend may be attributed to the CNC features of programmability, flexibility, repeatability, and reliability.

It is estimated that it may take 50 years for a new manufacturing technology to be fully accepted. NC technology has been available in the United States for 40 years, so one can hope that in the next 10 to 15 years the number of CNC machine tools used in the United States will grow from the current level of 6 percent to 30 percent (the ratio in Japan today) of the total machine tool population.

Although CNC is a relatively mature technology, efforts are still being made to improve its power and functionality. A number of developing trends in CNC technology are discussed below.

1. **Processors.** Today's CNCs use faster processors and multiprocessor systems. The fastest CNCs use 64-bit processors and coprocessors with a 64-bit communication bus. The use of faster processors provides higher clock frequency, up to 60 MHz, for executing data transactions. The multiprocessor configuration, consisting of several standard and dedicated microprocessors, allows many functions and features to be added to the system.
2. **RAM memory.** The use of zero-wait-state RAM permits stored part programs and data to be transmitted in a single tick of the microprocessor clock.
3. **Multifunctions.** Multifunction CNC machines have gradually gained popularity. Mill-turn machines, in which milling functions are added to turning machines, is one obvious example. This feature allows a complicated workpiece, requiring both turning and milling operations, to be produced in one machine setup. Dual-spindle CNC lathes are another example. With a dual-spindle lathe, a CNC machine can process two workpieces simultaneously or turn out a workpiece that requires two workholding setups.
4. **Communications capability.** Today's manufacturing plants are moving toward factory integration. Communications is the key element in connecting CNCs to manufacturing systems. New CNC controls should have the capabilities of distributive numerical control, status reporting, tool management, work scheduling, process monitoring, and input/output communications.
5. **Open architecture system.** Most modern CNCs use a proprietary architecture control system. This means that control builders provide both proprietary software and proprietary hardware to machine tool builders. However, the control hardware from one vendor is not, in most cases, compatible with the control software modules from

other vendors. Open CNC architecture will allow CNC manufacturers to share common hardware and software modules, similar to what is being implemented in the PC environment. Some CNC controls are already available with DOS platforms for the user to run off-the-shelf DOS-compatible software packages on the CNC. This may be the first step toward the open CNC architecture.

The major driving force for open CNC architecture is the Next Generation Controller (NGC) initiative, sponsored by the Manufacturing Technology Directorate of the Air Force Wright Research and Development Center (Herrin, 1991). Martin Marietta Corporation is the main contractor for the NGC team, which includes a number of other companies participating in various roles. The goal of NGC is to establish an open architecture standard that will govern the interaction and functioning of NGC-compliant products. This will allow individual manufacturers to develop interchangeable and interoperative NGC modules. In other words, the open architecture system will allow users to shop for various modules from different vendors and put together their own control. It is still uncertain what CNCs will look like 10 years from now. Proprietary architecture may still dominate the market, or open architecture might prevail. Only time will tell.

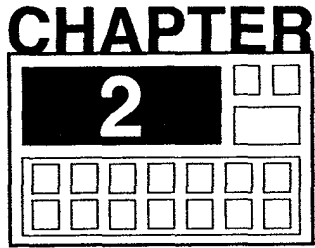
6. **CAD/CAM.** CAD/CAM NC programming has drastically changed the method of preparing part programs. Interactive graphics-based NC programming permits users to easily define part geometry, generate the tool path, obtain immediate feedback, and make changes if necessary. The trend in CAD/CAM NC programming is toward knowledge-based programming. Intelligent systems of this type can be developed to generate the machining strategy, tool path, cutting speeds, feeds, tooling selection and setup, optimal machine loading, and so on.
7. **Small size and affordability.** Many manufacturing plants are beginning to use small CNC machine tools to produce the small workpieces. These CNC machines are less expensive and are affordable to most manufacturers. The market for various types of small CNC machine tools is significantly increasing. This trend is expected to continue.

---

## REVIEW QUESTIONS

1. Define the following terms:
  - a. Numerical control (NC)
  - b. Computer numerical control (CNC)
  - c. Direct numerical control
  - d. Distributive numerical control
2. Briefly describe the development of numerical control.
3. List and describe the six major components of numerical control systems.
4. What are the advantages and disadvantages of direct numerical control?
5. List the main features of computer numerical control.

6. What are the main differences between direct numerical control and distributive numerical control?
7. What additional functions or features does distributive numerical control provide over direct numerical control?
8. What are the advantages of CNC?
9. What are the limitations of CNC?
10. List the five categories of CNC applications.
11. Describe the developing trends of CNC technology and applications.



# NC Systems

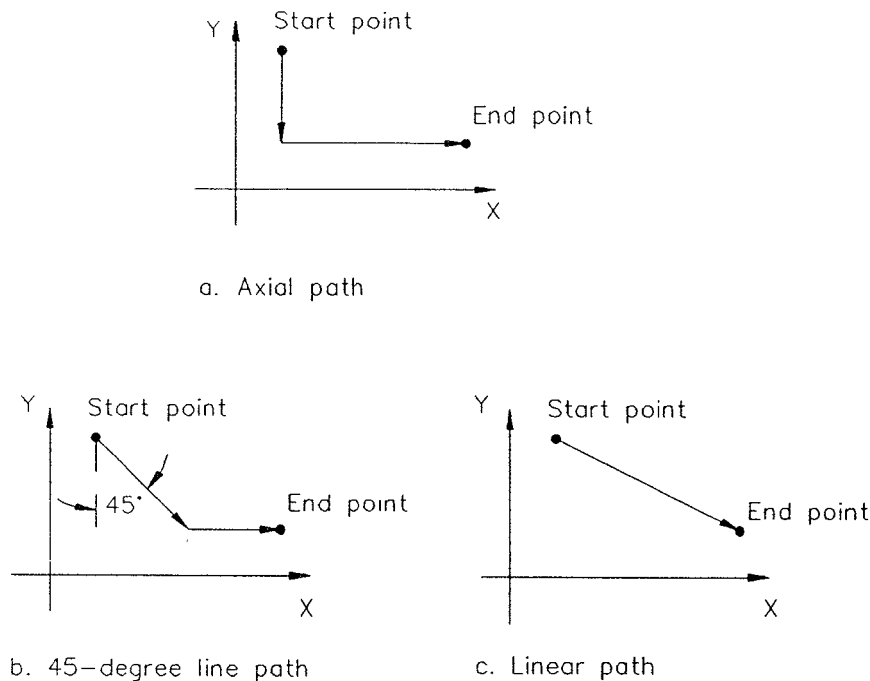
NC systems can be classified according to mode of the control (point-to-point versus continuous-path), structure of the controller (NC or CNC), and type of control loop (open-loop or closed-loop). This chapter focuses mainly on the mode of the NC control, the type of NC control loop, and CNC interpolation. The structure of the NC controller is thoroughly presented in Chapter 3.

## 2.1 MODE OF THE NC CONTROL SYSTEM

### 2.1.1 Point-to-Point

The point-to-point (PTP) control system is often referred to as the positioning system. The main function of the PTP control system is to move a tool from one point to a specified point for performing hole operations such as drilling, boring, reaming, and tapping. In traveling from point to point the tool may follow one of three paths, depending on the make of the controls:

1. **Axial path.** The tool moves in a right-angle pattern parallel to the two major axes, X and Y (Figure 2.1a). In this mode, the controller commands the tool to move first along the Y axis, for example, and the X coordinate does not change until the specified Y coordinate is reached. This is perhaps the slowest method in that the axes are controlled one at a time. The advantage is that a very simple control system is sufficient since coordinated movement of the axes is not required.
2. **45° line path.** The tool moves at a 45° angle until it is in line with the new position, and it then moves directly to it along one axis (Figure 2.1b). Both the X and Y axes move simultaneously until one of the specified coordinates is reached, and the tool then travels in a linear mode to the end point. This is the most common PTP system. Control requirements are kept simple without sacrificing too much speed.
3. **Linear path.** In this mode, the controller has the capability to synchronize the motion in X and Y directions to generate a linear path (Figure 2.1c). This method provides the quickest route between the two points, but sophisticated equipment is required to coordinate the speed of each axis to maintain a straight line. Such expense may not be justified for a simple PTP machine.



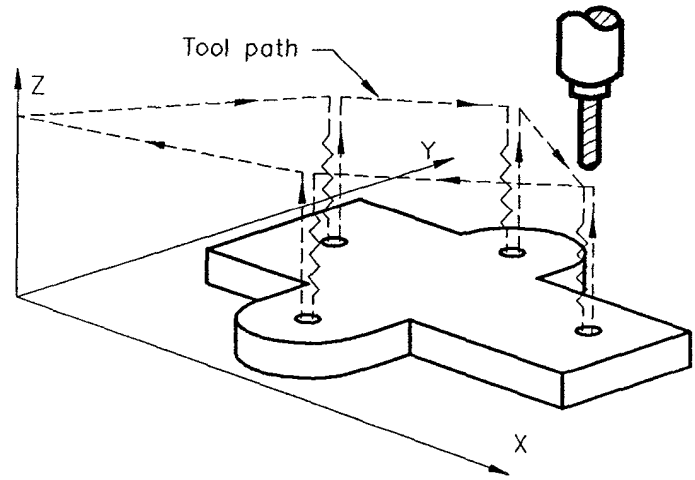
**FIGURE 2.1**  
Point-to-point tool motion patterns.

The most common applications of the PTP control system are drilling, boring, reaming, tapping, and sheet metal punching. These operations normally involve the following steps:

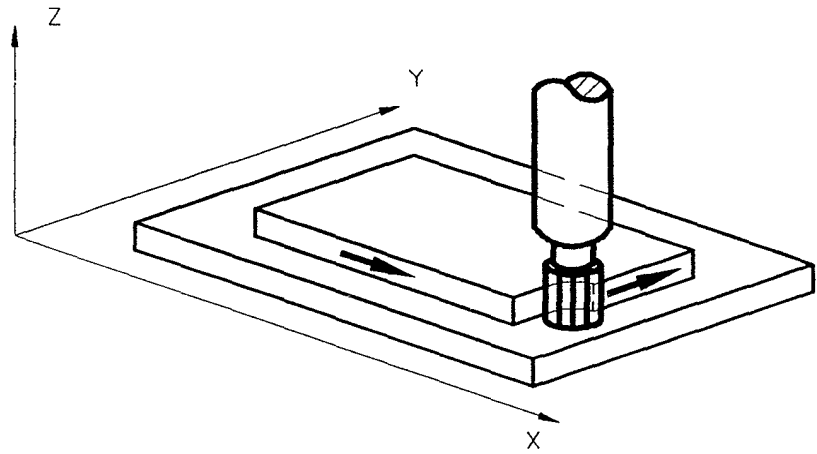
1. **Positioning.** The tool is moved along the axes of motion to the hole center, with no machining action taking place.
2. **Machining.** The machining operation, usually occurring on the Z axis, is performed with controlled spindle speed, feed rate, and depth.
3. **Retracting.** When the desired depth is reached, the tool moves out in a rapid-traverse mode to the initial level.
4. **Cycle repeat.** The tool moves to a new point and repeats the preceding sequence of operations until the work is completed.

Figure 2.2 illustrates a typical drilling application in a PTP system. In addition to positioning for hole operations, a PTP system can also be used to mill straight lines with the feed rate controlled along the X and Y axes, as shown in Figure 2.3. PTP systems that are capable of controlling both axes at the same feed rate can be used to cut straight lines at a 45° angle (Figure 2.4).

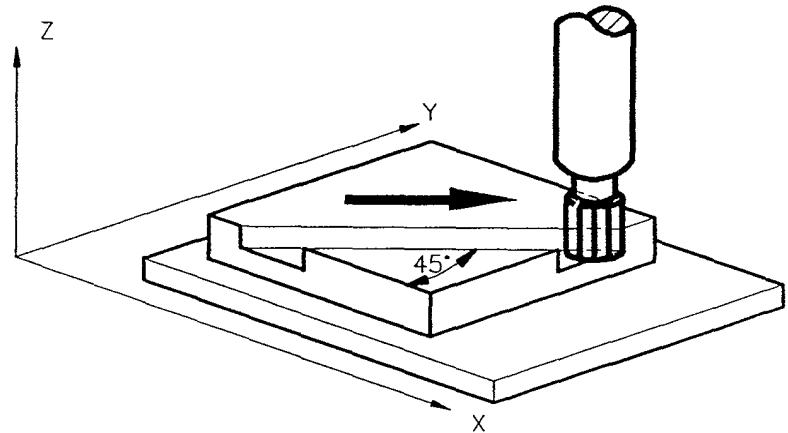
The cutting of arcs and angles other than 45° is not possible with the PTP system, but it can be approximated by a series of straight-line cuts, as shown in Figure 2.5 for an arc segment. The tolerance, or deviation of the actual path from the true path, can be specified in one of the three ways: bidirectional, with both inside and outside tolerance allowed (Figure 2.5a); unidirectional with outside tolerance (Figure 2.5b); and unidirectional



**FIGURE 2.2**  
A typical drilling application in a PTP system.

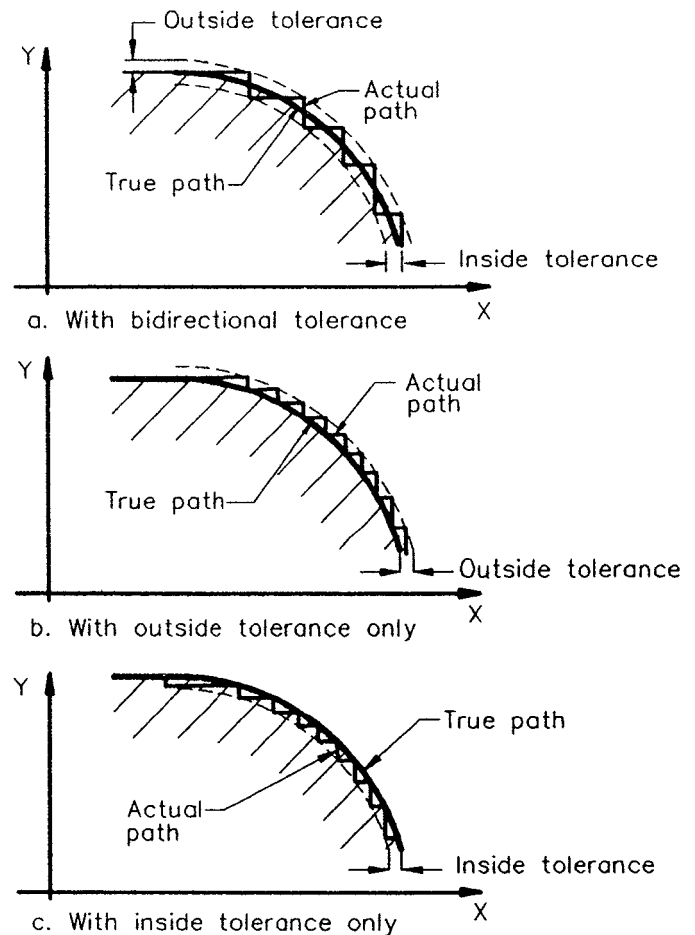


**FIGURE 2.3**  
Axial-path milling in a PTP system.



**FIGURE 2.4**  
45° linear path in a PTP system.





**FIGURE 2.5**  
Approximation of an arc by a series of straight lines.

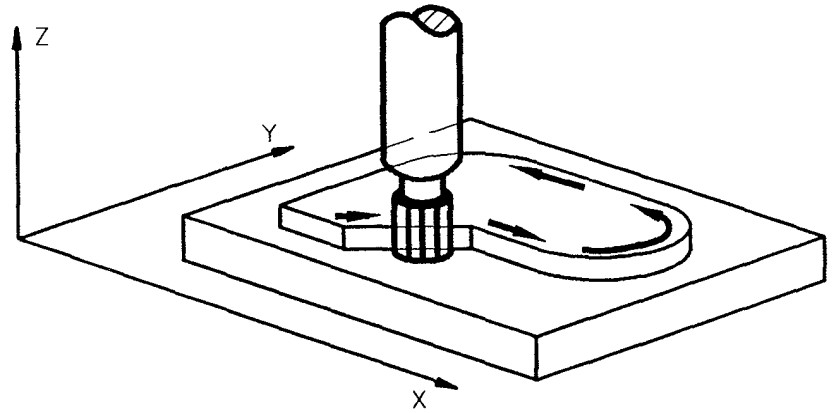
tional with inside tolerance (Figure 2.5c). The larger the tolerance, the smaller the number of programmed segments required.

### 2.1.2 Continuous-Path

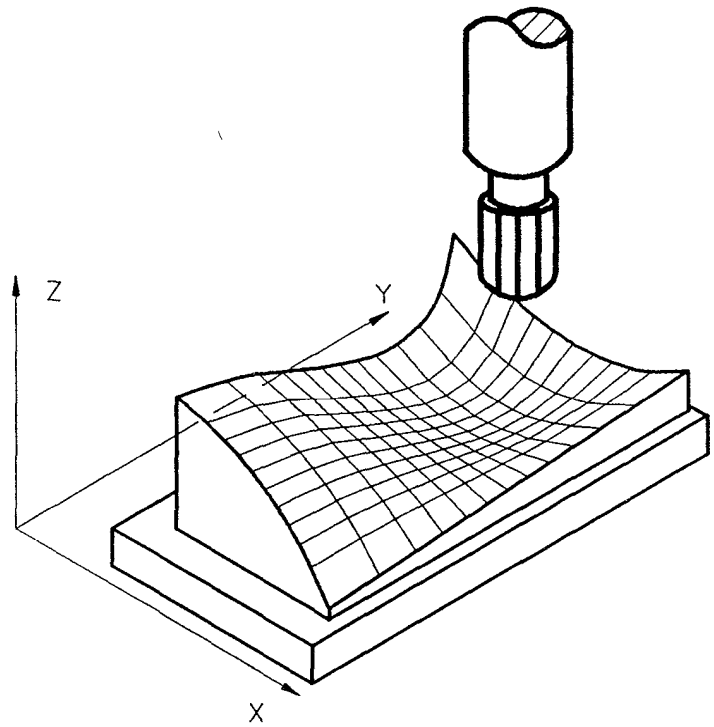
The continuous-path control system is also known as the contouring system. It synchronizes the axes of motion to produce a prescribed path, usually a line or a circular arc. The contouring system has the ability to control its driving motors independently at various speeds in the specified direction while positioning the machine. This involves simultaneous motion control of two or more axes. The contouring system is more complex because each axis of motion requires separate position and velocity loops.

All contouring systems have the ability to cut almost any contour on a two-axis plane (Figure 2.6). Most contouring systems are also capable of performing three-axis contour machining (Figure 2.7), and a few systems are able to perform four-axis contouring (three linear axes and one rotary axis), as shown in Figure 2.8. Very few CNC controls have five- or six-axis contouring capability.

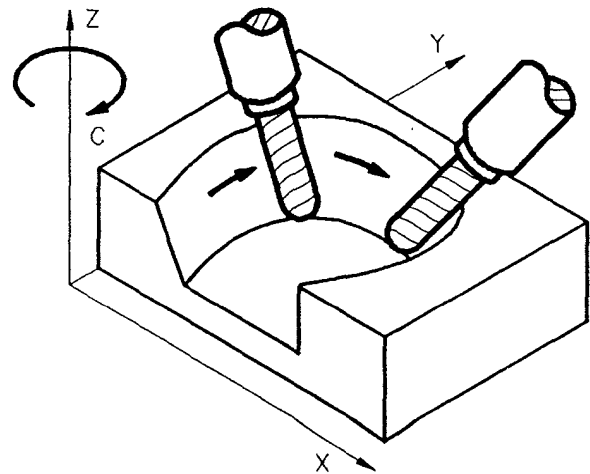
The contouring along a predefined tool path is implemented by means of interpolation, in which the system generates a set of intermediate data points between given coordinate positions.



