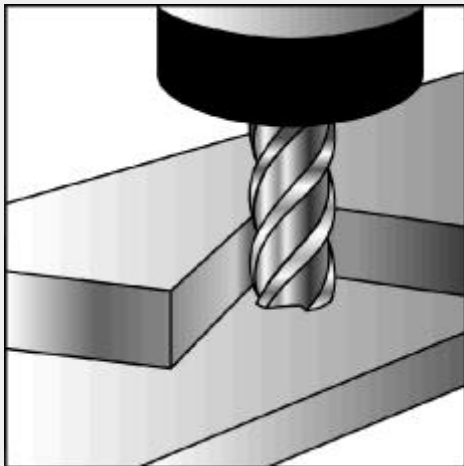




Allen-Bradley

9/Series CNC
Mill



Operation and Programming Manual

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Reproduction of the contents of this copyrighted publication, in whole or in part, without written permission of Allen-Bradley Company, Inc. is prohibited.

Throughout this manual we make notes to alert you to possible injury to people or damage to equipment under specific circumstances.



WARNING: Tells readers where people may be hurt if procedures are not followed properly.



CAUTION: Tells readers where machinery may be damaged or economic loss can occur if procedures are not followed properly.

Warnings and Cautions:

- identify a possible trouble spot
- tell what causes the trouble
- give the result of improper action
- tell the reader how to avoid trouble

Important: We recommend that you frequently back up your application programs on an appropriate storage medium to avoid possible data loss.

PLC is a registered trademark of Allen-Bradley Company, Inc.

Paramacro and PAL are trademarks of Allen-Bradley Company, Inc.

Summary of Changes

New Information

The following is a list of the larger changes made to this manual since its last printing. Other less significant changes were also made throughout.

- Error Message Log
- Paramacro Parameters
- Softkey Tree
- Error Messages

Revision Bars

We use revision bars to call your attention to new or revised information. A revision bar appears as a thick black line on the outside edge of the page as indicated here.



Chapter 1 Using This Manual

1.0 Chapter Overview	1-1
1.1 Audience	1-1
1.2 Manual Design	1-1
1.3 Reading this Manual	1-3
1.4 Terms and Conventions	1-4
1.5 Warnings, Cautions, and Important Information	1-5
1.6 Related Publications	1-5

Chapter 2 Basic Control Operation

2.0 Chapter Overview	2-1
2.1 Operator Panel Operations	2-1
2.1.1 Keyboard	2-3
2.1.2 Calculator Function	2-4
2.1.3 Softkeys	2-8
2.1.4 CRT	2-10
2.1.5 Portable Operator Panel	2-11
2.2 The MTB Panel	2-12
2.3 Software MTB Panel {FRONT PANEL}	2-15
2.4 Power Procedures	2-21
2.4.1 Turning Power On	2-21
2.4.2 Turning Power OFF	2-22
2.5 Control Conditions at Power-Up	2-23
2.6 Emergency Stop Operations	2-24
2.6.1 Emergency Stop Reset	2-24
2.7 Access Control	2-25
2.7.1 Assigning Access Levels and Passwords	2-26
2.7.2 Password Protectable Functions	2-29
2.7.3 Entering Passwords	2-31
2.8 Changing Operating Modes	2-33
2.9 Displaying System and Machine Messages	2-37
2.9.1 Clearing Active Messages {CLEAR ACTIVE}	2-40
2.10 The Input Cursor	2-41
2.11 {REFORM MEMORY}	2-41
2.12 Removing an Axis (Axis Detach)	2-43
2.13 Time Parts Count Display Feature	2-43

Chapter 3 Offset Tables and Setup

3.0 Chapter Overview	3-1
3.1 Tool Offset Table {TOOL GEOMET} and {TOOL WEAR}	3-1
3.1.1 Tool Offset Dimensional Parameters	3-2

3.1.2 Setting Tool Offset Tables	3-5
3.1.3 Setting Offset Data Using {MEASURE}	3-9
3.1.4 Tool Offset Range Verification	3-10
3.2 Changing the Active Tool Offset {ACTIVE OFFSET}	3-12
3.3 Work Coordinate System Offset Tables {WORK CO-ORD}	3-14
3.3.1 Setting Work Coordinate System Tables	3-15
3.4 Backing Up Offset Tables	3-17
3.5 Programmable Zone Table	3-21
3.6 Single Digit Feedrate Table	3-23

Chapter 4 Manual/MDI Operation Modes

4.0 Chapter Overview	4-1
4.1 Manual Operating Mode	4-1
4.1.1 Jogging an Axis	4-2
4.1.2 Continuous Jog	4-3
4.1.3 Incremental Jog	4-3
4.1.4 HPG Jog	4-4
4.1.5 Arbitrary Angle Jog	4-5
4.1.6 Jog Offset	4-6
4.1.7 Resetting Overtravels	4-6
4.2 Mechanical Handle Feed (Servo Off)	4-8
4.3 Removing an Axis (Axis Detach)	4-8
4.4 Manual Machine Homing	4-8
4.5 MDI Mode	4-11
4.5.1 MDI Basic Operation	4-12

Chapter 5 Editing Programs Online

5.0 Chapter Overview	5-1
5.1 Selecting the Program To Edit	5-2
5.2 Editing Programs at the Control (Online)	5-3
5.2.1 Moving the Cursor {STRING SEARCH} and Cursor Size {CHAR/WORD}	5-5
5.2.2 Entering Characters and Blocks	5-7
5.2.3 Changing and Inserting {MODIFY INSERT}	5-7
5.2.4 Erasing Characters and Blocks	5-11
5.2.5 Sequence Numbers {RENUM PROGRAM}	5-13
5.2.6 Merging Part Programs {MERGE PROGRAM}	5-15
5.2.7 Exiting Edit Mode {EXIT EDITOR}	5-16
5.3 Programming Aids {QUICK VIEW}	5-17
5.3.1 Selecting a QuickView Plane	5-19
5.3.2 Using {QPATH+ PROMPT} Sample Patterns	5-20
5.3.3 G-code Format Prompting {GCODE PROMPT}	5-24
5.3.4 Mill Cycle Format Prompting	5-26