**FIGURE 2.6**  
Two-axis contour machining.



**FIGURE 2.7**  
Three-axis contour machining.



**FIGURE 2.8**  
Four-axis contour machining.

## 2.2 CNC INTERPOLATION

The interpolator is either an electronic hardware device (for an NC system) or a software program (for a CNC system). An interpolator provides two functions: It computes individual axis velocities to drive the tool along the programmed path at the given feed rate, and it generates intermediate coordinate positions along the programmed path.

There are five types of interpolation: linear, circular, helical, parabolic, and cubic. Most CNC controls furnish both linear and circular interpolation. Some also have helical interpolation, but very few controllers provide parabolic or cubic interpolation.

### 2.2.1 Linear Interpolation

In linear interpolation, the tool moves from the start point to the end point in a straight line. In programming a series of linear moves, only the coordinates of the end point of each line must be specified because the end of one line becomes the beginning of the next. Linear interpolation can be performed for up to five simultaneously moving axes, including three linear axes (X, Y, and Z) and two rotary axes (A and B), to produce the tool path for any profile and solid curve. Two- and three-axis linear interpolations are the most common methods.

Linear interpolation requires three parameters: start point coordinates, end point coordinates, and the speed command for each axis. In two-axis linear interpolation, the interpolator calculates the speed commands, in pulses per second, for the X and Y axes in such a way that it maintains the speed ratio between the X and Y axes equal to the ratio of the required incremental distances ( $dx/dy$ ). For example, to cut a straight line between points S and E in Figure 2.9, the incremental distances along the X and Y axes are 8 and 4 units of length, respectively. Thus, the interpolator simultaneously furnishes 8 and 4 pulses (or a ratio of 2:1) to the control loops of the X and Y axes, respectively.

In three-axis linear interpolation, the interpolator calculates the incremental distances  $dx$ ,  $dy$ , and  $dz$  along the X, Y, and Z axes from the start point to end point. These three incremental distances become the direct input to the position control loops, and their ratio is used for generating speed commands for the three axes. Consider the example in Figure 2.10. To move from points S to E, the incremental distances for position control loops are

$$dx = 4 - 1 = 3$$

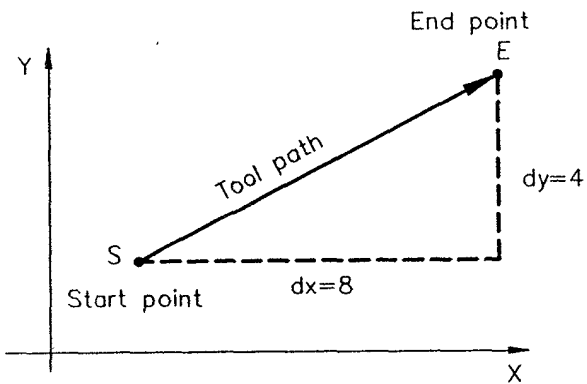
$$dy = 4 - 2 = 2$$

$$dz = 9 - 3 = 6$$

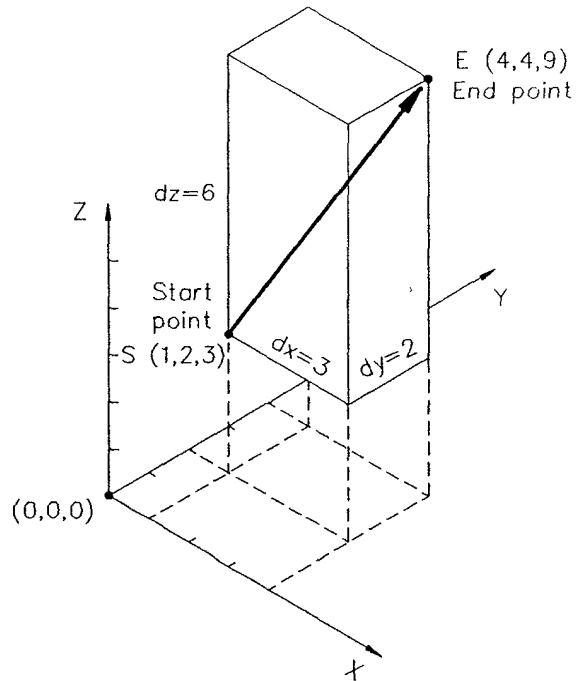
and the speed command ratio for speed control loops is

$$V_x : V_y : V_z = dx : dy : dz = 3 : 2 : 6$$

Theoretically, linear interpolation can be used to cut all types of tool paths, including straight lines, circles, arcs, curves, and helical contours, just to name a few. However, it takes much more data to cut contours other than straight lines. The use of circular, parabolic, helical, or cubic interpolation results in a substantial reduction of data to be processed and thus facilitates programming.

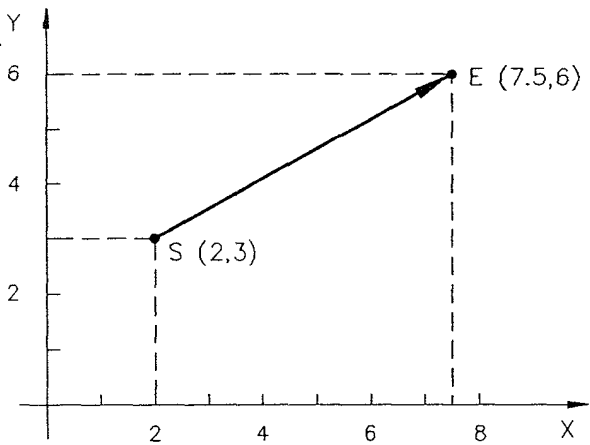


**FIGURE 2.9**  
Two-axis (plane) linear interpolation.

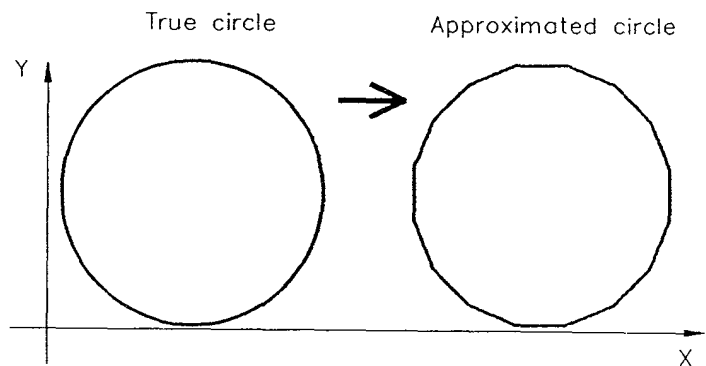


**FIGURE 2.10**  
Three-axis (spatial) linear interpolation.

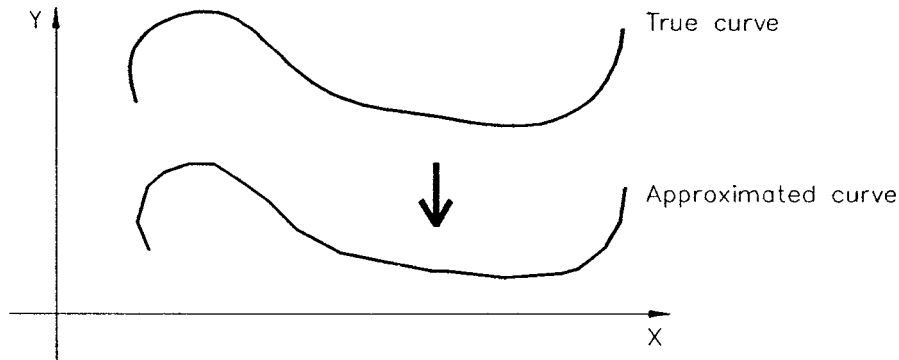
Figures 2.11 through 2.14 show some typical linear interpolation applications. A straight line is interpolated from S(2,3) to E(7.5,6) in Figure 2.11. A circle is approximated by 16 straight-line segments in Figure 2.12. As the number of segments increases, a smoother circle is obtained. Curves that are not able to be defined mathematically can only be approximated by using linear interpolation (Figure 2.13). Figure 2.14 shows an example of cutting a three-dimensional model using linear interpolation. The complex surface is cut by a series of straight-line segments. Most of today's CAD/CAM NC programming systems employ linear interpolation to generate tool paths for complex surfaces.



**FIGURE 2.11**  
Linear interpolation for a straight line.



**FIGURE 2.12**  
Approximation of a circle by a polygon.



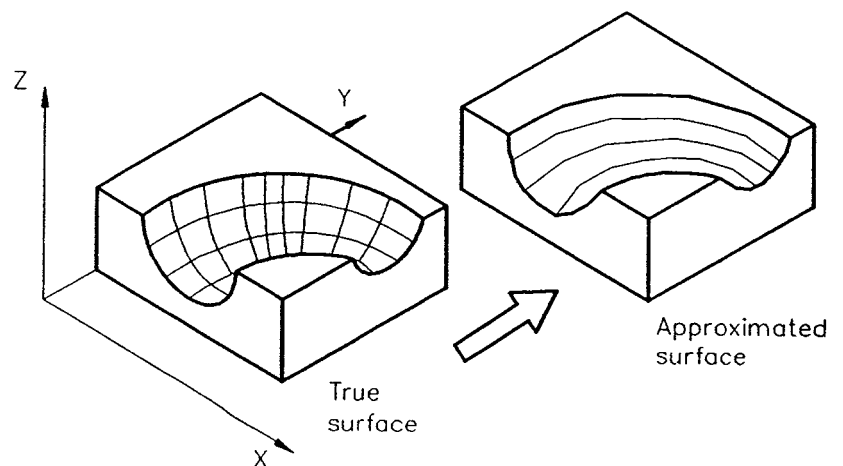
**FIGURE 2.13**  
Approximation of a curve by linear interpolation.

## 2.2.2 Circular Interpolation

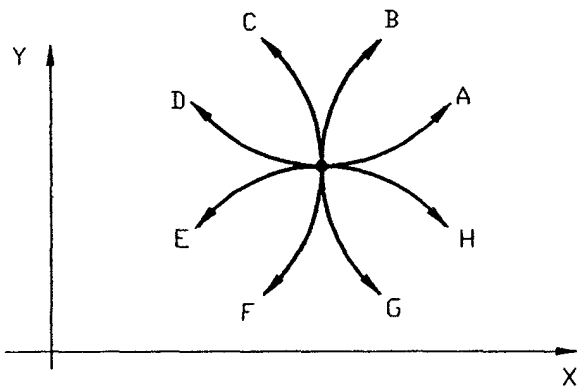
In circular interpolation, the circular span is broken up into a series of small straight-line moves of high resolution, usually a single output pulse of 0.0001 or 0.0002 in. The interpolator computes the axial velocity components  $V_x$  and  $V_y$  and produces a sequence of reference pulses for each control axis of motion. The cumulative number of pulses indicates the axis position, and the pulse frequency represents the axis velocity.

The advantage of circular interpolation is its ability to generate an arc in a single program block. It may require a thousand or more linear interpolation program blocks, each defining a span, to produce the same circle made by only four program blocks with circular interpolation. In some NC controls, circular interpolation is limited to a  $90^\circ$  arc in a single block. Thus, up to five blocks are needed to produce a complete circle. Most modern CNC controls furnish the capability of generating a complete circle with a single program block.

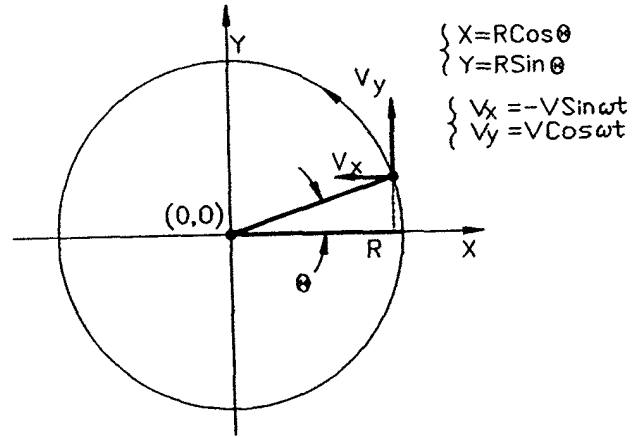
There are eight possible circular arcs that can start from a specified point, as shown in Figure 2.15. The arc C in Figure 2.15 is used to illustrate the circular interpolation principle in Figure 2.16, with its center located at the origin to simplify the derivation process.



**FIGURE 2.14**  
Approximation of a complex surface by linear interpolation.



**FIGURE 2.15**  
Eight possible arcs start from a given point.



**FIGURE 2.16**  
Circular interpolation and its axial feed rate components.

This circular arc must satisfy the circle equation  $X^2 + Y^2 = R^2$ , where  $R$  is the arc radius. The coordinates of any point on the arc can be found by  $X = R \cos \theta$ ,  $Y = R \sin \theta$ , and  $\theta = \omega t$ ,

where  $\theta$  = span angle of the interpolation point

$\omega$  = angular velocity in radians/second or radians/minute

$t$  = time, in seconds or minutes

The axial speed components can be obtained by taking the derivative of these two equations:

$$V_x = dX/dt = -R\omega \sin(\omega t) = -V \sin(\omega t)$$

$$V_y = dY/dt = R\omega \cos(\omega t) = V \cos(\omega t)$$

Note that  $V$  is equal to  $R\omega$ , which is the desired feed rate along the circular arc. These speed components can be used for other arc patterns simply by changing their sign.

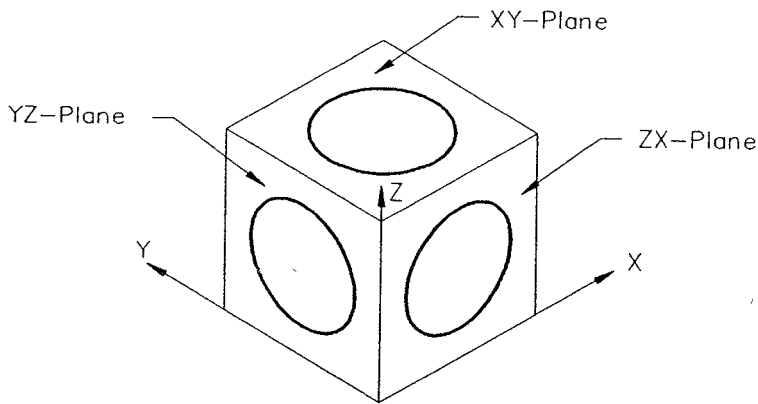
The information required for programming a circular interpolation includes:

1. Coordinates of the start point
2. Coordinates of the end point
3. Radius of the arc or coordinates of the arc center
4. Direction in which the tool is to proceed

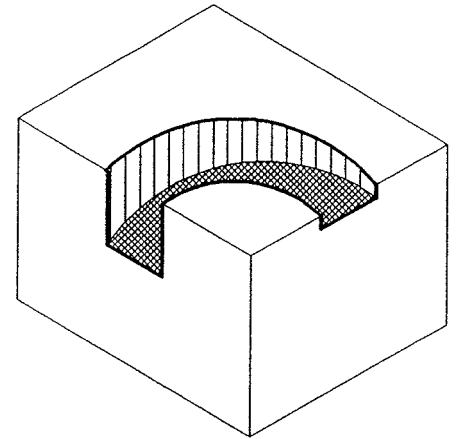
Circular interpolation is limited to the two-axis plane. A circle or arc can be produced in the XY-plane, ZX-plane, or YZ-plane, as shown in Figure 2.17.

### 2.2.3 Helical Interpolation

Helical interpolation is available in most modern CNC controls. It combines the two-axis circular interpolation and a linear interpolation along the third axis. Helical interpolation is a spatial control since all three axes move simultaneously to generate helical cutter paths.



**FIGURE 2.17**  
Circular interpolation in the three principal planes.

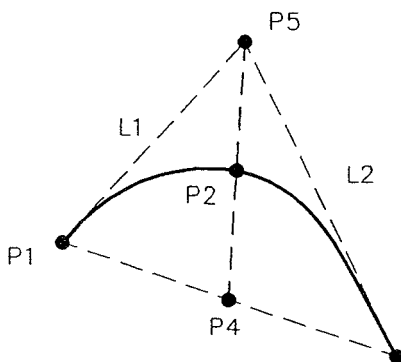


**FIGURE 2.18**  
Helical interpolation.

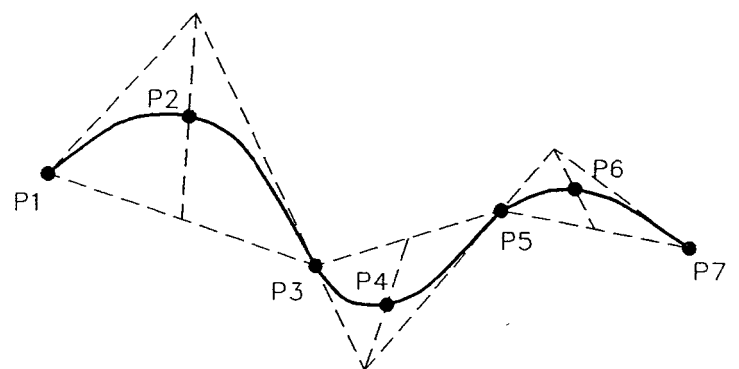
The helical cut can be less than a complete circle or involve more than one circle. It is often used to cut a large internal thread, which could be straight or tapered. Figure 2.18 shows a helical interpolation application.

### 2.2.4 Parabolic Interpolation

Parabolic interpolation uses three noncollinear points to approximate free-form curves. It can be used to cut either planar or spatial curves. Figure 2.19 shows the use of parabolic interpolation to approximate a curve based on three given points, P1, P2, and P3. Points P1 and P3 are the two end points, and P2 is the midpoint between P4 and P5. P4 is the midpoint between P1 and P3. Lines L1 and L2 are constructed to aid the construction of the curve; its two ends are tangent to lines L1 and L2 at P1 and P3, respectively, and it passes through point P2. In programming, P1 is defined from the preceding program block, and P2 and P3 are given in two successive blocks. It is possible to program two or more successive parabolic interpolations to approximate higher-order curves if their tangents at conjunction points are identical (Figure 2.20).



**FIGURE 2.19**  
Parabolic interpolation.



**FIGURE 2.20**  
Approximation of a high-order curve by parabolic interpolation.

The use of parabolic interpolation to closely approximate curve sections has the advantage of reducing the number programmed points by a factor of 50 compared with that required by the linear interpolation method. Parabolic interpolation is primarily used in mold and die making, where free-form designs are preferred over precisely defined shapes. The sculpturing of automobile sheet metal dies, for example, is a good application for parabolic interpolation, where aesthetic appeal is more important than mathematical significance.

### 2.2.5 Cubic Interpolation

Cubic interpolation is of the third order and can be used to generate complex tool paths for machining complicated shapes such as automobile sheet metal dies with a relatively small number of programmed points. It has the ability not only to approximate the curves but also to smoothly blend one curve segment with the next. A cubic interpolation program is very complex and requires considerable computing power and a large memory. Fortunately, modern computer technology provides the required computing power at reasonably low cost to make the implementation of cubic interpolation possible.

---

## 2.3 OPEN-LOOP AND CLOSED-LOOP CONTROL SYSTEMS

NC systems require servo drives to control both position and velocity of the machine tool axes. Each axis must be separately driven and must follow the command signal generated by the NC control. The servo drive control can be implemented in two ways: the open-loop system and the closed-loop system.