5.4 Digitizing a Program (Teach)	5-28
5.4.1 Linear Digitizing	5-30
5.4.2 Digitizing an Arc (3 Points)	5-32
5.4.3 Digitizing an Arc Tangent at End Points	5-34
5.5 Deleting Program {DELETE PRGRAM}	5-37
5.6 Renaming Programs {RENAME PRGRAM}	5-38
5.7 Displaying a Program {DISPLY PRGRAM}	5-39
5.8 Comment Display {PRGRAM COMENT}	5-40
5.9 Copying Programs {COPY PRGRAM}	5-41
5.10 Selecting the Protectable Part Program Directory	5-43
5.10.1 Protected Program Encryption and Decryption	5-46
5.10.2 Storing Encryption/Decryption Table to Backup Memory	5-49

Chapter 6 Editing Part Programs Offline (ODS)

6.0 Chapter Overview	6-1
6.1 Selecting the Part Program Application	6-2
6.2 Editing Part Programs Off Line	6-2
6.3 Interfacing the Workstation with the Control	6-5
6.4 Downloading Part Programs from ODS	6-5
6.5 Upload Part Programs to ODS	6-12

Chapter 7 Running a Program

7.0 Chapter Overview	7-1
7.1 Selecting Special Running Conditions	7-1
7.1.1 Block Delete	7-1
7.1.2 Miscellaneous Function Lock	7-2
7.1.3 Sequence Stop {SEQ STOP}	7-2
7.1.4 Single Block	7-4
7.2 Selecting a Part Program Input Device	7-5
7.3 Selecting a Program	7-6
7.4 Deselecting a Part Program	7-9
7.5 Program Search {SEARCH}	7-10
7.6 Search With Recall {MID ST PRGRAM}	7-13
7.7 Basic Program Execution	7-17
7.7.1 {QUICK CHECK}	7-19
7.7.2 Axis Inhibit Mode	7-20
7.7.3 Dry Run Mode	7-21
7.7.4 Part Production/Automatic Mode	7-23
7.8 Interrupted Program Recover {RESTRT PRGRAM}	7-25
7.9 Jog Retract	7-28
7.10 Block Retrace	7-31

Chapter 8 Display and Graphics

8.0 Chapter Overview	8-1
8.1 Selection of Axis Position Data Display	8-1
8.2 PAL Display Page	8-22
8.3 Changing Languages	8-23
8.4 Graphics	8-24
8.4.1 Selecting the Program for Graphics	8-24
8.4.2 Running Graphics	8-25
8.4.3 Disabling Graphics	8-27
8.4.4 Changing Parameters	8-27
8.4.5 Graphics in Single-Block	8-33
8.4.6 Clearing Graphics Screen	8-33
8.4.7 Displaying Machine Information in Graphics	8-33
8.4.8 Zooming Graphics	8-33
8.6 Power Turn-on Screen	8-37
8.7 Screen Saver	8-39

Chapter 9 Communications

9.0 Chapter Overview	9-1
9.1 Setting Communications	9-1
9.1.1 Setting Communication Port Parameter Values	9-1
9.1.2 Communication Port Parameters	9-3
9.2 Inputting Part Programs from a Tape Reader	9-9
9.3 Outputting Part Programs to a Tape Punch	9-13
9.4 Verifying Part Programs Against Source Programs	9-17
9.5 Error Conditions (Inputting and Outputting Part Programs)	9-18

Chapter 10 Introduction to Programming

10.0 Chapter Overview	10-1
10.1 Tape Format	10-1
10.2 Program Configuration	10-6
10.2.1 Program Names	10-8
10.2.2 Sequence Numbers	10-9
10.2.3 Comment Blocks	10-10
10.2.4 Block Delete and Multi Level Delete	10-10
10.2.5 End of Block Statement	10-11
10.3 Using Subprograms	10-12
10.3.1 Subprogram Call (M98)	10-13
10.3.2 Main and Subprogram Return (M99)	10-14
10.3.3 Subprogram Nesting	10-16
10.4 Word Formats and Functions	10-17

10.4.1 Minimum and Maximum Axis Motion (Programming Resolution)	10-21
10.5 Word Descriptions	10-22
10.5.1 A_ L_ ,R_ ,C_ (Quick Plus and Radius-Chamfer Words)	10-22
10.5.2 Axis Names	10-22
10.5.3 D -and H-Words (Tool Offsets)	10-22
10.5.4 F-words (Feedrate)	10-23
10.5.5 G-codes (Preparatory Functions)	10-24
10.5.6 I ,J, and K Integrand Words	10-30
10.5.7 M -codes(Miscellaneous Functions)	10-30
10.5.7.1 Auxiliary Miscellaneous Function (B-word)	10-37
10.5.8 N-words (Sequence Numbers)	10-37
10.5.9 O-words (Program Names)	10-37
10.5.10 P ,L (Main Program Jumps and Subprogram Calls)	10-37
10.5.11 S-word (Spindle Speed)	10-38
10.5.12 T-words (Tool Selection)	10-40

Chapter 11 Coordinate Systems Offsets

11.0 Chapter Overview	11-1
11.1 Machine Coordinate System (Absolute)	11-1
11.1.1 Motion in the Machine Coordinate System (G53)	11-2
11.2 Preset Work Coordinate Systems (G54-59.3)	11-4
11.2.1 Altering Work Coordinate Systems (G10L2)	11-7
11.3 Work Coordinate System External Offset	11-9
11.3.1 Altering External Offset (G10L2)	11-11
11.4 Offsetting the Work Coordinate Systems	11-13
11.4.1 Coordinate Offset Using Tool Position (G92)	11-13
11.4.2 Offsetting Coordinate Zero Points (G52)	11-16
11.4.3 {SET ZERO} Offset	11-18
11.4.4 Jogging an Offset	11-19
11.4.5 Canceling Coordinate System Offsets (G92.1)	11-20
11.4.6 Canceling Selected Coordinate System Offsets (G92.2)	11-21
11.5 PAL Offsets	11-22

Chapter 12 Overtravels and Programmable Zones

12.0 Chapter Overview	12-1
12.1 Hardware Overtravels	12-2
12.2 Software Overtravels	12-3
12.3 Programmable Zone 2 (G22, G23)	12-5
12.4 Programmable Zone 3 (G22.1, G23.1)	12-7
12.5 Resetting Overtravels	12-13

Chapter 13 Coordinate Control

13.0 Chapter Overview	13-1
13.1 Rotating the Coordinate Systems	13-1
13.1.1 Rotating the Current Work Coordinate System (G68, G69)	13-2
13.1.2 External Part Rotation	13-6
13.2 Plane Selection (G17, G18, G19)	13-11
13.3 Absolute/Incremental Modes (G90, G91)	13-12
13.4 Inch/Metric Modes (G20, G21)	13-13
13.5 Scaling	13-14
13.5.1 Scaling and Axis Position Display Screens	13-17
13.5.2 Scaling Magnification Data Screen	13-17
13.5.3 Scaling Restrictions	13-19