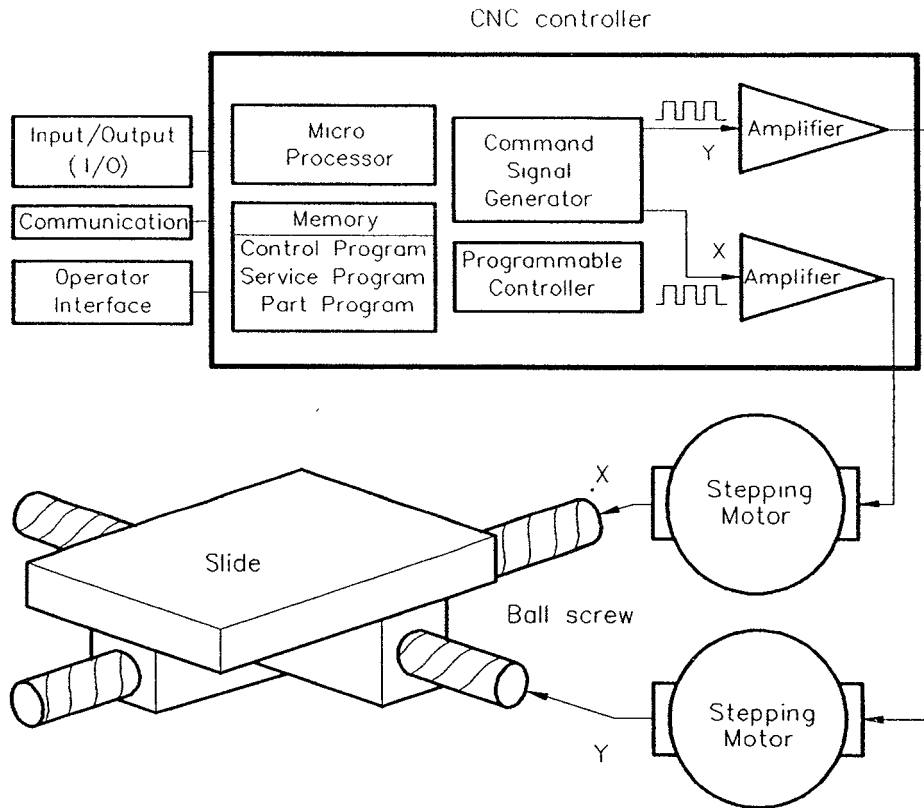
### 2.3.1 Open-Loop System

Figure 2.21 shows the block diagram of a typical open-loop drive system used in NC machines. The instruction program is fed into the controller through an input device, which could be a tape reader, RS-232 communication, or other means. The instructions are then converted to electrical pulses or signals in the controller and sent to the servo amplifiers to energize the servo drives. The cumulative number of electric pulses determines the distance each servo drive will move, and the pulse frequency determines the velocity.

One significant feature of an open-loop system is that there is no feedback signal for checking whether the programmed position and velocity have been reached. The open-loop system is very sensitive to variation in the loading conditions because it has no real-time data about system performance and thus cannot counteract any operating disturbance. If the load on the machine slide changes, the servo drive speed is affected accordingly. The accuracy of the positioning may be influenced by the backlash encountered in the machine drive mechanism. In addition, the performance of system components may be hampered by temperature, humidity, or lubrication, and the actual output may deviate from the desired output. For these reasons, the open-loop system is usually used in point-to-point systems, where the loading torque on the drive motors is nearly constant. Very few continuous-path systems use open-loop control.

Open-loop systems normally use stepping motors as the drive devices to move the machine slide. Due to the advent of precision ball screw and stepping motor control technology, open-loop control systems can be refined to 0.001 in. resolution, which is accurate





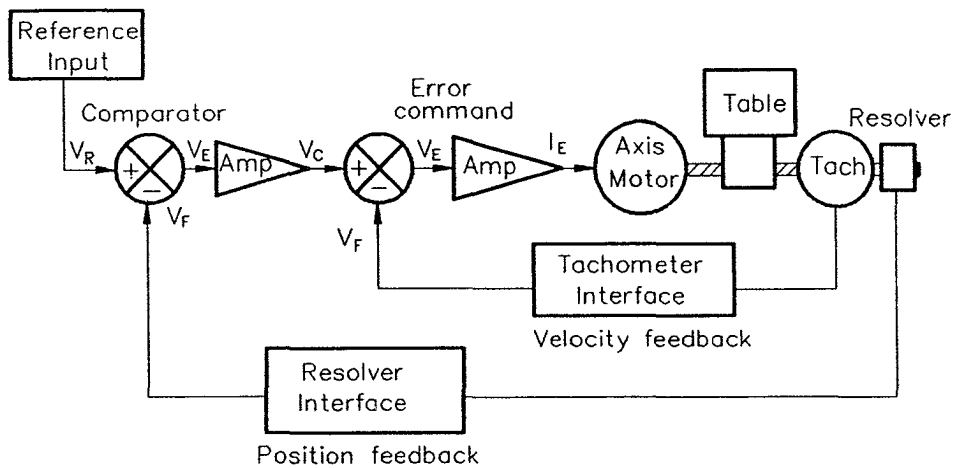
**FIGURE 2.21**  
Block diagram of an open-loop drive system.

enough to be used in many precision positioning and light-load contouring applications. The open-loop system is fairly simple and reliable. It is less expensive than its closed-loop counterpart, and its maintenance is far less complicated. It is not normally used in systems where the required accuracy is greater than 0.001 in.

### 2.3.2 Closed-Loop System

The main difference between open- and closed-loop systems is that the latter include feedback subsystems to monitor the actual output and correct any discrepancy from desired output. Feedback subsystems may be either analog or digital. Analog systems measure the variation of physical variables such as position and velocity in terms of voltage levels. Digital systems monitor output variations by means of electrical pulses.

In CNC applications, both position and velocity must be controlled. The tachometer, an analog type of transducer, is normally used to measure the velocity. Resolvers or encoders are employed to monitor the position and sometimes the velocity. Resolvers are analog, whereas encoders are digital. Figure 2.22 shows a resolver-based (analog) closed-loop control system. Upon receiving the program instructions, the interpolator produces reference signals in the form of voltage levels to each axis control loop to coordinate the contouring motion. A comparator compares the reference and feedback signals for each axis and then sends an error signal to the drive amplifier. The drive amplifier is used



**FIGURE 2.22**  
Resolver-based (analog-type) closed-loop drive system.

because the command error signal itself is not large enough to drive a servo motor. The amount of command signal magnification is called gain or loop gain. The four types of gain commonly used in closed-loop systems include (Parker Hannifin, 1990)

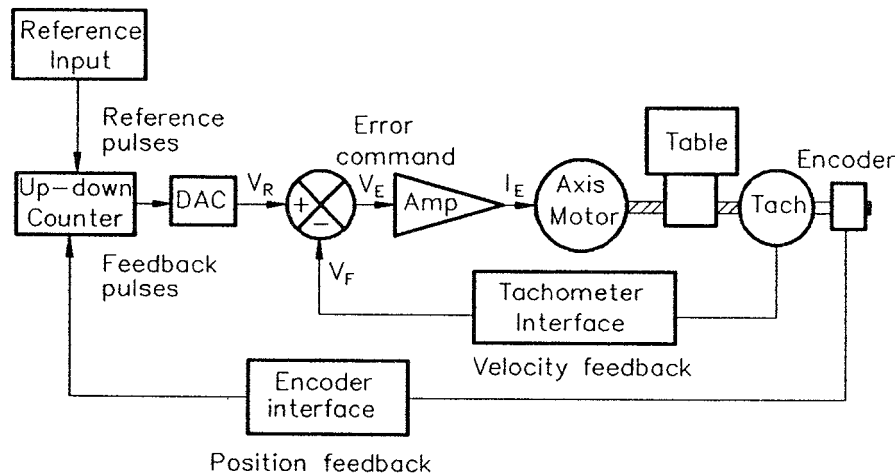
1. **Proportional gain.** The system error is magnified by this value. Proportional gain dictates the overall response of the system and magnitude of positional error.
2. **Integral gain.** The system positional error maintained at steady state is magnified by this value. This gain controls the final accuracy and stiffness of the servo drive.
3. **Derivative gain.** The system positional error due to the rate of change in position error is multiplied by this value. This gain dictates the damping or ringing of the transient response of the motor shaft due to high acceleration rates.
4. **Velocity gain.** The system velocity error is multiplied by this value. This gain allows for compensation over a wide range of inertia load.

The velocity feedback loop consists of a comparator, an amplifier, a tachometer attached to either the lead screw or the servomotor, and a tachometer interface including a signal filter and amplifier. This can be regarded as a subloop within the position feedback loop.

Figure 2.23 illustrates a typical encoder-based (digital-type) closed-loop control system. It is very similar to its resolver-based counterpart, with the following exceptions:

1. An up-down counter is used in place of the position comparator.
2. A digital-to-analog converter (DAC) is added to convert the digital signals to their equivalent analog form.

In operation, the up-down counter is fed by two sequences of pulses: the reference pulses from the interpolator and the feedback pulses produced by the encoder. The counter generates a digital number corresponding to the instantaneous position error. This digital word is then converted by the DAC to a voltage and applied to the velocity comparator. The velocity feedback loop in this system is identical to that in the resolver-based system.



**FIGURE 2.23**  
Encoder-based (digital-type) closed-loop drive system.

During operation, whenever the servomotor is slowed down due to a higher torque demand, the feedback frequency from the encoder is reduced accordingly. This results in an increased pulse difference in the up-down counter and consequently causes the voltage to increase across the servomotor to meet the excessive torque demand. Conversely, as the torque demand reduces, the pulse difference in the up-down counter diminishes to reduce the voltage applied to the servomotor.

The servomotor used in closed-loop systems can be either electrical or hydraulic. DC servomotors are normally used in CNC machine tools as part of the servomechanism.

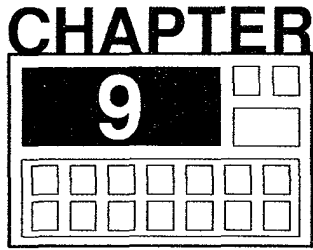
Closed-loop systems are very powerful and accurate because they are capable of monitoring operating conditions through feedback subsystems and automatically compensating performance variations due to load disturbances in a real-time fashion. Most modern closed-loop CNC systems are able to provide very fine resolution of 0.0001 in. This means that one pulse of command signal will cause a 0.0001 in. movement of the machine slide, which is accurate enough for most machining requirements.

Naturally, closed-loop systems require more control devices and circuitry in order for them to implement both position and velocity control. Inevitably, they are more complex in structure and more expensive than their open-loop counterparts.

## REVIEW QUESTIONS

1. What is a point-to-point control system?
2. Identify and describe the three kinds of traveling path for point-to-point systems.
3. What is a continuous-path control system?
4. Explain the two functions of NC interpolators.
5. What is the difference between the interpolators used in NC versus CNC systems?
6. What are the five types of CNC interpolation?

7. Describe how the incremental distance and the speed command ratio are determined in linear interpolation.
8. Show how linear interpolation can be used to approximate two-dimensional curves and three-dimensional surfaces, respectively.
9. What are the four pieces of information required for programming a circular interpolation?
10. What is helical interpolation, and when is it used?
11. Describe, with the aid of a simple sketch, how to use parabolic interpolation to approximate a free-form curve.
12. Illustrate and describe an open-loop type of NC system.
13. Describe the limitations of open-loop NC systems.
14. What are the main applications of open-loop NC systems?
15. What is the main difference between open-loop and closed-loop NC systems?
16. What are the two types of closed-loop CNC systems?
17. List and describe the four types of gain used in closed-loop NC systems.
18. Illustrate and explain the velocity feedback loop of NC systems.



# Milling Programming Fundamentals

Milling operations in numerical control involve the use of numerically controlled milling machines or machining centers to perform milling, drilling, boring, or tapping processes. These machining processes constitute about 75 percent of NC operations. To maximize the power of modern CNC milling machines, a programmer has to master the following five categories of programming command codes and techniques:

- 1. Basic programming commands.** The following command codes, listed with their functions, are used to write an NC program for machining a simple workpiece. Each program also has a safety feature, which normally includes command codes such as G90 (or G91), G80, G40, and G17 that are often given in the first block of the part program. Its main purpose is to clear the control from the previous program and set the control to appropriate modes for the new program.
  - Motion commands (G00, G01, G02, G03)
  - Plane selection (G17, G18, G19)
  - Positioning system selection (G90, G91)
  - Unit selection (G70 or G20, G71 or G21)
  - Work coordinate setting (G92)
  - Reference point return (G28, G29, G30)
  - Tool selection and change (Txx M06)
  - Feed selection and input (Fxxx.xx, G94, G95)
  - Spindle speed selection and control (Sxxxx, M03, M04, M05)
  - Miscellaneous functions (M00, M01, M02, M07, M08, M09, M30)
- 2. Compensation and offset.** The use of compensation and offset functions in defining work coordinate systems, performing tool diameter compensations, and accommodating tool length differences often results in reduced programming effort. The main compensation and offset functions are
  - Work coordinate compensation (G54–G59)
  - Tool diameter (radius) compensation (G40, G41, G42)
  - Tool length offset (G43, G44, G49)

3. **Fixed cycles.** The purpose of a fixed cycle is to execute a series of repetitive machining operations with a single block command. Fixed cycles may be classified into the following three categories:
  - Standard fixed cycles (G80–G89)
  - Special fixed cycles
  - User-defined fixed cycles
4. **Macro and subroutine programming.** Most modern CNC controls furnish the power of computer programming to define variables, perform arithmetic operations, execute logical decisions, and so on. These features allow easy implementation of repetitive machining patterns and complex workpiece shapes that can be defined mathematically.
5. **Advanced programming features.** These commands are dependent on user control. They are used to simplify programming effort and reduce programming time and program size. Typical features include scaling, rotation, and mirror image.

The first four categories are discussed in Chapters 9 through 12, respectively.

---

## 9.1 TOOL MOTION COMMANDS

There are three types of tool motion commands available in most CNC machines: rapid traverse, linear interpolation, and circular interpolation. This section discusses the G-code as used in the command format of these three types of motion commands.

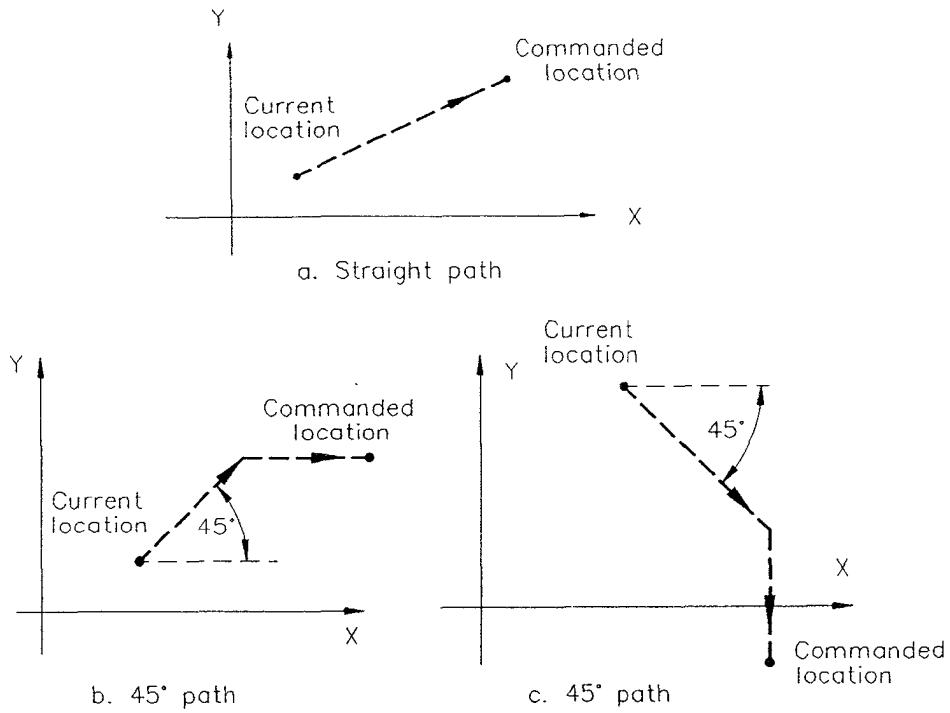
### 9.1.1 Rapid Traverse (G00)

Rapid-traverse code, G00, is often referred to as the positioning command. It directs the control unit to move the tool in rapid-traverse mode to the position specified by the coordinate word (X, Y, Z). In most modern CNC systems, the G00 code commands the tool to take the shortest path from the current position to the end position (Figure 9.1a). In some old NC systems, however, the tool will first move from the current position along a straight line at a  $\pm 45^\circ$  angle with the X axis to an intermediate location, and then move parallel to the X or Y axis until the final position is reached (Figures 9.1b and 9.1c).

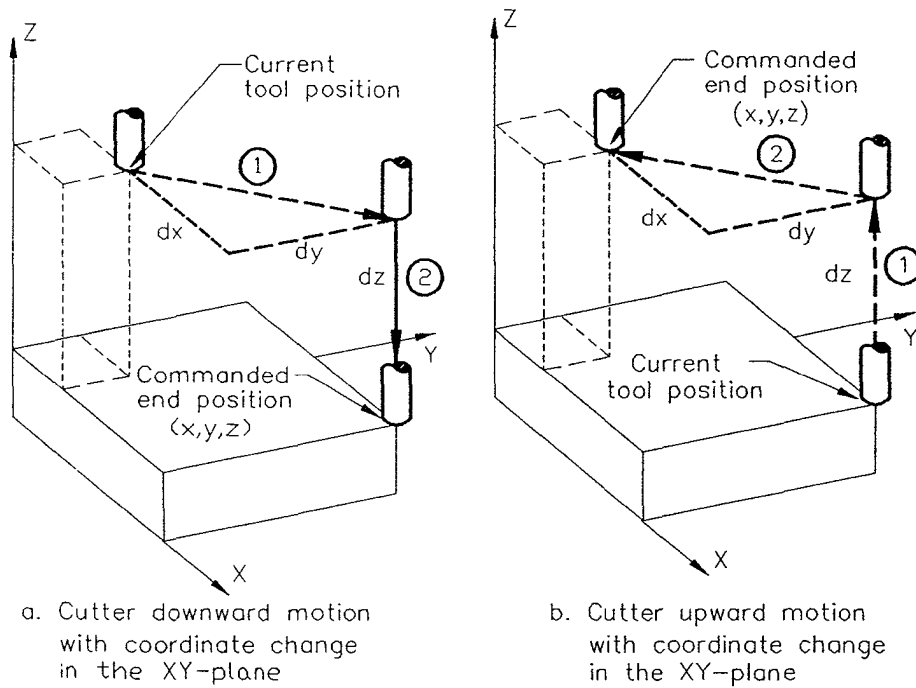
The command format of the rapid traverse is

G00 X $x$  Y $y$  Z $z$

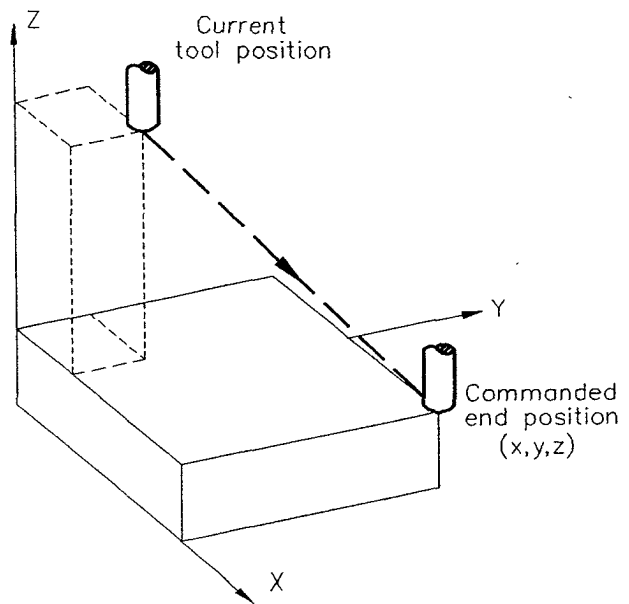
where  $x$ ,  $y$ , and  $z$  are the coordinate values of the end position in the X, Y, and Z axes, respectively. When Z movement is included along with X or Y motion, two possible motion patterns may occur, depending upon the control unit type. In some systems, the control unit directs the motion in the XY-plane first, and then executes the Z downward motion if  $z$  is decreased (Figure 9.2a). Conversely, if  $z$  in the commanded position is larger than  $z$  in the current position (spindle retraction), the upward motion is executed first, followed by the motion in the XY-plane (Figure 9.2b). For some systems, the tool takes the shortest distance (straight line) to the end position even if Z motion is involved (Figure 9.3).