Chapter 14 Axis Motion

14.0 Chapter Overview	14-1
14.1 Positioning Axes	14-1
14.1.1 Rapid Positioning Mode (G00)	14-1
14.1.2 Linear Interpolation Mode (G01)	14-3
14.1.3 Circular Interpolation Mode (G02, G03)	14-5
14.1.4 Helical Interpolation Mode (G02, G03)	14-10
14.1.5 Positioning Rotary Axes	14-12
14.1.6 Cylindrical Interpolation	14-14
14.1.7 PAL Axis Mover	14-20
14.2 Polar Coordinate Programming (G15, G16)	14-21
14.2.1 Polar Programming Special Cases	14-25
14.3 Automatic Motion To and From Machine Home	14-29
14.3.1 Automatic Machine Homing (G28)	14-29
14.3.2 Automatic Return to Machine Home (G28)	14-30
14.3.3 Automatic Return From Machine Home (G29)	14-32
14.3.4 Machine Home Return Check (G27)	14-33
14.3.5 Return to Alternate Home (G30)	14-34
14.4 Dwell (G04)	14-35
14.4.1 Dwell - Seconds	14-36
14.4.2 Dwell - Number of Spindle Revolutions	14-36
14.5 Programmable Mirror Image (G50.1 - G51.5)	14-36
14.5.1 Manual Mirror Image	14-38
14.6 Axis Clamp	14-39
14.7 Feed to Hard Stop (G24)	14-40

Chapter 15

Using QuickPath Plus†

15.0 Chapter Overview	15-1
15.1 Using QuickPath Plus	15-1
15.2 Linear QuickPath Plus	15-3
15.3 Circular QuickPath Plus (G13, G13.1)	15-7

Chapter 16

Using Chamfers and Corner Radius

16.0 Chapter Overview	16-1
16.1 Chamfers and Corner Radius	16-1

Chapter 17

Spindles

17.0 Chapter Overview	17-1
17.1 Controlling Spindle (G12.1, G12.2, G12.3)	17-1
17.2 Spindle Orientation (M19)	17-3
17.3 Spindle Direction (M03, M04, M05)	17-5
17.4 Synchronized Spindles	17-6
17.4.1 Using the Spindle Synchronization Feature	17-7
17.5 Special Considerations for Spindle Synchronization	17-9

Chapter 18

Programming Feedrates

18.0 Chapter Overview	18-1
18.1 Feedrates	18-1
18.1.1 Feedrates Applied During Cutter Compensation	18-2
18.1.2 Inverse Time Feed Mode (G93)	18-4
18.1.3 Feed-Per-Minute Mode (G94)	18-5
18.1.4 Feed-Per-Revolution Mode (G95)	18-5
18.1.5 Rapid Feedrate	18-6
18.1.6 Feedrate Overrides	18-7
18.1.7 Feedrate Limits (Clamp)	18-8
18.2 Feedrates to Control Torque Adaptive Feed (G25)	18-9
18.3 Special AMP Assigned Feedrates	18-12
18.3.1 Single Digit F-words	18-12
18.3.2 External Feedrate Switch	18-13
18.4 Automatic Acceleration/Deceleration (Acc/Dec)	18-14
18.4.1 Exponential Acc/Dec	18-15
18.4.2 Linear Acc/Dec	18-16
18.4.3 S-Curve Acc/Dec	18-17
18.4.4 Programmable Acc/Dec	18-18
18.4.5 Precautions on Corner Cutting	18-20
18.4.6 Spindle Acceleration (Ramp)	18-22

18.4.7 Short Block Acc/Dec G36, G36.1	18-22
---	-------

Chapter 19 Dual-axis Operation

19.0 Chapter Overview	19-1
19.1 Dual-axis Operation	19-1
19.1.1 Parking a Dual Axis	19-3
19.1.2 Homing a Dual Axis	19-4
19.1.3 Programming a Dual Axis	19-5
19.1.4 Offset Management for a Dual Axis	19-7

Chapter 20 Tool Control Functions

20.0 Chapter Overview	20-1
20.1 Programming a T-word	20-1
20.2 Tool Length Offset Function (G43, G44, G49)	20-3
20.2.1 Activating Tool Length Offsets	20-8
20.2.2 Tool Length Offset (TLO) Axis Selection (G43.1, G44.1)	20-9
20.3 Random Tool	20-11
20.4 Programming Alterations of the Offset Tables (G10L10 - G10L13)	20-18
20.5 Automatic Tool Life Management	20-19
20.5.1 Tool Directory Data	20-20
20.5.2 Assigning Detailed Tool Data	20-25
20.5.3 Programming Data and Backing Up Tool Management Tables (G10L3, G11)	20-29
20.5.4 Programming Using Tool Management	20-33

Chapter 21 Cutter Diameter Compensation (G40, G41, G42)

21.0 Chapter Overview	21-1
21.1 Active Cutter Compensation	21-3
21.2 Cutter Compensation Generated Blocks G39, G39.1	21-7
21.3 Cutter Compensation (Type A)	21-10
21.3.1 Cutter Compensation Type A Entry Moves	21-10
21.3.2 Cutter Compensation Type A Exit Moves	21-14
21.4 Cutter Compensation (Type B)	21-20
21.4.1 Cutter Compensation Type B Entry Moves	21-20
21.4.2 Cutter Compensation Type B Exit Moves	21-24
21.5 Tool Path During Cutter Compensation	21-30
21.6 Cutter Compensation Special Cases	21-35
21.6.1 Changing Cutter Compensation Direction	21-35
21.6.2 Too Many Non-Motion Blocks	21-39
21.6.3 Corner Movement After Generated Blocks	21-41
21.6.4 Changing Cutter Radius During Compensation	21-43
21.6.5 MDI or Manual Motion During Cutter Compensation	21-46

21.6.6 Moving To/From Machine Home	21-48
21.6.7 Changing or Offsetting Work Coordinate System	21-49
21.6.8 Block Look-Ahead	21-50
21.7 Error Detection for Cutter Compensation	21-51