**FIGURE 9.1**  
Tool motion patterns in the XY-plane for the G00 command.



**FIGURE 9.2**  
Three-dimensional motion paths with G00.



**FIGURE 9.3**  
Three-dimensional straight-line path with G00.

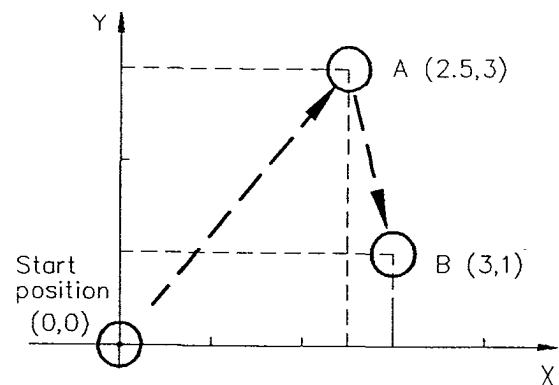
### EXAMPLE 9.1

The tool is programmed to move from the origin to A(2.5,3.0) and continue to B(3,1) in the XY-plane with the rapid-traverse mode, as shown in Figure 9.4. The command program, with decimal point, is

```
G00 X2.5 Y3.0      rapid to A
G00 X3.0 Y1.0      rapid to B
```

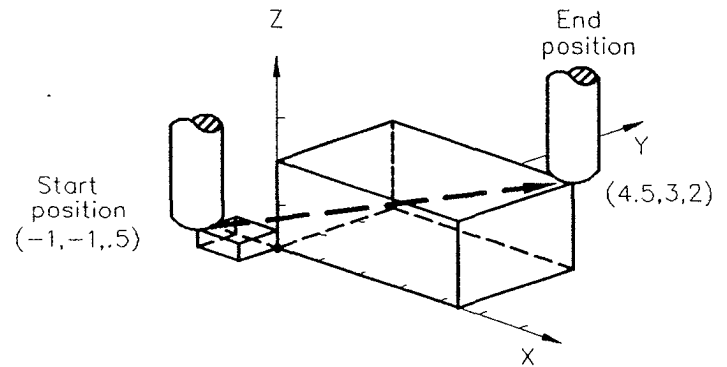
Without decimal point, the program is

```
G00 X25000 Y30000  rapid to A
G00 X30000 Y10000  rapid to B
```



**FIGURE 9.4**  
Rapid-traverse mode in the XY-plane.





**FIGURE 9.5**  
Rapid-traverse mode in  
three-dimensional space.

## EXAMPLE 9.2

Program the tool to move in rapid mode from the start position  $(-1, -1, 0.5)$  to the end position  $(4.5, 3.0, 2.0)$  shown in Figure 9.5. The command statement is

```
G00 X4.5 Y3.0 Z2.0
```

In practice, two blocks are used to achieve part clearance:

```
G00 Z2.0
G00 X4.5 Y3.0
```

### 9.1.2 Linear Interpolation (G01)

The linear interpolation code G01 moves the tool from its current position to the commanded position  $(X, Y, Z)$  in a straight line at the specified feed rate given by the address F. Its command format is

```
G01 Xx Yy Zz Ff
```

where  $x$ ,  $y$ , and  $z$  are coordinate values of the end position and  $f$  indicates the feed rate.

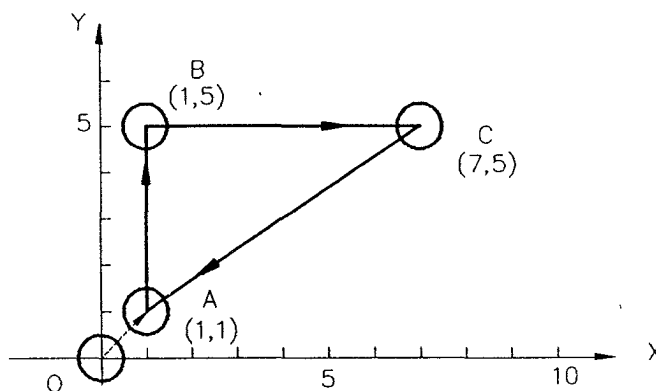
## EXAMPLE 9.3

Assume you want the cutter to move rapidly from the origin O to point A, then cut with feed control from  $A \rightarrow B \rightarrow C \rightarrow A$  (Figure 9.6). The program for this purpose is

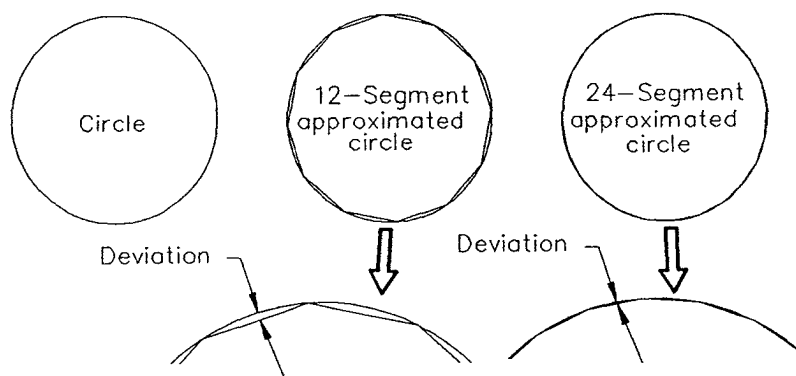
```
N05 G90          absolute coordinate system
N10 G00 X1.0 Y1.0  rapid mode, move to A(1.0,1.0)
N15 G01 Y5.0 F7.5  linear interpolation, cut to B(1.0,5.0), 7.5 IPM feed
N20 X7.0          cut to C(7.0,5.0)
N25 X1.0 Y1.0    cut to A(1.0,1.0)
```

The motion commands (G00, G01, G02, and G03), linear dimension commands (X, Y, and Z), positioning commands (G90 and G91), and some M-functions are known as *modal commands* or *functions*. Once they are used in the program, the functions remain

**FIGURE 9.6**  
G01 programming example.



**FIGURE 9.7**  
Using linear interpolation to approximate a circle.



in effect until canceled or superseded by a function of the same type. This modal feature allows the part programmer to omit repeated words and consequently reduce the program size. The word X1.0 in line N10, for example, is implicit in line N15 as a modal command. Similarly, G01, Y5.0, and F7.5 in line N15 become modal commands in line N20.

Note that the feed rate must be given when linear interpolation (G01) or circular interpolation (G02/G03) is used. Since the feed function is a modal command, it can be specified in a previous block or in the same block with the G01 or G02/G03 command.

Curves and circles can be approximated with linear interpolation by connecting a series of straight-line segments. It requires a large number of programmed points to cut a contour shape involving a curve. Figure 9.7 illustrates how a circle is approximated by 12 and 24 line segments, respectively.

### 9.1.3 Circular Interpolation (G02 and G03)

Circular interpolation commands are used to move a tool along a circular arc to the commanded end position. Five pieces of information are required for executing a circular interpolation: plane selection, arc start position coordinates, rotation direction, arc end position coordinates, and arc center coordinates, or arc radius. Table 9.1 summarizes this required circular interpolation information.

There are two basic command formats for circular interpolation, depending on whether the IJK method or the R method is used to define the arc center.

**TABLE 9.1 Required circular interpolation information**

Item	Command	Remark
Plane selection command	G17	Arc parallel to XY-plane
	G18	Arc parallel to ZX-plane
	G19	Arc parallel to YZ-plane
Arc start position coordinate	X, Y, Z	Coordinate of the arc start position
Rotation direction	G02	Clockwise direction
	G03	Counterclockwise direction
Arc end position coordinate		
Absolute (G90)	X, Y, Z	Coordinate of the end position on the work coordinate system
Incremental (G91)	X, Y, Z	Distance components from start position to end position in X, Y, and Z axes, respectively
Arc center or radius		
IJK method (arc center coordinate)	I, J, K	Distance components from start position to arc center in X, Y, and Z axes, respectively
R method (arc radius)	R	Arc radius value

**Command Format with IJK Method**

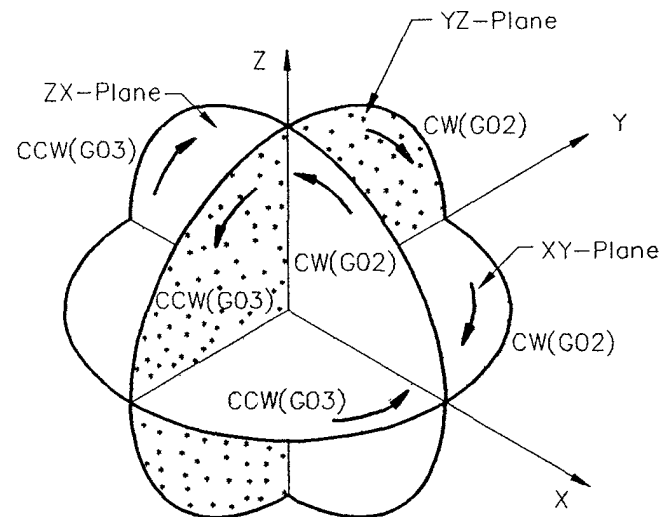
(G17) G02 (or G03)  $X_x Y_y I_i J_j F_f$       on XY-plane  
(G18) G02 (or G03)  $X_x Z_z I_i K_k F_f$       on ZX-plane  
(G19) G02 (or G03)  $Y_y Z_z J_j K_k F_f$       on YZ-plane

**Command Format with R Method**

(G17) G02 (or G03)  $X_x Y_y R_r F_f$       on XY-plane  
(G18) G02 (or G03)  $X_x Z_z R_r F_f$       on ZX-plane  
(G19) G02 (or G03)  $Y_y Z_z R_r F_f$       on YZ-plane

The plane selection command (G17, G18, or G19) is included in the circular interpolation statement to indicate which plane the arc cutting is to be executed on (see Section 9.2). The plane selection command is often omitted in the circular interpolation command statement because the use of any two of the three addresses X, Y, and Z will uniquely designate the plane on which the arc is cutting.

The arc start position is normally identified automatically at the current cutter position, which is the end point of a line or circular arc from the preceding motion statement. The controller thus takes the current position as the start point of the arc to eliminate the entry of X, Y, and Z coordinates. For this reason, the X, Y, and Z coordinates of the arc start position do not appear in the circular interpolation command statements.



**FIGURE 9.8**  
Rotation directions in three planes.

Two preparatory codes, G02 and G03, are used to determine the rotation direction of the circular arc. G02 indicates that the arc path rotates clockwise around the center, and G03 indicates a counterclockwise rotation. The direction for G02 (CW) and G03 (CCW) on the three coordinate planes can be determined by viewing from the positive direction to the negative direction on the axis perpendicular to the plane in the right-hand coordinate system. Figure 9.8 indicates rotation directions on three coordinate planes.

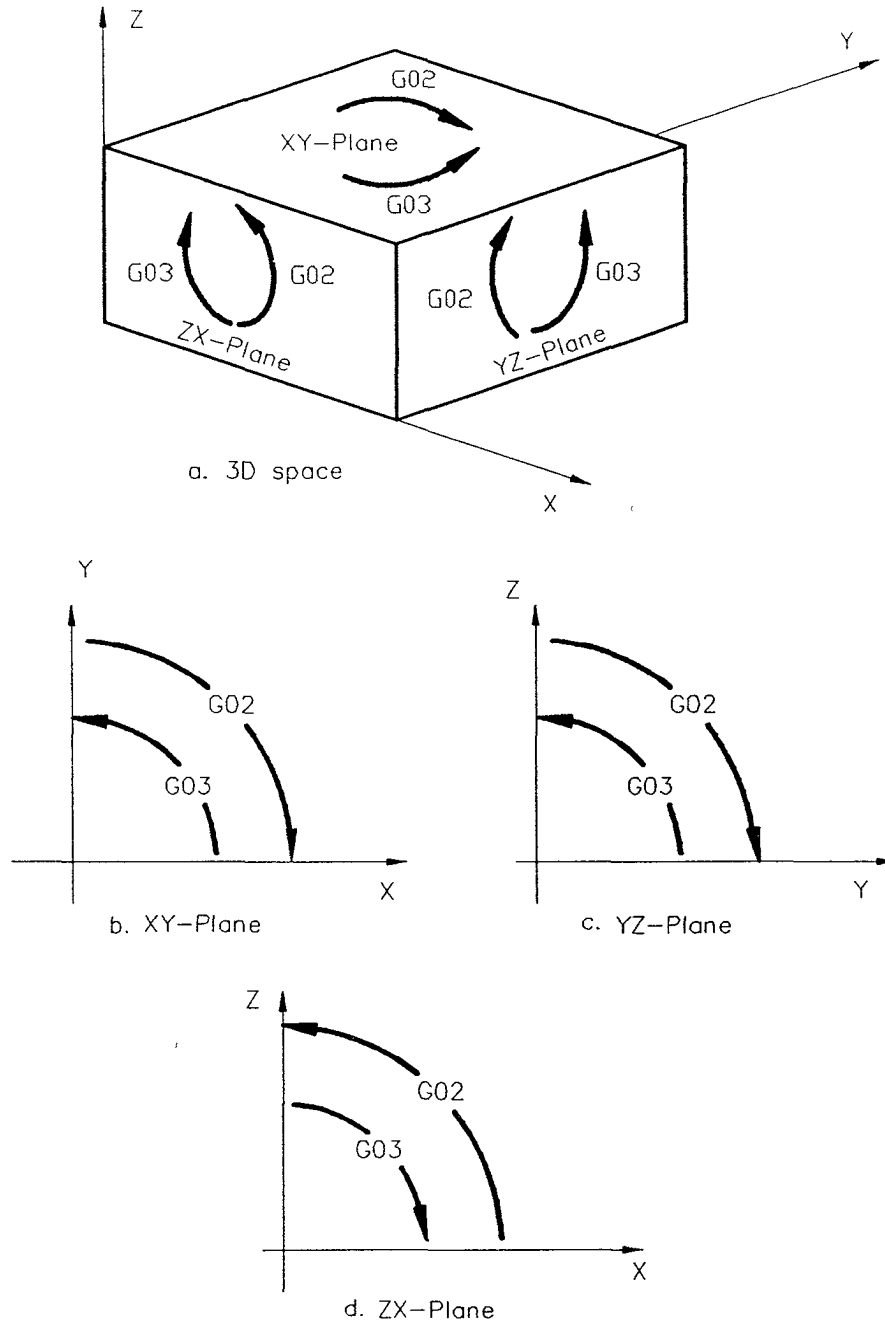
Another way to determine the rotation direction is based on the sequence of axes in plane designation. The G02 move is in the rotation direction of the second designated positive axis to the first designated positive axis. In the XY-plane, for example, the X axis is the first designated axis, and the Y axis is the second designated axis. The G02 thus rotates from the positive Y axis to the positive X axis, as shown in Figures 9.9a and 9.9b. The G03 is in the direction of the first designated axis of the plane (positive X axis in the XY-plane) to the second designated axis (positive Y axis). Figures 9.9c and 9.9d show the G02 and G03 rotation directions on the YZ- and ZX-planes, respectively.

Figure 9.10 illustrates the relationships of G02 and G03 to the three principal coordinate planes on a vertical NC machine.

The arc end position is specified by two X, Y, or Z coordinates. X and Y coordinates should be given for an arc being cut on the XY-plane; similarly, Y and Z coordinates are specified for an arc on the YZ-plane, and Z and X values for an arc on the ZX-plane. (For some controllers with the capability of helical cutting, X, Y, and Z are all programmed to cut a circular arc on a specified plane and a linear interpolation on the third axis.) With the absolute coordinate system (G90), the arc end position coordinates X, Y, and Z are measured from the program origin. In the incremental coordinate system (G91), they are measured from the start position.

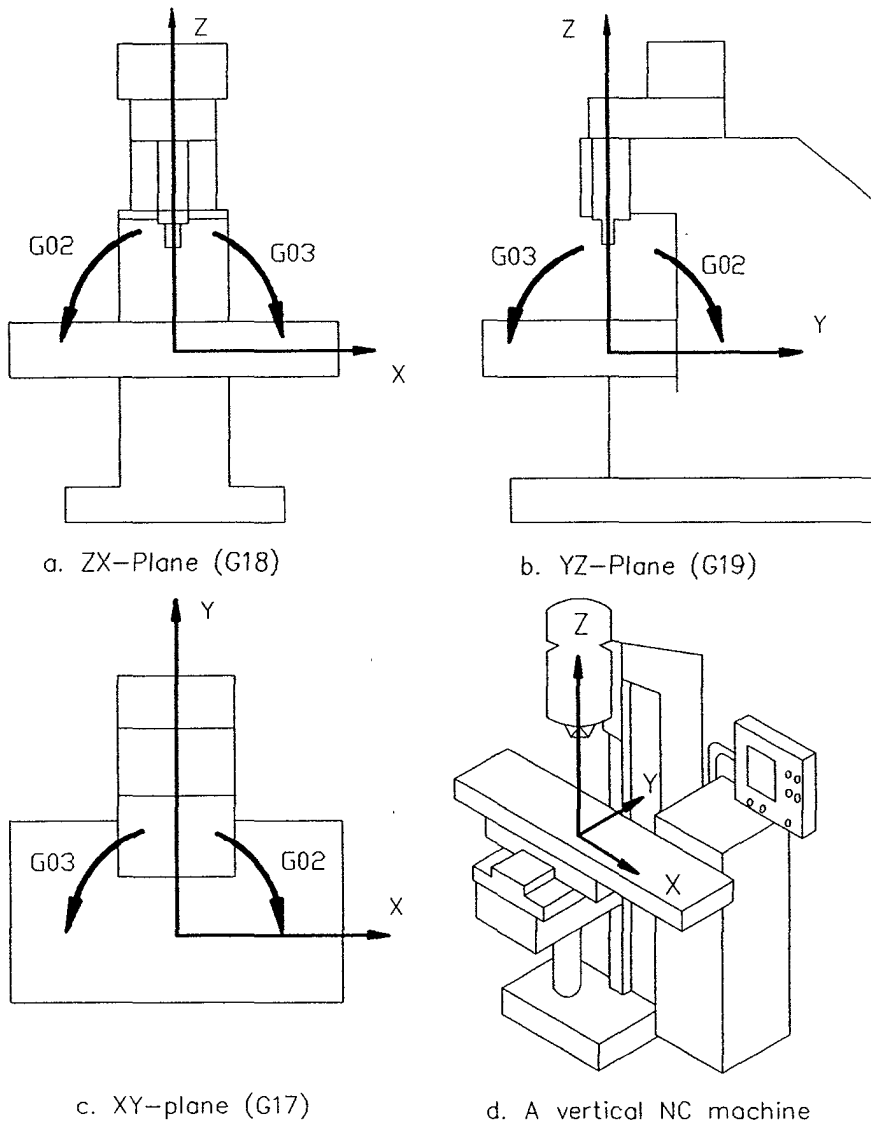
### IJK Method

With the IJK method, the I, J, and K addresses are used to specify the arc center coordinates in the X, Y, and Z directions. The I, J, and K values are usually measured incrementally from the start position to the center of the arc in the X, Y, and Z directions, respectively.



**FIGURE 9.9**  
Determination of rotation direction based on sequence of axes.

The I, J, and K values can be thought of as components of the vector being drawn from the start position to the center position of the arc in the X, Y, and Z directions, respectively, as shown in Figure 9.11. The I, J, and K values are always measured incrementally for either the absolute or the incremental coordinate mode.



**FIGURE 9.10**  
Directions of G02 and G03 on a vertical NC machine.

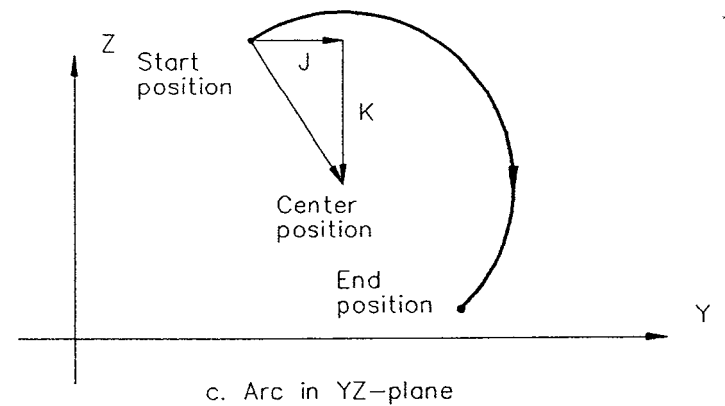
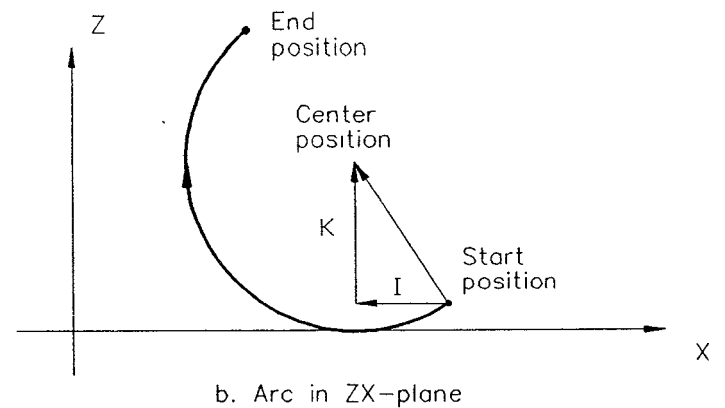
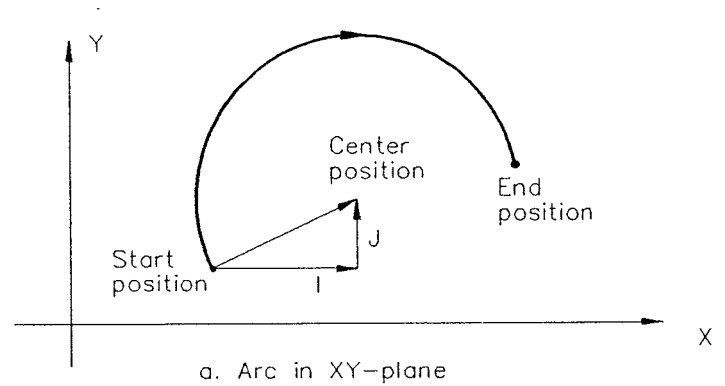
The I, J, and K values can be readily obtained from the following equations:

$$I = X_c - X_s$$

$$J = Y_c - Y_s$$

$$K = Z_c - Z_s$$

where  $X_c$ ,  $Y_c$ , and  $Z_c$  are the coordinates of the arc center and  $X_s$ ,  $Y_s$ , and  $Z_s$  are the coordinates of the arc start position.



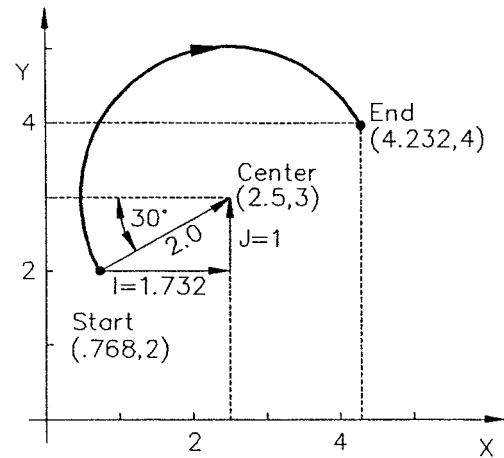
**FIGURE 9.11**  
Determination of I, J, and K  
with a given arc.

## EXAMPLE 9.4

The circular arc shown in Figure 9.12 has a radius of 2 in. It starts at (0.768,2) and ends at (4.232,4). The center coordinates in the X and Y directions are (2.5,3). Thus,  $X_C = 2.5$ ,  $Y_C = 3$ ,  $X_S = 0.768$ , and  $Y_S = 2$ . I and J are then computed as

$$I = X_C - X_S = 2.5 - 0.768 = 1.732$$

$$J = Y_C - Y_S = 3 - 2 = 1$$



**FIGURE 9.12**  
An example of calculating I and J.

Some controllers will not allow an arc of over  $90^\circ$  to be cut in a single circular interpolation command block. For arcs larger than  $90^\circ$ , including even full circles, two or more consecutive circular interpolation blocks should be used. Most modern CNC controllers, however, are capable of producing a full circle in a single circular interpolation block.

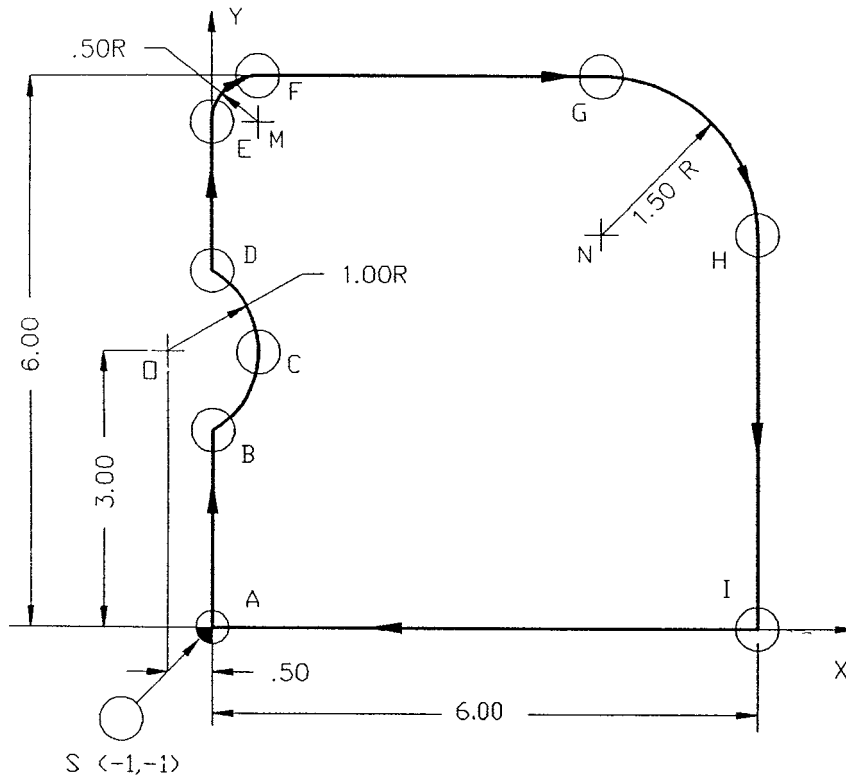
## EXAMPLE 9.5 ARC PROGRAMMING WITH THE IJK METHOD

The workpiece shown in Figure 9.13 consists of three circular arcs and four straight lines. The arc with 1 in. radius spans more than  $90^\circ$  but less than  $180^\circ$ ; it thus requires two circular interpolation blocks for some NC controllers. The tool path is shown in Figure 9.13. To make the program simpler, cutter diameter compensation is not included in this example.

The critical programming dimensions for this workpiece are as follows:

Position (location)	X coordinate	Y coordinate
A	0	0
B	0	2.134
C	0.5	3.0
D	0	3.866
E	0	5.5
F	0.5	6.0
G	4.5	6.0
H	6.0	4.5
I	6.0	0
M	0.5	5.5
N	4.5	4.5
O	-0.5	3.0



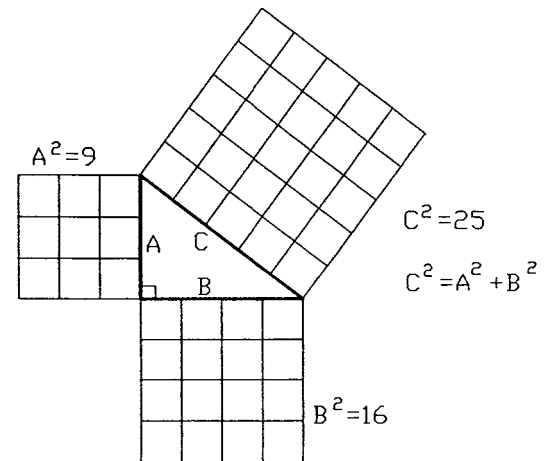


**FIGURE 9.13**  
A programming example using the IJK method.

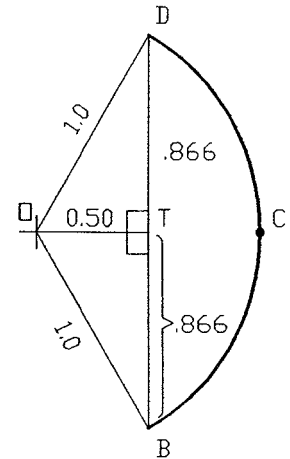
Finding the coordinate values at positions B and D requires use of the Pythagorean theorem: the sum of the squares of two lesser sides of a right triangle is equal to the square of the hypotenuse. The formula is

$$A^2 + B^2 = C^2$$

where *C* is the hypotenuse (see Figure 9.14).



**FIGURE 9.14**  
Pythagorean theorem.



**FIGURE 9.15**  
Pythagorean relation for a given arc and chord.

To find the coordinate values for position at B (Figure 9.13), we can construct a right triangle OBT as shown in Figure 9.15 and then apply the Pythagorean relationship to obtain

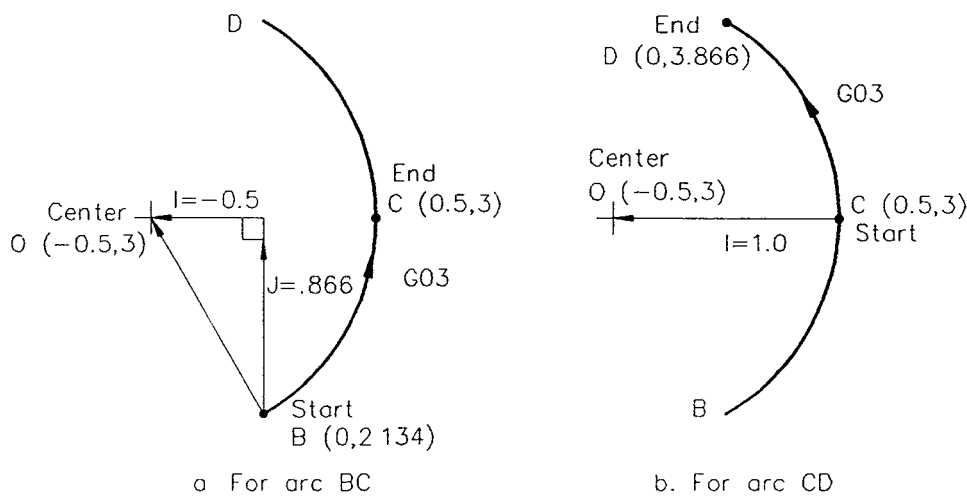
$$\overline{OB}^2 = \overline{BT}^2 + \overline{OT}^2$$

where  $\overline{OB}$  is the radius of the arc, which has the value of 1, and  $\overline{OT}$  has a given value of 0.5. The unknown distance  $\overline{BT}$  can be obtained by rearranging the above equation and substituting the given values as follows:

$$\overline{BT} = \sqrt{\overline{OB}^2 - \overline{OT}^2} = \sqrt{1^2 - 0.5^2} = 0.866$$

Thus, the Y coordinate at position B is 2.134 (or 3.0 - 0.866). Similarly, the Y coordinate at position D is found to be 3.866 (or 3.0 + 0.866).

The determination of the I and J vector component values for cutting the arc BC is illustrated in Figure 9.16a. The arc start point is (0, 2.134), and the arc center is (-0.5, 3.0).



**FIGURE 9.16**  
Calculating for I and J for arc BD of Example 9.5.

The I and J values can be found as follows:

$$I = X_c - X_s = -0.5 - 0 = -0.5$$

$$J = Y_c - Y_s = 3.0 - 2.134 = 0.866$$

Similarly, the I and J values for the arc CD (Figure 9.16b) are

$$I = X_c - X_s = -0.5 - 0.5 = -1.0$$

$$J = Y_c - Y_s = 3.0 - 3.0 = 0$$

The following program, which is programmed in the absolute coordinate mode, can be used to machine the part:

N50 G90	absolute coordinate mode
N55 G00 X-1.0 Y-1.0	rapid traverse to S(-1,-1)
N60 G01 X0 Y0 F7.5	linear feed mode, cut to A, feed rate 7.5 IPM
N65 Y2.134	cut straight line to B (start point of the arc BC)
N70 G03 X0.5 Y3.0 I-0.5 J0.866	circular interpolation, cut arc BC to point C
N75 X0.0 Y3.866 I-1.0 J0	cut arc CD to point D
N80 G01 Y5.5	linear feed mode, cut straight line to E (start point of the arc EF)
N85 G02 X0.5 Y6.0 I0.5 J0	circular interpolation, cut arc EF to point F
N90 G01 X4.5	linear feed mode, cut straight line to G (start point of the arc GH)
N95 G02 X6.0 Y4.5 I0 J-1.5	circular interpolation, cut arc GH to point H
N100 G01 Y0	linear feed mode, cut straight line to point I
N105 X0	cut straight line to point A
N110 G00 X-1.0 Y-1.0	rapid-traverse mode, move to initial tool position S

Note that the arc BD can be programmed in a single circular interpolation block in some modern CNC controllers. In this case, the command statements in N70 and N75 can be combined as

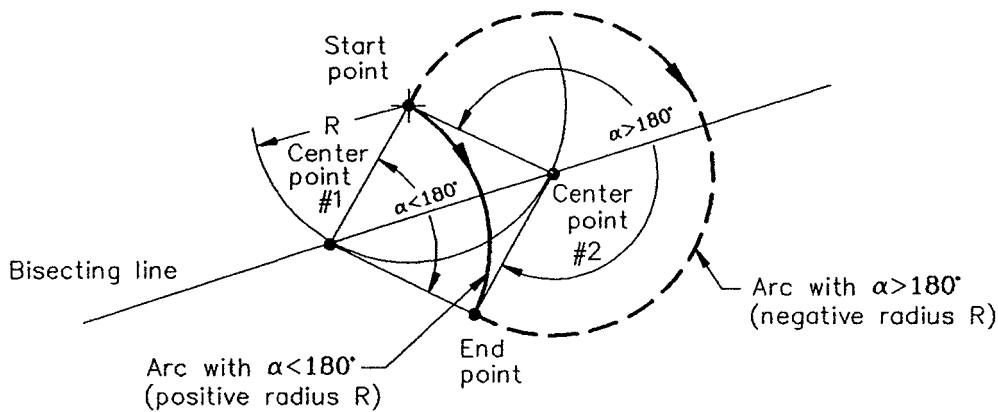
```
N70 G03 X0 Y3.866 I-0.5 J0.866
```

## R (Radius) Method

The radius method for circular interpolation requires two entry parameters in the command statement: the coordinates of the end point and the radius value. Recall that its command format for an arc in the XY-plane is

G02 (or G03) Xx Yy Rr Ff

The arc center can be mathematically determined by constructing a line bisecting the chord that connects the start point and the end point. Then the arc start point can be used as the center of an arc that intersects the bisecting line at two points. Two possible arcs can be defined with these two points as the arc centers, as shown in Figure 9.17. One



**FIGURE 9.17**  
Signs of radius R.

arc spans an angle that is greater than  $180^\circ$  (or more than half a circle), and the other arc covers the angle equal to or less than  $180^\circ$  (half a circle or less). Positive and negative R values are used to distinguish one from the other. A radius programmed with a positive value is for the arc that is half a circle or less, and the negative radius value is for the arc greater than half a circle.

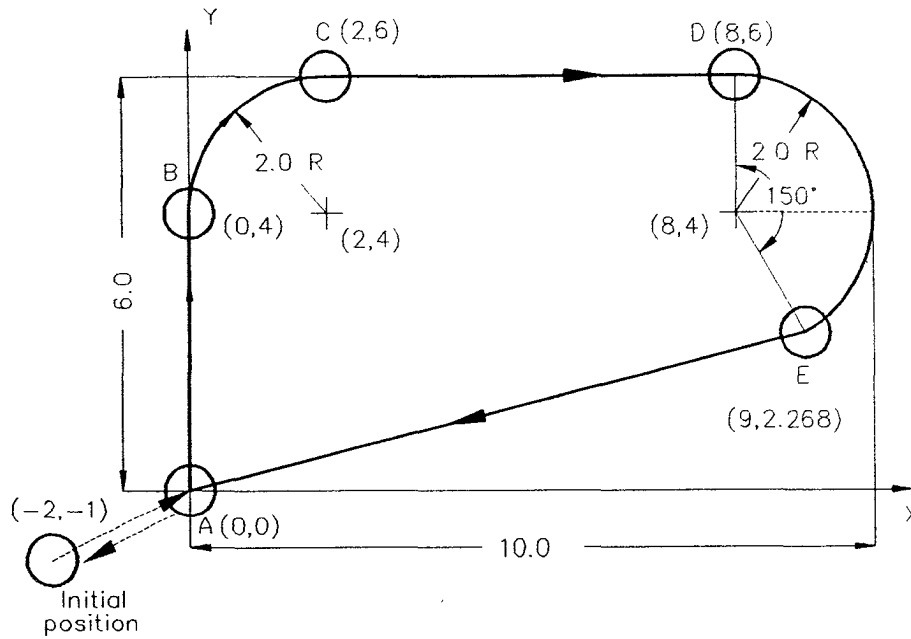
## EXAMPLE 9.6 ARC PROGRAMMING WITH THE R METHOD

The radius method of circular interpolation is to be used to program two circular arcs on the workpiece shown in Figure 9.18. The cutter path is shown in the figure, and absolute coordinate programming is to be used.