Chapter 22 Using Pocket Milling Cycles

22.0 Chapter Overview	22-1
22.1 Pocket Milling Roughing Cycle (G88.1)	22-1
22.1.1 Rectangular Pocket Roughing Using G88.1	22-2
22.1.2 Rectangular Pocket Enlarging Using G88.1	22-5
22.1.3 Slot Roughing Using G88.1	22-8
22.1.4 Circular Pocket Roughing Using G88.1	22-10
22.1.5 Circular Pocket Enlarging Using G88.1	22-13
22.2 Pocket Milling Finishing Cycle (G88.2)	22-15
22.2.1 Rectangular Pocket Finishing Using G88.2	22-16
22.2.2 Circular Pocket Finishing Using G88.2	22-19
22.2.3 Slot Finishing Using G88.2	22-20

Chapter 23 Using Post Milling Cycles

23.0 Chapter Overview	23-1
23.1 Post Milling Roughing Cycle (G88.3)	23-1
23.1.1 Rectangular Post Roughing Using G88.3	23-2
23.1.2 Circular Post Roughing Using G88.3	23-5
23.2 Post Milling Finishing Cycle (G88.4)	23-7
23.2.1 Rectangular Post Finishing Using G88.4	23-8
23.2.2 Circular Post Finishing Using G88.4	23-11

Chapter 24 Using Hemisphere Milling Cycles

24.0 Chapter Overview	24-1
24.1 Hemisphere Milling Roughing Cycle (G88.5)	24-1
24.1.1 Concave Hemisphere Roughing Using G88.5	24-2
24.1.2 Convex Hemisphere Roughing Using G88.5	24-5
24.2 Hemisphere Milling Finishing Cycle (G88.6)	24-7
24.2.1 Concave Hemisphere Finishing Using G88.6	24-8
24.2.2 Convex Hemisphere Finishing Using G88.6	24-10

Chapter 25 Irregular Pocket Milling Cycles

25.0 Chapter Overview	25-1
25.1 Irregular Pocket Milling	25-1
25.1.1 Irregular Pocket Roughing (G89.1)	25-2

25.1.2 Irregular Pocket Finishing (G89.2)	25-10
---	-------

Chapter 26 Milling Fixed Cycles

26.0 Chapter Overview	26-1
26.1 Milling Fixed Cycles	26-2
26.2 Positioning and Hole Machining Axes	26-4
26.3 Parameters	26-7
26.4 Milling Fixed Cycle Operations	26-8
(G73): Deep Hole Peck Drilling Cycle with Dwell	26-9
(G74): Left-Hand Tapping Cycle	26-10
(G74.1): Left-Hand Solid-Tapping Cycle	26-12
(G76): Boring Cycle, Spindle Shift	26-15
(G80): Cancel or End Fixed Cycles	26-18
(G81): Drilling Cycle, No Dwell/Rapid Out	26-18
(G82): Drill Cycle, Dwell/Rapid Out	26-20
(G83): Deep Hole Drilling Cycle	26-21
(G84): Right-Hand Tapping Cycle	26-23
(G84.1): Right-Hand Solid-Tapping Cycle	26-25
(G85): Boring Cycle, No Dwell/Feed Out	26-28
(G86): Boring Cycle, Spindle Stop/Rapid Out	26-30
(G87): Back Boring Cycle	26-32
(G88): Boring Cycle, Spindle Stop/Manual Out	26-34
(G89): Boring Cycle, Dwell/Feed Out	26-36
26.5 Altering Milling Fixed Cycle Operating Parameters	26-38
26.6 Examples of Drilling Cycles	26-40

Chapter 27 Skip, Gauge, and Probing Cycles

27.1 Chapter Overview	27-1
27.2 External Skip, Gauge, and Probe Functions	27-1
27.2 External Skip Functions (G31 codes)	27-2
27.3 Tool Gauging External Skip Functions (G37 codes)	27-4
27.4 Hole Probing (G38)	27-8
27.5 Parallel Probing Cycle (G38.1)	27-12
27.6 Probing Parameters Table	27-15
27.7 Adaptive Depth (G26)	27-18

Chapter 28 Paramacros

28.0 Chapter Overview	28-1
28.1 Paramacros	28-1
28.2 Parametric Expressions	28-2
28.2.1 Basic Mathematical Operators	28-2
28.2.2 Mathematical Function Commands	28-4
28.2.3 Parametric Expressions as G- or M-Codes	28-6
28.3 Transfer of Control Commands	28-7
28.3.1 Conditional Operators	28-7
28.3.2 GOTO and IF-GOTO Commands	28-8
28.3.3 DO-END and WHILE-DO-END Commands	28-10
28.4 Parameter Assignments	28-12
28.4.1 Local Parameter Assignments	28-12
28.4.2 Common Parameters	28-15
28.4.3 System Parameters	28-15
28.4.4 PAL Parameters	28-33
28.4.5 Shared Dual-Process Parameters (#7100 - 7199)	28-35
28.5 Assigning Parameter Values	28-36
28.6 Macro Call Commands	28-44
28.6.1 Non-Modal Paramacro Call (G65)	28-45
28.6.2 Modal Paramacro Call (G66)	28-46
28.6.3 Modal Paramacro Call (G66.1)	28-48
28.6.4 AMP-defined G-Code Macro Call	28-50
28.6.5 AMP-Defined M-Code Macro Call	28-51
28.6.6 AMP-Defined T-, S-, and B-Code Macro Call	28-51
28.6.7 Nesting Macros	28-52
28.7 Macro Output Commands	28-54

Chapter 29 Program Interrupt

29.0 Chapter Overview	29-1
29.1 Enabling and Disabling Interrupts (M96/M97)	29-1
29.2 Interrupt Request Considerations	29-4
29.3 Interrupt Types	29-5
29.4 The Interrupt Program	29-8

Chapter 30 Using a 9/Series Dual-processing System

30.0 Chapter Overview	30-1
30.1 Defining of a Dual- processing System	30-1
30.2 Operating a Dual-processing System	30-2
30.3 Synchronizing Multiple Part Programs	30-7
30.4 Spindle Control for Dual- processing Systems	30-12

30.5 Using Interference Checking with a Dual-process Mill	30-12
30.5.1 Measuring Interference Boundaries	30-16
30.5.2 Entering Interference Values Manually	30-19
30.5.3 Entering Interference Values through Programming (G10L5 and G10L6)	30-21
30.5.4 Backing Up Interference Tables	30-23
30.6 Shared Axes on Dual-processing Systems	30-26
30.6.1 Operating a Shared Axis	30-26
30.6.2 Switching a Shared Axis to a Different Process	30-28
30.6.3 Setting up a Shared Axis	30-29
30.7 Dual Axes on a Dual-processing System	30-31
30.7.1 Decoupling a Dual Axis Group	30-32
30.7.2 Independently Programming Dual Axis Members	30-33

Chapter 31 Using Transfer Line Cycles

31.0 Chapter Overview	31-1
31.1 Transfer Line Cycles	31-2
31.1.1 Using Transfer Line Cycles	31-3
31.1.2 Selecting the Program to Edit or Create	31-6
31.1.3 Creating a Transfer Line Part Program	31-9
31.1.4 Editing Part Programs	31-12
31.1.5 Reloading Part Program Templates	31-17
31.1.6 Running the Cycles	31-25
31.2 Understanding the QuickView Templates	31-25

Appendix A Softkey Tree

Appendix Overview	A-1
Understanding Softkeys	A-1
Describing Level 1 Softkeys	A-3
Using the Softkey Tree	A-3

Appendix B Error and System Messages

Overview	B-1
----------------	-----

Appendix C G-code Tables

Appendix Overview	C-1
G-code Tables	C-1

Appendix D

Allen-Bradley 7300 Series CNC Tape Compatibility

Appendix Overview	D-1
G-code Compatibility Considerations	D-1
M-code Compatibility Considerations	D-3
Offset Compatibility Considerations	D-4
Additional Feature Compatibility Considerations	D-6
9/Series G-codes Applicable to the 7300 Series CNC	D-9
7300 Series Features Not Supported	D-10

Using This Manual

1.0 Chapter Overview

This chapter describes how to use this manual. Major topics include:

- **how the manual is organized and what information can be found in it.**
- **how this manual is written and what fundamentals are presumed to be understood by reader.**
- **definitions for certain key terms.**

1.1 Audience

We intend this manual for use by those who program and/or operate any one of the family Allen-Bradley 9/Series CNCs. We assume that a person has some familiarity with the operation and programming of a CNC.

1.2 Manual Design

We divided the manual this way:

For information about:	Refer to:
how to operate the control	chapters 3 - 9
how to program the control	chapters 10 - 29
softkeys	appendix A
error and operator messages in alphabetical order	appendix B
standard G-codes used to program the control	appendix C
the Allen-Bradley 7300 Series CNC tape reader	appendix D

We placed section headings in the left margin of each page, and included illustrations and examples as aids in programming and operating the control.

Table 1.A provides a summary of each chapter.

Table 1.A
Manual Organization

Chapter	Title	Summary
1	Manual Overview	Manual overview, intended audience, definition of key terms, how to proceed.
2	Basic Control Operation	A brief description of the control's basic operation including power up, MTB panel, operator panel, access control, and E-STOP.
3	Offset Tables and Setup	Basic setup of the offset table, other initial operating parameters.
4	Manual and MDI Operation	How to use the manual operate mode including, homing the machine, jog hand-wheel, jog continuous, and jog increment. Also covered are the basics for MDI operation.
5	Editing Programs On Line	How to create, edit, and save a part program on line.
6	Editing Part Program Off Line	How to create, edit, and save a part programs from ODS off line.
7	Running a Program	How to select and execute a program automatically. This covers program checking as well as part production. Also details on special running conditions.
8	Displays/ Graphics	How to access and interpret the different position displays. How to use the Quick Check and Active Program graphics features.
9	Communications	Communications with peripheral devices. Includes sections on communication port parameters, inputting and outputting AMP, PAL, Offsets, and programs.
10	Introduction to Programming	Tape format, structure and format of the programming language for the control.
11	Coordinate System Offsets	Machine coordinate system, Preset Work coordinate systems, PAL offsets, and external offsets
12	Overtravels and Programmable Zones (G22, G23)	Hardware and software overtravels, programmable zone 2 (G22, G23), programmable zone 3 (G22.1, G23.1), and resetting overtravels
13	Coordinate Control	Describes absolute/incremental modes, inch/metric modes, radius/diameter modes, and scaling
14	Axis Motion	G-words define how the tool is positioned to the endpoint of a move. Also sections on automatic machine home, dwell, mirroring, and axis clamp
15	QuickPath Plus	Describes QuickPath Plus programming
16	Using Chamfers and Corner Radius	Describes the ,C- and ,R-words programmed for chamfering and corner radius
17	Spindles	Describes spindle speed control, spindle orientation, spindle direction, and Virtual C axis
18	Programming Feedrates	Describes acc/dec, AMP-assigned feedrates, feedrate control, short block acc/dec
19	Dual Axis Operation	Describes parking, homing, programming, offset management for a dual axis
20	Tool Control	Selecting a tool. Activating and deactivating tool length offsets. Also tool control features such as Random Tool and Tool Life Management.
21	Cutter Compensation	Describes the Tool Tip Radius Compensation feature (TTRC) that offsets for different tool diameters.
22	Using Pocket Milling Cycles	Describes the fixed cycles (canned cycles) for drilling operations and the G-words and parameters used to define them.
23	Using Post Milling Cycles	
24	Using Hemisphere Milling Cycles	
25	Using Irregular Pocket Milling Cycles	
26	Milling Fixed Cycles	
27	Skip and Gauging Cycles	Describes the 9/Series Probing features. Includes the tool measuring gauge feature.
28	Paramacros	Describes paramacros including calling, arithmetic functions, looping, decision making
29	Program Interrupts	Describes the program interrupt feature. This feature is used to call a subprogram or paramacro program whenever a signal corresponding to that program is sent to PAL by the operator.
30	Dual Processing Systems	Necessary information on capabilities and programming methods for dual processing systems.
31	Transfer Line Cycles	Describes operation and programming of T-Line-9 part program templates for transfer line controls.

Table 1.A (cont.)
Manual Organization

Appendix	Title	Summary
Appendix A	Softkeys	Describes softkeys and their functions for softkey levels 1 and 2. Also the softkey tree displaying all levels of softkeys and their location is shown.
Appendix B	Error and Operator Messages	An alphabetical listing of 9/Series system messages with brief descriptions.
Appendix C	G and M Code Tables	Lists the G-codes used to program the control.
Appendix D	A-B 7300 Series CNC Tape Reader	Detailed 7300 Series CNC tape compatibility feature developed on the control.