N50 G90 G00 X-2.0 Y-1.0	absolute positioning mode, rapid traverse to initial position (-2,-1)
N55 G01 X0 Y0 F8.0	linear interpolation mode, cut straight line to A, feed rate 8 IPM
N60 Y4.0	cut straight line to B (start point of arc BC)
N65 G02 X2.0 Y6.0 R2.0	circular interpolation (CW), cut arc BC to point C
N70 G01 X8.0	linear feed mode, cut straight line to D (start point of arc DE)
N75 G02 X9.0 Y2.268 R2.0	circular interpolation (CW), cut arc DE to point E
N80 G01 X0 Y0	linear feed mode, cut straight line to A
N85 G00 X-2.0 Y-1.0	rapid mode, move to initial position

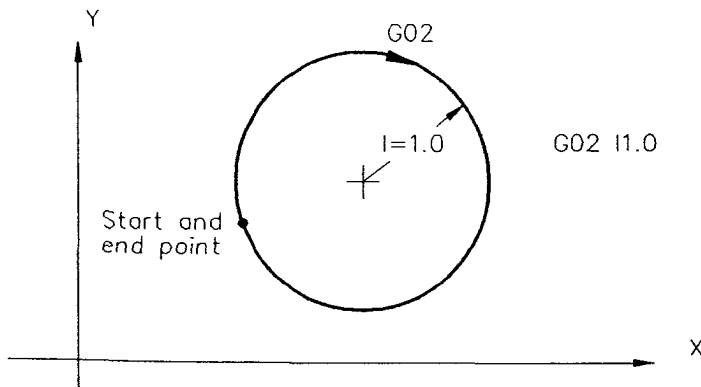
Here are some suggestions for programming circular arcs:

1. In a circular interpolation statement, I0, J0, and K0 may be omitted from the statement because their values are zero.
2. For cutting a full circle, the X, Y, and Z commands can be omitted because the start point and end point are identical. The statement G02 I1.0, for example, will cut a full circle with the radius of 1.0, as shown in Figure 9.19.

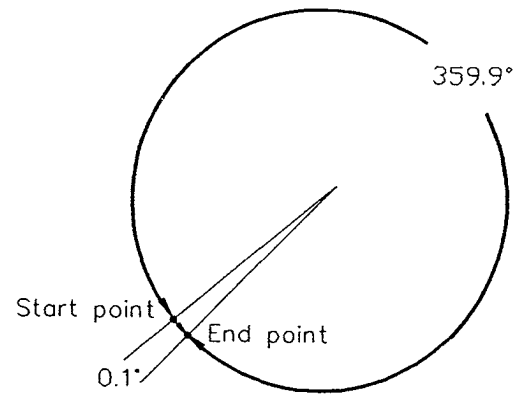


**FIGURE 9.18**  
Circular interpolation programming with the R method.

3. When the command address R is used to specify the arc radius, the arc must be less than 359.9°. If an arc spanning an angle of 359.9° or more is commanded with R, the system will reverse the rotation direction, and the result will be an arc that is less than 0.1° (Figure 9.20). Two consecutive arc interpolation blocks with the use of an R address may be programmed to cut an arc between 359.9° to 360°.
4. When I, J or K, and R are specified in the same circular interpolation block, the R command will be chosen.

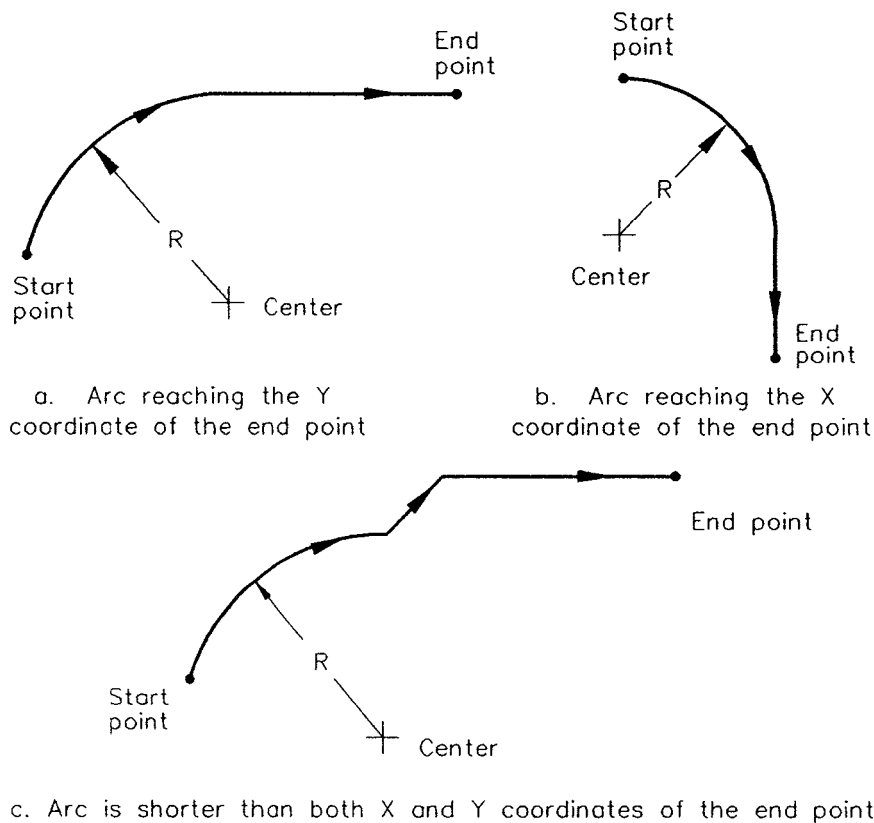


**FIGURE 9.19**  
Cutting a complete circle.

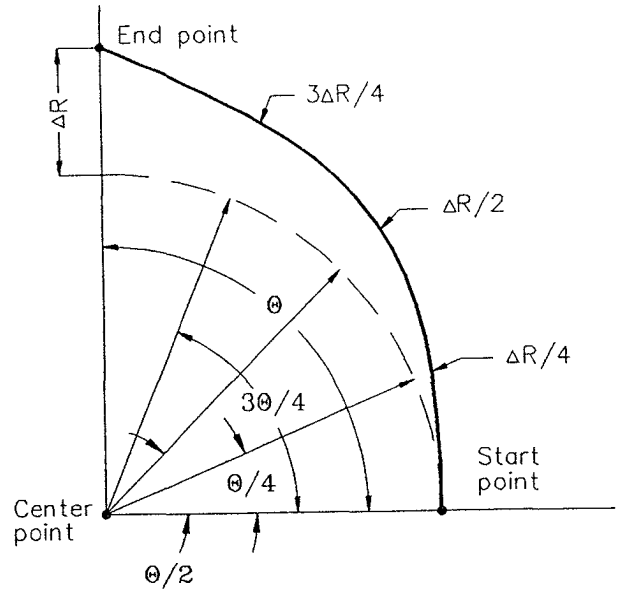


**FIGURE 9.20**  
Valid arc angle with R method.

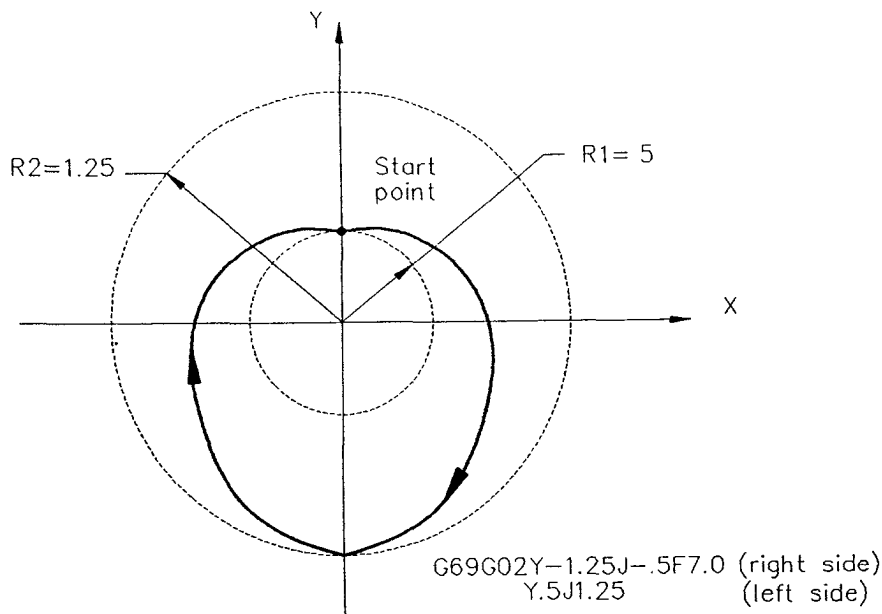
5. When the commanded arc end point is not on the circumference, the system may respond differently from controller to controller. Here are some typical responses:
- A specific condition must be satisfied for circular interpolation programming in some controllers, for example,  $0.001 \leq L/2R \leq 1$ , where  $L$  is the distance from start point to end point.
  - If the arc is longer than one coordinate of the arc end point, the tool will move along with the commanded circular arc to reach one axis coordinate value and then move along the other axis to the end point (Figures 9.21a and 9.21b). If the arc is shorter than both X and Y coordinates of the arc end point, the tool will follow the path shown in Figure 9.21c.
  - Some controllers furnish a special G-code—for example, G69 in Mitsubishi Meldas-M0—to cancel the end-point error check. This feature allows an arc interpolation to be executed for arcs whose end points are not on their circumference. In these cases, the arc radius varies linearly along with the arc, as shown in Figure 9.22. This feature can be applied to heart cam cutting, taper thread cutting, cone cutting, and the like. Figure 9.23 illustrates the use of radius difference to cut a heart-shaped cam.

**FIGURE 9.21**

Tool movements when the commanded end point is not on the arc.



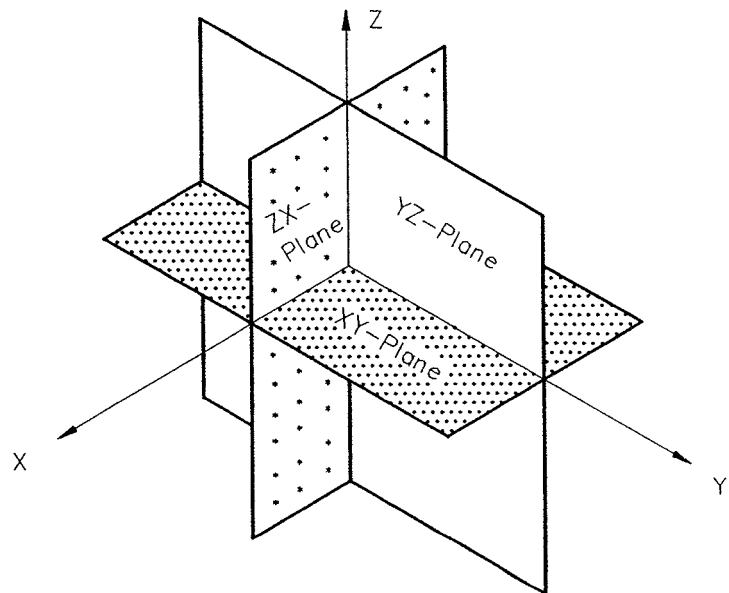
**FIGURE 9.22**  
Arc radius varies linearly when the end point is not on the commanded arc's circumference.



**FIGURE 9.23**  
Cutting a heart-shaped cam using the radius variance method.

## 9.2 PLANE SELECTION

The working coordinate of an NC machine tool is defined by three primary axes: X, Y, and Z. These three axes are mutually perpendicular and form three primary planes: the XY-plane, the ZX-plane, and the YZ-plane (Figure 9.24). Each primary plane contains two axes and is perpendicular to the third axis. The XY-plane, for example, includes the X and Y axes and is perpendicular to the Z axis. Machining actions usually take place on



**FIGURE 9.24**  
Three primary planes.

selected plane. The plane on which the cutting actions occur is selected by two methods: G-code priority plane selection and axis-address priority plane selection.

### 9.2.1 G-Code Priority Plane Selection

The three primary planes can be specified by the following three G codes:

G17      XY-plane selection  
G18      ZX-plane selection  
G19      YZ-plane selection

### 9.2.2 Axis Address Priority Plane Selection

The selection of a machining plane can also be determined by two axis addresses included in the block statement. The axis addresses that can be used for executing plane selection include X, Y, Z, and additional axis addresses U, V, or W if available. Neither the angular axis addresses A, B, and C nor the arc center addresses I, J, and K can be used for selecting the machining plane. Some valid axis-address priority plane selection examples are

G02 X<sub>x</sub> Y<sub>y</sub> I<sub>i</sub> J<sub>j</sub> F<sub>f</sub>      XY-plane selection  
G03 X<sub>x</sub> Z<sub>z</sub> I<sub>i</sub> K<sub>k</sub> F<sub>f</sub>      ZX-plane selection  
G03 Y<sub>y</sub> Z<sub>z</sub> J<sub>j</sub> K<sub>k</sub> F<sub>f</sub>      YZ-plane selection

Axis-address priority plane selection is not a modal command and is thus effective for only one block. It is possible to use a combination of G-code selection and axis-address selection modes for certain types of applications. If there is a conflict between the commanded planes in the same block for a circular interpolation or a cutter diameter compensation command, the plane commanded by the two axis addresses will be effective for that particular block and for that block only. The axis-address plane command mode does not override the G-code selection mode for the rest of the program.



TABLE 9.2 Plane Selection Table

Axis addresses in one block	Plane selection G code		
	G17 (XY-plane)	G18 (ZX-plane)	G19 (YZ-plane)
X	XY-plane	ZX-plane	YZ-plane <sup>a</sup>
Y	XY-plane	ZX-plane <sup>a</sup>	YZ-plane
Z	XY-plane <sup>a</sup>	ZX-plane	YZ-plane
X Y	XY-plane	XY-plane	XY-plane
X Z	ZX-plane	ZX-plane	ZX-plane
Y Z	YZ-plane	YZ-plane	YZ-plane
X Y Z	XY-plane <sup>a</sup>	ZX-plane <sup>a</sup>	YZ-plane <sup>a</sup>

<sup>a</sup>Indicates the combination with helical cutting in G02/G03.

The combined effect of G-code plane selection and axis-address plane selection is summarized in Table 9.2.

Here are three plane selection examples:

G17

G03 X<sub>x</sub> Z<sub>z</sub> I<sub>i</sub> J<sub>j</sub> F<sub>f</sub>      arc interpolation to be executed in the ZX-plane

G17

G03 X<sub>x</sub> Y<sub>y</sub> Z<sub>z</sub> R<sub>r</sub> F<sub>f</sub>      helical cutting with circular motion in the XY-plane and  
linear motion in Z axis

G17 G01 X<sub>x</sub> Z<sub>z</sub> F<sub>f</sub>      linear interpolation in the ZX-plane

Plane selection has a significant effect on the cutter diameter compensation functions, which are discussed in Chapter 10.

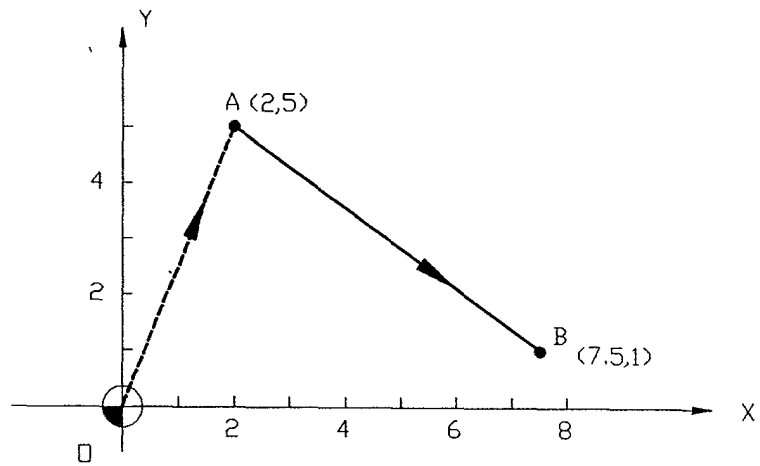
### 9.3 POSITIONING SYSTEMS (G90 AND G91)

As indicated previously, there are two positioning systems: absolute positioning and incremental positioning. The G90 code is used to specify the absolute system, G91 the incremental system. In the absolute positioning mode, the tool moves to the commanded position defined by the coordinates' absolute values measured from a fixed point (called the origin) on the work coordinate system. With incremental positioning mode, the current tool position is taken as the reference point and the tool moves the distance given in the command. The command format of positioning system is

G90 (or G91) G00 (or G01, G02, or G03) X<sub>x</sub> Y<sub>y</sub> Z<sub>z</sub>

The motion command code determines whether the motion type is rapid traverse (G00), linear interpolation (G01), or circular interpolation (G02/G03).

G90 and G91 commands are of the modal type; that is, they are effective for succeeding blocks until canceled.



**FIGURE 9.25**  
Programming with the  
absolute positioning system.

### EXAMPLE 9.7

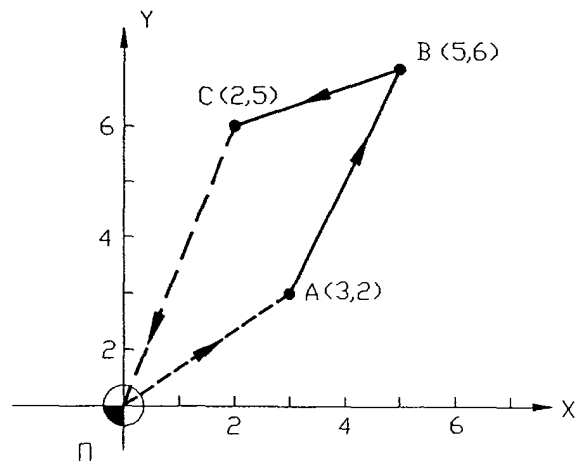
The following two program blocks are used to move the tool from the origin to point A and then cut to point B with the absolute positioning system (Figure 9.25). For simplicity, no Z axis movement is included.

```
N05 G90 G00 X2.0 Y5.0      absolute mode, rapid to A(2.0,5.0)
N10 G01 X7.5 Y1.0 F7.0    absolute mode, feed to B(7.5,1.0)
```

### EXAMPLE 9.8

The following program is used to direct the tool to move rapidly from the origin to A(3,2), then feed to B(5,6) and to C(2,5), and finally return rapidly to the origin (Figure 9.26). Both absolute and incremental modes are used in this program.

```
N05 G90 G00 X0 Y0          absolute mode, rapid to O(0,0)
N10 G91 X3.0 Y2.0          incremental mode, rapid to A(3.0,2.0)
N15 G1 X2.0 Y4.0 F10.0     incremental mode, feed to B(5.0,6.0)
```



**FIGURE 9.26**  
Cutting a contour with both incremental  
and absolute positioning systems.

```

N20 X-3.0 Y-1.0          incremental mode, feed to C(2,5)
N25 G90 G00 X0 Y0       absolute mode, rapid to O(0,0)

```

As demonstrated in Example 9.8 both absolute and incremental positioning modes may be used in the same program. Note, however, that the old generation of NC controls furnished only one positioning mode, either absolute or incremental but not both. Most modern CNC controls provide both modes to facilitate the dimensional features of the workpiece.

---

## 9.4 INPUT UNIT SELECTION (G70 AND G71)

There are two common length unit systems used in NC work: the inch system and the metric system. G70 code is used to specify the inch system, G71 the metric system. The unit in the inch system, is, of course, the inch, and the unit for the metric system is the millimeter. In some controls, G20 is used for the inch system and G21 for the metric system. Unit selection commands are of the modal type.