1.3 Reading this Manual

To make this manual easier to understand, we included these explanations of terms and symbols:

- All explanations, illustrations, and charts presented are based on standard CNC functions. Operations may differ from the basic information provided in this manual depending on the configuration of the machine tool. For details, refer to the manuals prepared and supplied by the system installer.
- Some of the softkey functions may be purchased as optional features. This manual assumes that all of the optional features have been purchased.
- Explanations and illustrations are presented based on the movement of the cutting tool on a fixed workpiece.
- The control allows the use of any alphabetic character for expressing a numerically controlled axis. This manual uses X, Y, and Z for the first, second, and third axes on the basic coordinate system respectively. I, J, and K represent the integrand words for the axes.
- The term AMP is an abbreviation for Adjustable Machine Parameters. These parameters are used to configure a control to a specific machine. Setting of AMP is usually done by the system installer.
- Key names designated between the [] symbols are found on the operator panel.
- Key names designated between the { } symbols are softkeys found below the CRT.
- Switch and button names on the standard MTB panel are designated between the < > symbols.

- The term PAL is an abbreviation for Programmable Application Logic. This is a ladder logic program that processes signals between the CNC and the machine. It is usually programmed by the system installer.
- System Characteristics:

Metric
Absolute
IPM

1.4 Terms and Conventions

To make this manual easier to read and understand, we shortened the full product names and features. Shortened terms include:

- AMP – Adjustable Machine Parameters
- Backup – Memory storage area not requiring battery maintenance
- CNC – Computer Numerical Control
- CPU – Central Processing Unit (the computing part of the control)
- CRT – Cathode Ray Tube (the control's monitor screen)
- the control – the 9/230, 9/240, 9/260 or 9/290 CNC
- ESTOP – Emergency Stop
- Flash memory – programmable, non-volatile memory
- HPG – Hand Pulse Generator
- I/O – Input/Output
- MDI – Manual Data Input
- modal – an operating condition that remains in effect on the control until cancelled or replaced
- MTB – Machine Tool Builder
- ODS – Offline Development System
- PAL – Programmable Application Logic
- RAM – Random Access Memory resident on the 9/240
- softkeys – the row of keys directly below the screen
- system installer – the company or contractor responsible for installing this control on the machine

1.5
Warnings, Cautions, and
Important Information

We indicate information that is especially important by the following:



WARNING: indicates circumstances or practices that can lead to personal injury as well as to damage to the control, the machine, or other equipment.



CAUTION: indicates circumstances or practices that can lead to damage to the control or other equipment.

Important: indicates information that is necessary for successful application of the control.

1.6
Related Publications

9/Series Documentation

Pub. No.	Document Name
8520-4.3	9/Series CNC PAL Reference Manual
8520-5.1.1	9/Series CNC Lathe Operation and Programming Manual
8520-5.1.3	9/Series CNC Mill Operation and Programming Manual
8520-5.1.4	9/Series CNC Grinder Operation and Programming Manual
8520-5.1.5	9/Series Data Highway Plus Communication Module User Manual
8520-5.1.6	9/Series MMS/Ethernet Communication Module User Manual
8520-5.2	9/Series CNC OCI User Manual Supplement
8520-6.2	9/Series CNC Integration and Maintenance Manual
8520-6.4	9/Series CNC AMP Reference Manual
8520-6.5	T-Line-9 Transfer Line Quick Start Guide
8520-6.6	9/Series CNC OCI Installation Manual
8520-6.7	9/Series CNC OCI API Developer's Guide
MCD-5.1	Offline Development System User's Manual

END OF CHAPTER

Basic Control Operation

2.0 Chapter Overview

This chapter describes how to operate the Allen-Bradley 9/Series control, including:

Topic:	On page:
MTB panel	2-12
{FRONT PANEL}	2-15
Power-up	2-23
Emergency stops	2-24
Access control	2-25
Changing modes	2-33
Display system and messages	2-37
Input cursor	2-41
{REFORM MEMORY}	2-41
Removing an axis	2-43
Time part count	2-43

We also tell you about the control conditions automatically assumed at power up.

2.1 Operator Panel Operations

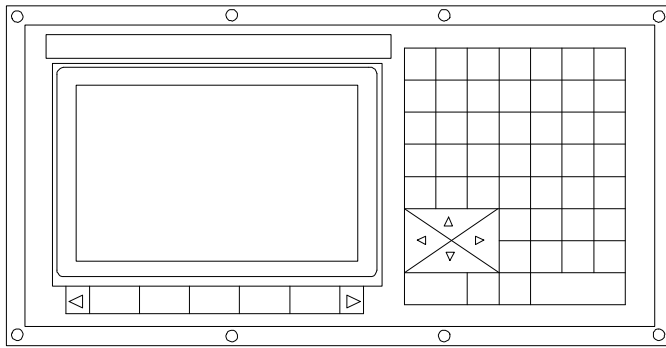
Use the operator panel to perform these operations:

- display a part program
- display control status and tool position
- edit a part program
- display and enter tool offset data
- display the status of input/output signals
- display and enter programmable zone boundaries
- set the level of protection of part programs, tool offset data, AMP data, etc.

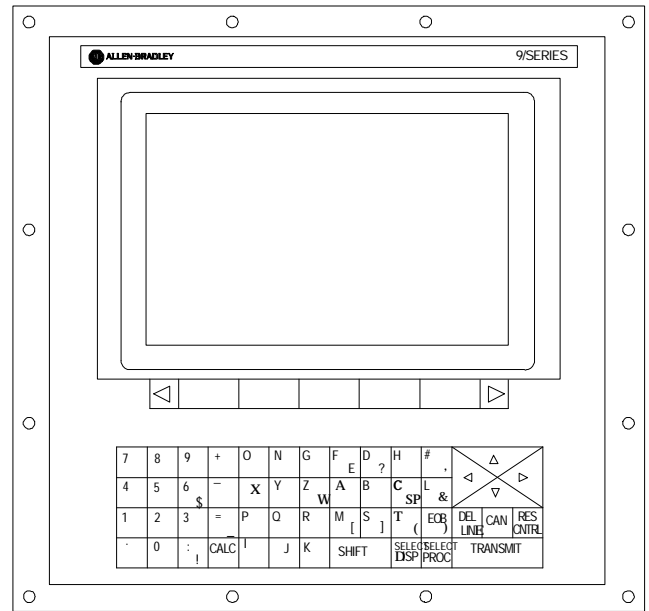
Use the operator panel to perform many other operations. We describe these operations in the remaining chapters of this manual.