As discussed in Chapter 5, the minimum input value of dimension words (X, Y, Z, I, J, K, etc.) varies from one unit system to the other. Most modern CNC controls have a resolution of 0.0001 in. (minimum setting unit) for the inch system and 0.001 mm for the metric unit. The selection of unit has an effect on the following operations:

1. Feed rate function (F)
2. Position dimension (X, Y, Z, I, J, K)
3. Compensation and offset value (registers D and H)
4. Parameter setting

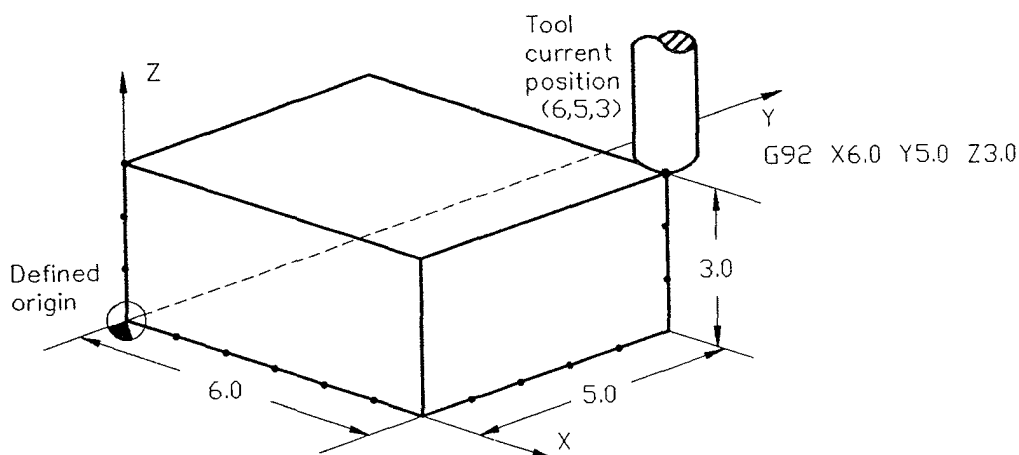
The general rules for selecting a unit system are as follows:

1. The default unit selection can be set by parameter. In some controls, the default unit is the unit used by the part program previously run on the machine. If the previous program used G70 (inch), the default unit will be G70. If the previous program was G71 (metric), the unit will be G71. The default unit selection may vary from one control to another.
2. Generally speaking, G70 and G71 codes cannot be used in the same part program. In some CNC controllers, switching back and forth between G70 and G71 in the same part program is allowed, with the condition that the minimum setting unit be set properly. This is especially important for converting the amount of compensation for tool length, tool position, and tool radius.

---

## 9.5 ABSOLUTE ZERO SETTING (G92)

A program coordinate system must be defined before any motion statement can be specified. All NC machine tools have a fixed machine origin, but this origin is seldom used. The control normally allows selection of any desired position as the program origin. Establishing the origin of the part program on the machine for absolute programming is referred to as absolute zero setting, work origin setting, program origin setting, absolute register preset, or position absolute coordinates setting. The preparatory code G92 is used

**FIGURE 9.27**

The use of G92 in setting the program origin.

to directly or indirectly define the origin of the absolute coordinate system. It has the following command format:

$$G92 X_x Y_y Z_z$$

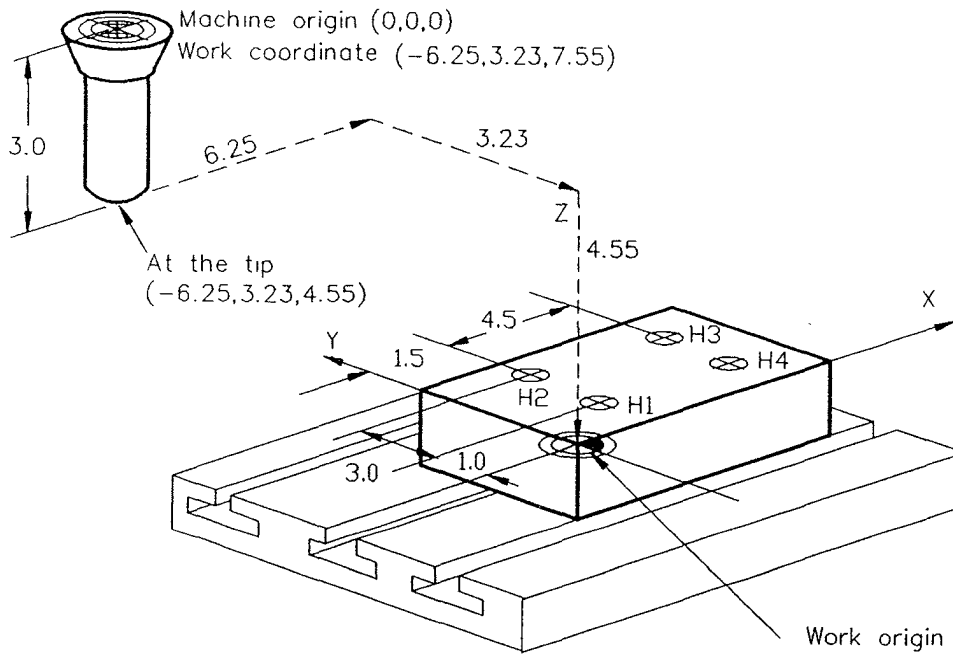
where  $x$ ,  $y$ , and  $z$  are the coordinate values of the current tool position in the defined coordinate system. The G92 code does not cause any tool motion; it simply indicates where the tool is located relative to the program zero. In a sense, this command tells the control system where the program origin is in relation to the current tool position. The use of G92 in setting the program origin is illustrated in Figure 9.27. The tool is moved to the position as indicated and then uses the following command statement to set the program origin:

$$G92 X6.0 Y5.0 Z3.0$$

In this example, the program origin is located 6 in. in the  $-X$  direction, 5 in. in the  $-Y$  direction, and 3 in. in the  $-Z$  direction from the current tool position.

The G92 code is commonly used for two reasons: to set the program or part origin, and to compensate for the difference between the program origin and the work origin. Most CNC machines require the table and spindle to be returned to the machine home (origin) position before the program can be run. This machine home position is fixed. Consider a workpiece clamped on the machine table as shown in Figure 9.28. Assume that the work origin is located on the top surface of the left front corner. The distances from the selected work origin to the machine origin are  $-6.25$  in.,  $3.23$  in., and  $7.55$  in. in the X, Y, and Z directions, respectively. (The  $-$  sign indicates that the distance is measured in the negative direction.) There are two ways to handle the tool length (assumed here to be 3.0 in.) One is to use the tool length offset command (G43 or G44) to compensate for the tool length on the Z distance between the machine origin and work origin. This method is discussed in Chapter 10. The other way is to subtract the tool length from the Z distance (between the machine origin and the work origin) and then use this distance difference as the value for Z in the G92 command. With this method the G92 command to set the work origin in this example becomes

$$G92 X-6.25 Y3.23 Z4.55$$



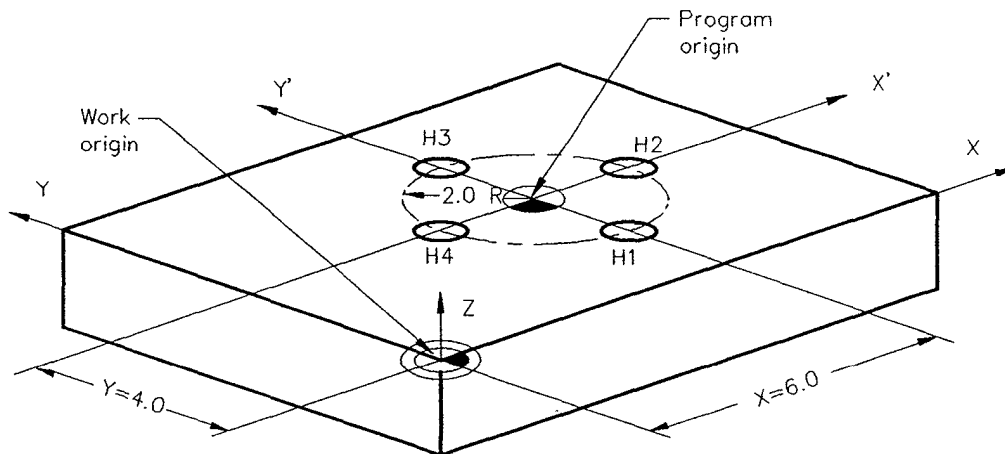
**FIGURE 9.28**  
Using G92 to set a work origin.

As indicated previously, the G92 command does not cause any tool motion; it simply defines the work coordinate system. This action is often called “zero shift.”

The complete program to drill the four holes on the workpiece shown in Figure 9.28 is as follows:

N05 G90 G17	absolute positioning, XY-plane
N10 T1 M6	tool change to T1
N15 G92 X-6.25 Y3.23 Z4.55	set program origin
N20 G00 X1.5 Y1.0	move rapidly to center of H1
N25 Z0.2	move rapidly to Z0.2
N30 G01 Z-0.5 F7.5	drill H1 to depth of -0.5, feed rate 7.5 IPM
N35 G00 Z0.2	retract tool rapidly to Z0.2
N40 Y4.0	move rapidly to center of H2
N45 G01 Z-0.5	drill H2 to depth of -0.5
N50 G00 Z0.2	retract tool rapidly to Z0.2
N55 X6.0	move rapidly to center of H3
N60 G01 Z-0.5	drill H3 to depth of -0.5
N65 G00 Z0.2	retract tool rapidly to Z0.2
N70 Y1.0	move rapidly to center of H4
N75 G01 Z-0.5	drill H4 to depth of -0.5
N80 G00 Z0.2	retract tool rapidly to Z0.2
N85 X-6.25 Y3.23 Z4.55	move rapidly to machine origin
N90 G92 X0 Y0 Z0	reset machine coordinate system
N95 M30	end of program

Often the work origin does not coincide with the program origin; this situation is shown in Figure 9.29. The part programmer may choose any convenient point as the program origin to reduce the coordinate calculation effort, and the machine operator could select a different point located on the datum surface as the work origin. The coordinate



**FIGURE 9.29**  
Work origin different from program origin.

difference between the work origin and the program origin can be resolved simply by inserting a G92 command into the part program. In Figure 9.29, for example, the discrepancy between the two origins is 6.0 in. and 4.0 in. in the X and Y directions, respectively. The complete program for this project is as follows:

N5 G90 G17	absolute positioning, XY-plane
N10 T2 M6	tool change to T2
N15 G00 X0 Y0	move rapidly to work origin
N20 G92 X-6.0 Y-4.0	set program origin in relation to part coordinate system
N25 G00 X0 Y-2.0	move rapidly to center of H1
N30 G01 Z-0.5 F5.0	drill H1 to depth of -0.5; feed rate is 5 IPM
N35 G00 Z0.2	retract tool rapidly to Z0.2
N40 X2.0 Y0	move rapidly to center of H2
N45 G01 Z-0.5	drill H2 to depth of -0.5
N50 G00 Z0.2	retract tool rapidly to Z0.2
N55 X0 Y2.0	move rapidly to center of H3
N60 G01 Z-0.5	drill H3 to depth of -0.5
N65 G00 Z0.2	retract rapidly to Z0.2
N70 X-2.0 Y0	move rapidly to center of H4
N75 G01 Z-0.5	drill H4 to depth of -0.5
N80 G00 Z0.2	retract rapidly to Z0.2
N85 X-6.0 Y-4.0	move rapidly to part origin
N90 G92 X0 Y0	return to part coordinate
:	
:	

## 9.6 REFERENCE POINT (ORIGIN) RETURN (G28, G29, AND G30)

The reference point is a fixed position to which a machine tool can be moved by the reference point return function. There are up to four reference points available in CNC machines, called the first, second, third, and fourth reference points. The first reference

point is simply referred to as reference point and is normally located at the machine origin. The other three reference points can be set to any convenient position with parameter setting. Reference point return is primarily used to (1) return the spindle to home position for tool change, (2) move the tool to a convenient position for starting a new machining operation, or (3) park the spindle and machine table at home position at the end of the part program. The spindle must return to home position before the tool change command can be issued. Three G codes are used to execute reference point return in the part program: G28, G29, and G30.

### 9.6.1 Automatic Reference Point Return (G28)

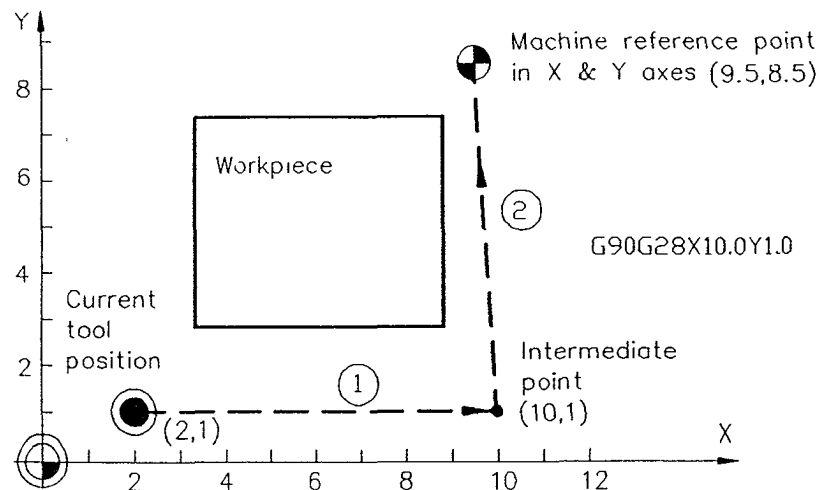
The G28 code directs the control to move the tool rapidly from the current position to an intermediate point and then back to the machine origin. The machine origin is normally called the first reference point or tool change point. The reference point return can be executed in any selected single axis, a combination of any two axes, or three axes simultaneously. It has one of the following command formats:

G28 Xx Yy Zz	three axes
G28 Xx Yy	two axes
G28 Xx Zz	two axes
G28 Yy Zz	two axes
G28 Xx (or Yy or Zz)	one axis

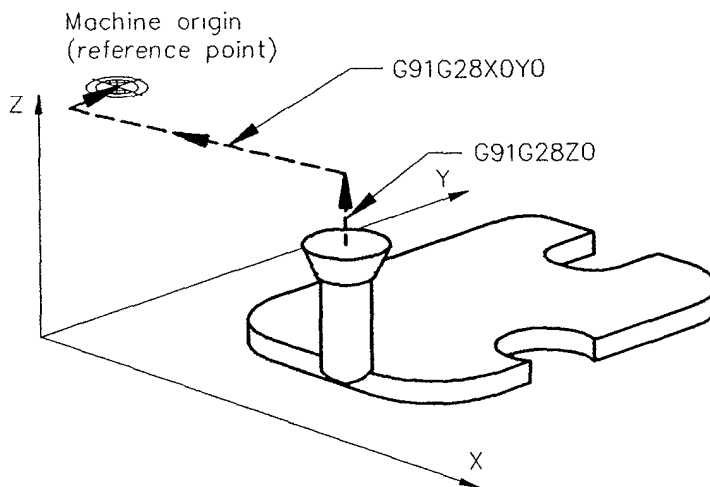
where  $x$ ,  $y$ , and  $z$  are the coordinates of the intermediate point. The intermediate point is used to clear workpiece and clamping obstructions that may interfere with the tool path. Figure 9.30 shows a typical tool path for the reference point return command. In this example, the tool is currently located at (2,1) and is intended to be returned to the machine origin (9.5,8.5). The intermediate point is chosen to be (10,1) to prevent the tool from clashing with the workpiece. The reference point return command block

```
G90 G28 X10.0 Y1.0
```

generates the tool path shown in Figure 9.30. The tool moves rapidly from the current tool position (2,1) to the intermediate point (10,1) and then to the machine origin (9.5,8.5).



**FIGURE 9.30**  
Tool path of G28 command  
(reference point return).



**FIGURE 9.31**  
Using the G91 G28 command to return the tool to home.

The intermediate point can be located at any convenient location. However, attention must be given to interference with the workpiece. The reference return function is often used at the end of each part program to return the spindle and the machine table back to their respective machine home positions. This leaves the control ready to rerun the same part program or run a new program. The following two command statements are used for this purpose:

```
G91 G28 Z0.0
G91 G28 X0.0 Y0.0
```

The reason for using two separate reference point return statements, one for the Z direction and the other for X and Y directions, is to clear the tool in the Z direction from the workpiece before other movements are commanded. The tool movement pattern for these two command statements is shown in Figure 9.31.

**9.6.2 Return from Reference Point (G29)**

The G29 code directs the control to move the tool in rapid-traverse mode from the machine origin (or reference point) to the intermediate point and then continue to the commanded position (Figure 9.32). The G29 code does not define its own intermediate point; instead, it uses the intermediate point previously defined by the G28 code. Thus, G29 is used after the execution of G28. The command format for G29 is

G29 Xx Yy Zz

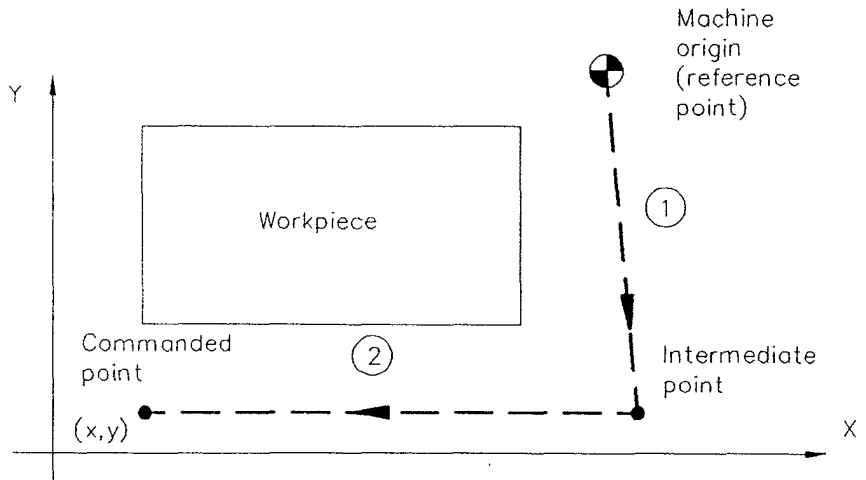
where x, y, and z are the coordinates of the commanded end position.

The G29 code is often used to position the tool in a convenient location for starting to machine in one block of instruction after the tool change is executed. This results in reducing the programming effort and shortening the program size. Figure 9.33 illustrates a typical application of this type, a partial program for which follows:

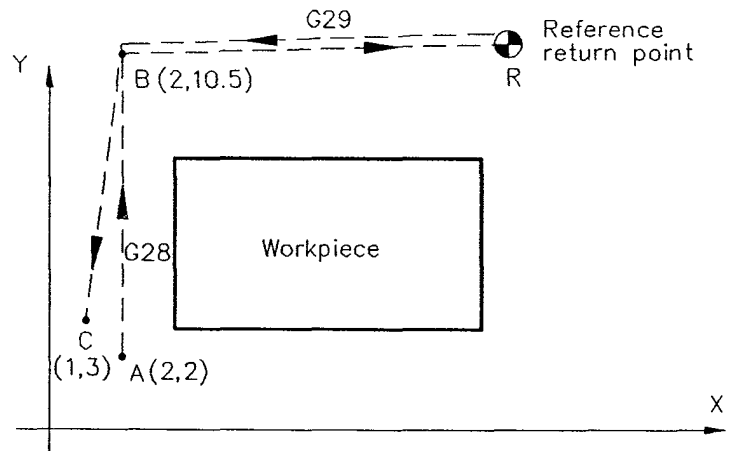
```

:           tool motion statements
N45 G1 X2.0 Y2.0      feed to A
N50 G90 G28 X2.0 Y10.5  reference point return
N60 T2 M6            tool change
    
```





**FIGURE 9.32**  
The working sequence of the G29 command.



**FIGURE 9.33**  
The use of G28 and G29 to go to and from a reference point.

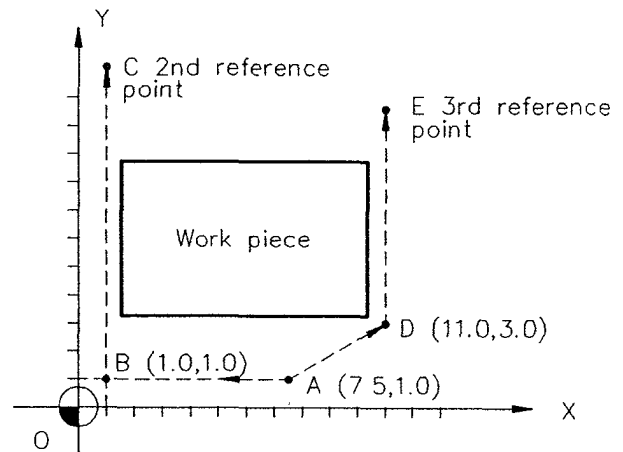
```
N70 G29 X1.0 Y3.0      return from reference point
:                    tool motion statements
:
```

After part machining for one tool is completed, the tool moves to point A(2,2). Then the G28 code in N50 directs the tool to move to point B(2,10.5), the intermediate point, and continue to the machine origin. N60 executes the tool change. Then the G29 code in N70 instructs the control to move the tool from the reference point (machine origin) to the intermediate point B(2,10.5), specified in the G28 statement, and proceed to the commanded position C(1,3).