Figure 2.1 shows the different operator panels available. The color operator panel has identical keys and softkeys in a slightly different configuration. The portable operator panel has the same key locations as the monochrome operator panel but can be removed from the 9/Series I/O ring.

Figure 2.1
Operator Panels



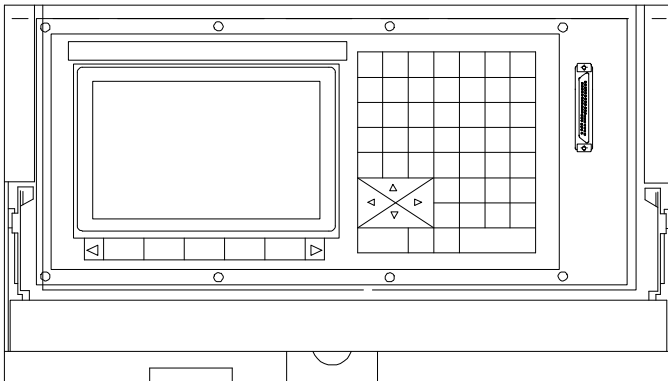
Monochrome Operator Panel



Color Operator Panel
(CRT and Flat Panel)

1946

Removable Operator Panel



2.1.1
Keyboard

Table 2.A explains the functions of keys on the operator panel keyboard. In this manual, the names of operator panel keys appear between [] symbols.

Table 2.A
Key Functions

Key Name	Function
Address and Numeric Keys	Use these keys to enter alphabetic and numeric characters. If a key has two characters printed on it, pressing it normally enters the upper left character. Holding down the [SHIFT] key while pressing it enters the lower right character.
Cursor Keys • , • , • , •	Use these keys to move the cursor left, right, up and down in the data display area (lines 4-21) of the screen. We refer to these keys as the left, right, up, and down cursor keys respectively.
[SHIFT] and • or •	Press the right or left cursor keys while holding down the [SHIFT] key to move the cursor right and left on any line on which that data is being input. (normally screen lines 2-3).
[SHIFT] and • or •	Press the up or down cursor key while holding down the [SHIFT] key to roll the display page forward or backwards.
Calculation Key [CALC]	Pressing this key enables a calculator type function on the control. Basic mathematical expressions may be evaluated using this feature. For details, see page 2-4 .
Delete Key [DEL]	Pressing this key deletes the character to the left of the cursor on input lines.
[SHIFT] and [DEL]	Pressing this key deletes all keyed-in data currently displayed on input lines.
[CAN]	Clear the most recently generated, active error message
Transmit Key [TRANSMIT]	The data entered and displayed on the input lines is sent to the control when the [TRANSMIT] key is pressed, for example a password or a program block.
Block Reset [RESET]	This performs a block reset. See page 2-4 .
Control Reset [RESET] + [SHIFT]	Pressing the reset key while holding down the shift key performs a control reset. See page 2-4 for details.
Display Select [DISP SELECT]	Use this key to display the different axis position display softkeys.
End of Block [E. O. B.]	Use this key to enter an End of Block character when editing a program or when writing an MDI program.
[PROC SELECT]	This key is used for dual process lathe systems only. Pressing this key toggles the control from process to process.
[SHIFT] and [•]	Press the [SHIFT] key while holding down the [•] softkey to jump to the online search monitor screen. Press these keys again to return to the previous screen.

Reset Operations

Block Reset

Use the block reset feature to force the control to skip the block execution. To use the block reset function, program execution must be stopped. If program execution stops before the control has completely finished the block execution, a block reset aborts any portion of that block that has not been executed. If program execution stops after the complete block execution (as in the case of single block execution or a M00 etc.), the control aborts the execution of the entire following block.

Press [RESET] key on the operator panel to perform a block reset.

Control Reset

You can return the control to the default parameters, clear any programming errors, and cancel any MDI commands by executing a control reset. After you execute a control reset, any active program resets to the first block; any programmed offsets or rotations of the coordinate systems reset to default, and any MDI command is discarded. All of the operating parameters return to the standard AMP-assigned values, including any AMP-assigned G-codes active at power-up (except Inch/Metric which remains in its last programmed state at control reset).

Press the [RESET] key on the operator panel while holding down the [SHIFT] key to execute a control reset.

2.1.2 Calculator Function

The 9/Series control is equipped to evaluate simple mathematical expressions during the course of operation or programming.

To use the calculator function, line 2 of the screen must be blank. There can be no prompt on the input line of the screen when you attempt to do calculations. This completely disables any calculation operation when in MDI mode. If you attempt to enter the calculator function while another prompt is active, the control generates the error message "CANNOT CALCULATE - PROMPT PRESENT."

Use the calculator function as follows:

1. Press the [CALC] key on the operator panel. The "CALC:" prompt appears on the input line of the screen (line 2).
2. Enter a mathematical equation on the input line by pressing the desired keys on the operator panel.
3. Press the [TRANSMIT] key to evaluate the expression. The answer to the expression is displayed on the input line.

Expressions entered on the input line cannot exceed a total of 25 characters. Only numeric or special mathematical operation characters as described below can be entered next to the "CALC:" prompt. Any character that is not numeric or an operation character you enter on the input line generates the error message "INVALID CHARACTER."

The largest number you can enter for a calculate function is 214748367. You cannot enter a number larger than 10 digits. If control displays a number that is too large (longer than 10 digits), the control displays the error message "NUMBER IS OUT OF RANGE". If the number entered or calculated is greater than 10 digits, control displays the error message "MATH OVERFLOW."

Any fractional numbers cannot exceed .999999 (6 decimal places). If you exceed this number of decimal places, the control automatically rounds off. If this seventh digit is less than 5, the control rounds down. If this seventh digit is 5 or greater, the control rounds up.

Any data entered on the input lines can be edited as described on page 2-41.

To disable the calculator function, press the [CALC] key again. The "CALC:" prompt is removed from the input line.

Use the characters in Table 2.B to indicate mathematical operations.

Table 2.B
Mathematical Operators

*	Multiplication
/	Division
+	Addition
-	Subtraction
[]	Brackets
#	Get Paramacro Value

The control executes mathematical operations in this order:

1. Any part of the expression that is between the brackets [] is evaluated first. The values of paramacro variables are also substituted for the #xxxx as the first operation performed.
2. Multiplication and division are evaluated second.
3. Addition and subtraction are evaluated last.