### 9.6.3 Second, Third, and Fourth Reference Point Return (G30)

Many CNC controls provide multiple reference points, with four being the most common, to which the tool may be returned. The first reference point is usually the machine origin, which is often used for the tool change position. It is a fixed point and cannot be altered easily. The locations for the second, third, and fourth reference points are set with parameters. The G28 code is used for the first reference point return and the G30 code

**FIGURE 9.34**  
G30 programming example.



for the second, third, and fourth reference points. The G30 command has the following format:

`G30 Pp Xx Yy Zz`

where  $p$  specifies the number of the reference point, either 2, 3, or 4 (P2, the default, can be omitted in the statement). The values of  $x$ ,  $y$ , and  $z$  define the intermediate point.

Figure 9.34 shows the tool motion patterns for the G30 command. Assume that the tool is currently positioned at A(7.5,1.0). The statement

```
G30 X1.0 Y1.0
```

directs the control to move the tool from A to the intermediate point B(1.0,1.0) and continue to second reference point, C. The statement

```
G30 P3 X11.0 Y3.0
```

moves the tool path from A to the specified intermediate point D(11.0,3.0) and then to the third reference point, E.

## 9.7 TOOL SELECTION AND CHANGE

Most part programs require more than one tool to machine a workpiece. When this happens, the part program has to tell the control which tool is being selected and give commands to execute the tool change procedure. Three factors have to be carefully considered in programming a tool change in part programs:

1. Most CNC milling types of machine tools require that the spindle be returned to a Z home position before any tool change action can be taken.
2. The T-function with a two-digit number selects the tool but does not actually perform the tool change. The T-function must therefore be used in conjunction with the tool change command, M06.
3. Tool length may vary from one tool to another. This necessitates the use of a tool length offset, which is discussed in Chapter 10.

## 9.8 FEED SELECTION AND INPUT

The feed indicates the traveling speed at which the tool passes through the workpiece. Conventionally, feed is expressed in inches per minute (IPM), inches per revolution (IPR), or inches per tooth (IPT). In the metric system, feed is specified in millimeters per minute, revolution, or tooth. Although either IPM or IPR can be used as the feed unit, IPM is normally used in milling operations and IPR for turning types of machinings. Two preparatory codes used for specifying the feed unit are G94 (feed per minute) and G95 (feed per revolution). They are used with G70 and G71 as follows:

G70 G94	inches per minute
G71 G94	millimeters per minute
G70 G95	inches per revolution
G71 G95	millimeters per revolution

The G94 code is the default setting. The actual feed rate is given by the F-function. Some examples are

G94 G70 F7.5	7.5 IPM feed
G95 G70 F0.02	0.02 IPR feed
G94 G71 F150.0	150 MMPM feed
G95 G71 F1.5	1.5 MMPR feed

A feed must be given for linear and circular interpolation commands. The feed function is of the modal type; once specified, it remains in effect until overridden by another feed function.

---

## 9.9 SPINDLE SPEED SELECTION AND CONTROL

The S-function specifies the rotational speed of the machine spindle. The statement S3000, for example, instructs the control to regulate the spindle rotation at 3000 RPM. The S-function alone does not turn the spindle on, however; it must be used in conjunction with the spindle control miscellaneous functions: M03 (spindle on clockwise), M04 (spindle on counterclockwise), and M05 (spindle off). Here are some of the spindle speed selection and control examples:

S1000 M03	turn on spindle clockwise at 1000 RPM
S1500 M04	turn on spindle counterclockwise at 1500 RPM
M05	turn spindle off

---

## 9.10 MISCELLANEOUS FUNCTIONS

Three groups of miscellaneous functions are often used in part programs but have not yet been discussed: the program stop function, the end-of-program function, and the coolant control function.

### 9.10.1 Program Stop Function

Two program stop functions are available: unconditional stop (M00) and conditional (or optional) stop (M01). When the program stop function (M00) is encountered, the control halts all machine operations, including spindle off, feed off, and coolant off, until the restart signal is given by depressing the automatic start button on the control panel. Some controls stop the feed but not the spindle or coolant upon receiving M00.

The optional stop function, M01, has the same operation as M00, but it is effective only when the optional stop switch is set to "on." The optional stop switch is of toggle type and can be switched to "on" or "off." When the optional stop switch is set to "off," the M01 function is ignored.

Program stop functions are often inserted into a program when dimensional measurement and inspection of the workpiece and tool are to be performed.

### 9.10.2 End-of-Program Functions

The NC control requires a command to indicate where the end of the program is. The end-of-program command is placed in the last block of a part program. M02 and M30 are commonly used to indicate the end of program. The functions of these two commands are very similar, but their features vary from one control to another, so always refer to the instruction manual for proper use.

### 9.10.3 Coolant Control Function

The following three miscellaneous functions allow the control system to control the coolant condition:

M07	mist coolant on
M08	flood coolant on
M09	coolant off

---

## 9.11 PROGRAMMING EXAMPLE

This chapter has presented most of the essential concepts, principles, commands, and techniques for writing a simple part program for NC milling machines. Two programming examples (one in the inch system and the other in the metric system) will be used to consolidate these discussions and make them more practical for beginners.

---

### EXAMPLE 9.9 PART PROGRAMMING IN THE INCH SYSTEM

Consider the workpiece shown in Figure 9.35. The part origin is located (4.58, -5.75, -6.55) from the machine origin. The machining consists of two types of cutting: profile contouring and hole drilling. A 0.5 in. end mill is used for contouring and a 0.5 in. drill for hole drilling. The tool length of the end mill is 2.5 in., and the length of the drill is 3 in. The desired cutter path is indicated by the arrows. The part coordinates for 17 critical points are given in Table 9.3. Note that these coordinates cannot be used directly

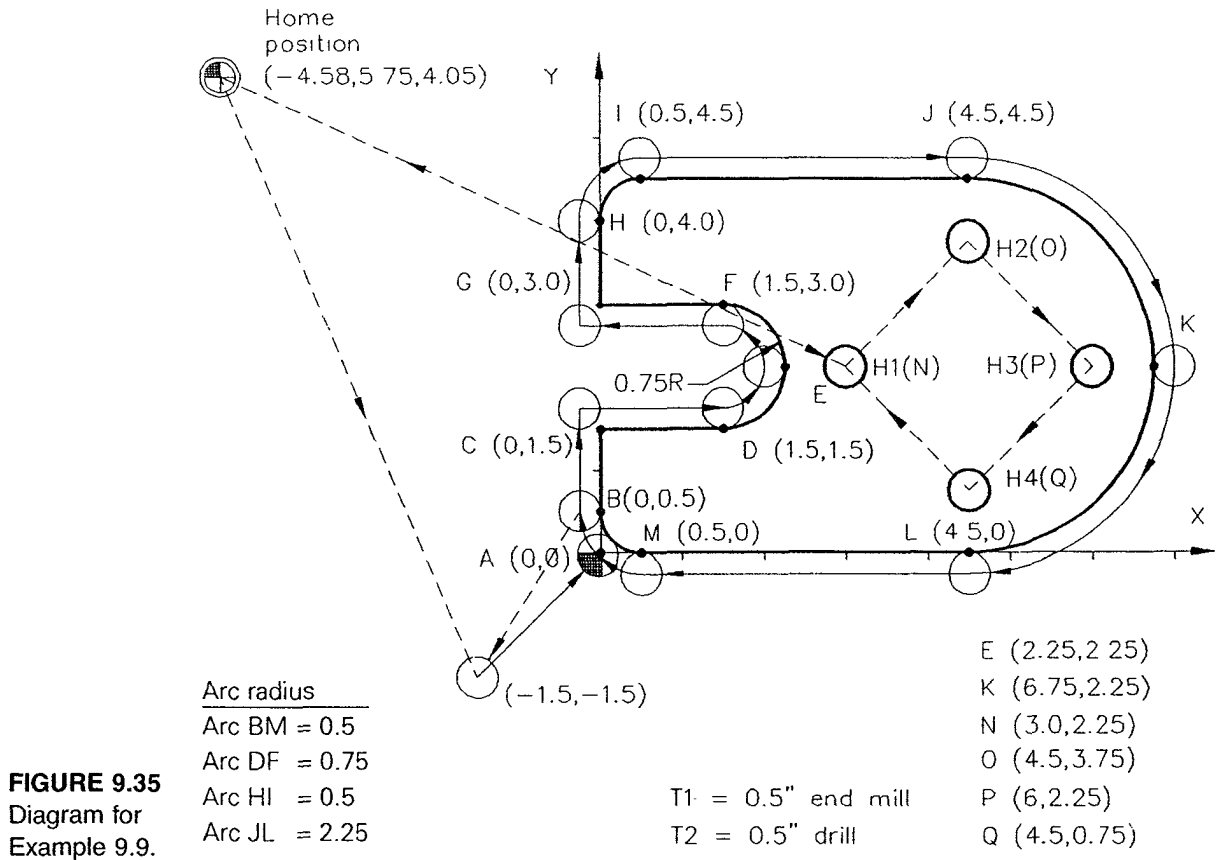


FIGURE 9.35 Diagram for Example 9.9.

TABLE 9.3 Part Coordinates for Example 9.9

Point	Part coordinates		Cutter center coordinates	
	X coordinate	Y coordinate	X coordinate	Y coordinate
A	0	0	-0.25	0
B	0	0.5	-0.25	0.5
C	0	1.5	-0.25	1.75
D	1.5	1.5	1.5	1.75
E	2.25	2.25	2.0	2.25
F	1.5	3.0	1.5	2.75
G	0	3.0	-0.25	2.75
H	0	4.0	-0.25	4.0
I	0.5	4.5	0.5	4.75
J	4.5	4.5	4.5	4.75
K	6.75	2.25	7.0	2.25
L	4.5	0	4.5	-0.25
M	0.5	0	0.5	-0.25
N	3.0	2.25	3.0	2.25
O	4.5	3.75	4.5	3.75
P	6.0	2.25	6.0	2.25
Q	4.5	0.75	4.5	0.75

for cutter center coordinates unless the cutter diameter compensation commands are used. In this example, we use the concept of "cutter center programming," in which the programmed coordinates are given on the basis of cutter center locations. To obtain the desired part profile, the cutter center has to be displaced by an amount of cutter radius from the part profile in the normal direction. The cutter center coordinates for each significant point are shown to the right of their part coordinate counterparts in Table 9.3. Note that the cutter centers at points N, O, P, and Q are not offset because of the drilling operation.

The complete part program for machining this workpiece, with a brief explanation for each command block, is as follows.

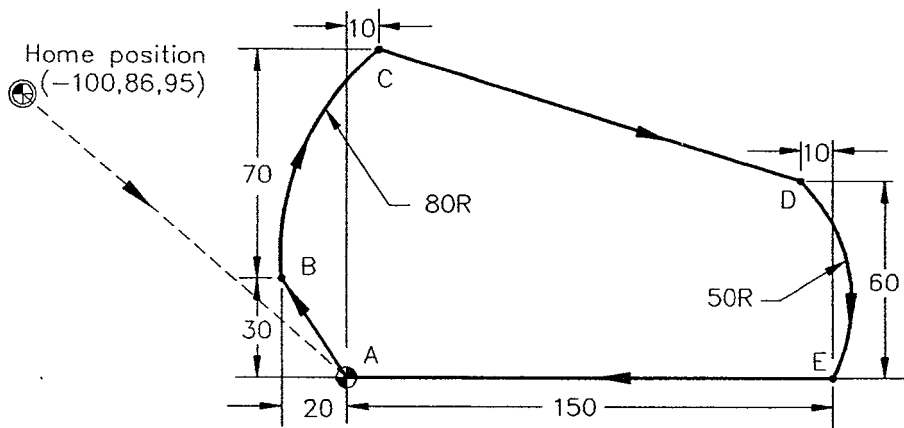
N5 G90 G80 G40 G17	safety feature
N10 T1 M6	tool change to T1
N15 G92 X-4.58 Y5.75 Z4.05	define part origin
N20 G0 X-1.5 Y-1.5 Z0.5 S1500 M3	move rapidly to initial point, spindle on (CW, 1500 RPM)
N25 G1 Z-0.75 F15.0	feed to desired depth
N30 X-0.25 Y0 F7.5	cut to point A
N35 Y1.75	cut to point C
N40 X1.5	cut to point D
N45 G3 X2.0 Y2.25 I0 J0.5 F5.0	cut arc DE to point E
N50 X1.5 Y2.75 I-0.5 J0	cut arc EF to point F
N55 G1 X-0.25 F7.5	cut to point G
N60 Y4.0	cut to point H
N65 G2 X0.5 Y4.75 I.75 J0 F5.0	cut arc HI to point I
N70 G1 X4.5 F7.5	cut to point J
N75 G2 X7.0 Y2.25 I0 J-2.5	cut arc JK to point K
N80 X4.5 Y-0.25 I-2.5 J0	cut arc KL to point L
N85 G1 X0.5	cut to point M
N90 G2 X-0.25 Y0.5 I0 J0.75 F5.0	cut arc MB to point B
N95 G1 X-1.5 Y-1.5 F15.0	return to initial point
N100 G92 X3.08 Y-7.25 Z-4.8	return to machine coordinate system
N105 G91 G28 Z0 M5	machine origin return (Z direction), spindle off
N110 G91 G28 X0 Y0	machine origin return (X and Y directions)
N115 G90	absolute mode
N120 T2 M6	tool change to T2
N125 G92 X-4.58 Y5.75 Z3.55	set work coordinate for T2
N130 S1000 M3	spindle on at 1000 RPM
N135 G0 X3.0 Y2.25 Z0.5	move rapidly to center of H1
N140 Z0.2	lower Z axis to Z0.2
N145 G1 Z-0.8 F5.0	drill H1 to Z-0.8
N150 G0 Z0.2	retract rapidly to Z0.2
N155 X4.5 Y3.75	move rapidly to center of H2
N160 G1 Z-0.8	drill H2 to Z-0.8
N165 G0 Z0.2	retract rapidly to Z0.2
N170 X6.0 Y2.25	move rapidly to center of H3
N175 G1 Z-0.8	drill H3 to Z-0.8
N180 G0 Z0.2	retract rapidly to Z0.2
N185 X4.5 Y0.75	move rapidly to center of H4
N190 G1 Z-0.8	drill H4 to Z-0.8
N195 G0 Z0.2	retract rapidly to Z0.2

N200 X0 Y0 M5	move rapidly to work origin; turn spindle off
N205 G91 G28 Z0	machine origin return (Z direction)
N210 G91 G28 X0 Y0	machine origin return (X and Y directions)
N215 G92 X0 Y0 Z0	reset machine coordinate
N220 M30	end of program

## EXAMPLE 9.10 PART PROGRAMMING IN THE METRIC SYSTEM

The following part program takes a 0.25 in. end mill along the contour indicated in Figure 9.36. To simplify coordinate calculations, no coordinate offset for significant points is required in this example.

N5 G90 G71	absolute positioning, metric unit
N10 T1 M6	tool change to T1
N15 G92 X-100.0 Y86.0 Z95.0	define work zero point at A
N20 G0 X0 Y0 S2500 M3	rapid traverse to A, spindle on (2500 RPM, CW)
N25 Z12.5	rapid plunge to 12.5 mm above Z0
N30 G1 Z-12.5 F150.0	feed to Z-12.5, feed rate 150 MPPM
N35 X-20.0 Y30.0	cut line AB to B
N40 G2 X10.0 Y100.0 R80.0	cut arc BC to C
N45 G1 X140.0 Y60.0	cut line CD to D
N50 G2 X150.0 Y0 R50.0	cut arc DE to E
N55 G1 X0 Y0	cut line EA to A
N60 G0 Z12.5	rapid retract to Z12.5
N65 G91 G28 Z0 M5	reference point return in Z direction, spindle off
N70 G91 G28 X0 Y0	reference point return in X and Y directions
N75 M30	end of program



**FIGURE 9.36**  
Diagram for Example 9.10.

## REVIEW QUESTIONS

1. Identify the five categories of programming command codes and techniques.
2. List and describe the three types of tool motion commands.
3. Illustrate and explain the two motion patterns of rapid-traverse code, G00.
4. State the command format of G00.
5. List and describe the command format of the linear interpolation code, G01.
6. What are the modal commands or functions? What special properties do they have?
7. Identify the five pieces of information required to execute a circular interpolation.
8. What are the two methods for programming arc centers?
9. Illustrate and describe how to determine the vector components I, J, and K of the arc center vector.
10. Find the I and J values for an arc whose start point is located at (5.414,2.586) and whose center point is at (4,4).
11. What are the two methods for plane selection?
12. Describe the following three G-codes:
  - a. G17
  - b. G18
  - c. G19
13. What is axis-address priority plane selection?
14. Describe the two positioning systems: absolute positioning and incremental positioning.
15. What are the preparatory codes for inch system and metric system selection, respectively?
16. Illustrate and describe the function of the absolute zero setting command, G92.
17. Identify two typical applications of the G92 code.
18. What are the reference points, and what are they used for?
19. Describe and illustrate the tool motion sequence of reference point return command G28.
20. Explain how CNC controls use T-functions and the M6 command to select and change tools.
21. Explain the following feed rate command statements:
  - a. G70 G94 F15.0
  - b. G70 G95 F.005
  - c. G71 G94 F200.0
  - d. G71 G95 F.15
22. Explain the following spindle speed control command statements:
  - a. S2000 M3
  - b. S2500 M4
  - c. M5
23. Describe the difference between two program stop functions M00 and M01.
24. Depict the following three coolant control functions:
  - a. M07
  - b. M08
  - c. M09





25. Write a part program to machine three part features—one pocket, one angle strip, and one flat surface—of the part shown in Figure 9.37, using the inch system. Use a 1 in. end mill to cut the angle strip and the flat surface, and a 0.75 in. end mill to produce the pocket. Note that the tool path for the flat surface must be carefully planned to cover the whole surface.
26. Write a part program (metric system) to take a 25 mm end mill along the part profile shown in Figure 9.38 in the clockwise direction. The machine home position is located at  $(-100, 130, 80)$  from the selected program zero point.