If the same level of evaluation is performed the left most operation takes priority.

Example 2.1
Mathematic Expressions

Expression Entered	Result Displayed
12/4*3	9
12/[4*3]	1
12+2/2	13
[12+2]/2	7
12-4+3	11
12-[4+3]	5

Table 2.C lists the function commands available with the [CALC] key.

Table 2.C
Mathematical Functions

Function	Meaning
SIN	Sine (degrees)
COS	Cosine (degrees)
TAN	Tangent (degrees)
ATAN	Arc Tangent (degrees)
ASIN	Arc Sine (degrees)
ACOS	Arc Cosine (degrees)
SQRT	Square Root
ABS	Absolute Value
BIN	Conversion from Decimal to Coded Decimal
BCD	Conversion from Coded Decimal to Decimal
ROUND	Rounding Off (nearest whole number)
FIX	Truncation Down
FUP	Truncation Up
LN	Logarithms (natural log)
EXP	Exponent

When you program these functions, place the value that the function is to be performed on in brackets, for example, SIN [10]. The exception to this is the arc tangent function. The format for ATAN requires the division of two values. For example, ATAN [10]/[2] is used to calculate the arc tangent of 5.

The functions in Table 2.C are executed from left to right in a program block. These functions are executed before the control executes any mathematical operators like addition or subtraction. This order of execution can only be changed by enclosing operations in brackets []. Operations enclosed in brackets are executed first.

Example 2.2
Format for [CALC] Functions

SIN[2]	This evaluates the sine of 2 degrees.
SQRT[14+2]	This evaluates the square root of 16.
SIN[SQRT[14+2]]	This evaluates the sine of the square root of 16.

Example 2.3
Mathematical Function Examples

Expression Entered	Result
SIN[90]	1.0
SQRT[16]	4.0
ABS[-4]	4.0
BIN[855]	357.0
BCD[357]	855.0
ROUND[12.5]	13.0
ROUND[12.4]	12.0
FIX[12.7]	12.0
FUP[12.2]	13.0
FUP[12.0]	12.0
LN[9]	2.197225
EXP[2]	7.389056

Important: Precaution must be taken when performing calculations within the brackets []. The operations within the bracket are performed first, and then the function is performed on this resultant. For example

ROUND[2.8+2.6]; The result of this is 5.0

The values in the brackets are added together first and then rounded, not rounded and then added together.

Paramacro Variables in CALC Operations

Any paramacro variable can be accessed through the CALC function. Include a # sign followed by the paramacro variable number. When the calculation is performed the value of that paramacro variable is substituted into the equation. You can not change the value of paramacro variables with the CALC function. Local parameters are only available for the currently active nesting level of the control (main program, or one of four nested macro programs). You can not perform calculations that contain any paramacro variables if the control is currently executing a program block. The control must be in either cycle stop state, or E-Stop.

Example 2.4
Calling Paramacro Variables with the CALC Function

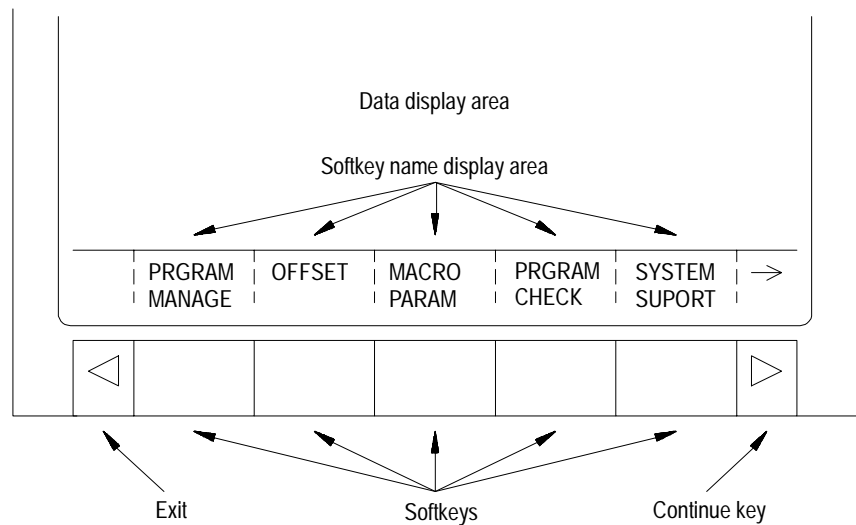
Expression Entered	Result Displayed
#100	Display current value of variable #100
12/#100*3	Divide 12 by the current value of #100 and multiply by 3
SIN[#31*3]	Multiply the value of #31 (for the current local parameter nesting level) by 3 and take the sine of that result

2.1.3
Softkeys

We use the term **softkey** to describe the row of 7 keys at the bottom of the CRT. Each function is displayed on the CRT directly above the softkey. Softkey names are shown between the { } symbols in this manual.

The control offers a variety of functions that can be initiated by using the softkeys. The operator panel has 5 softkeys whose function names are displayed in the softkey area at the bottom of the screen (lines 23-25 – see Figure 2.2).

Figure 2.2
Softkeys



We often describe softkeys as being on a certain level, for example softkey level 3. We use the level of the softkey to determine the location or necessary path to reach that particular softkey function. For example, to get to a softkey on level 3, you must press a specific softkey on level 1, followed by a specific softkey on level 2. For a listing of all the softkeys and their respective levels, refer to appendix A.

Softkey level 1 is the initial softkey level the control displays at power-up. Softkey level 1 always remains the same and all other levels are referenced from softkey level 1.

The softkeys on opposite ends of the softkey row have a specific use that remains standard throughout the different softkey levels.

On the:	Is the:
left	exit softkey displayed with the up arrow {•}
right	continue softkey displayed with the right arrow {•}

- Use the exit softkey {•} on the far left to regress softkey levels. For example, if you are currently on softkey level 3 and you press the exit softkey, the softkeys change to the softkeys previously displayed on softkey level 2. When you press the exit softkey while holding down the shift key, the softkey display is returned to softkey level 1 regardless of the current softkey level.
- When more than 5 softkey functions are available on the same level, the control activates the continue {•} softkey at the far right of the softkey area. When you press the continue softkey, the softkey functions change to the next set of softkeys on that level.

Important: The continue softkey is not active when the number of softkey functions on that level does not exceed 5.

For example:

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
--	------------------	--------	----------------	-----------------	------------------	---

When softkey level 1 is reached, the above set of softkeys is displayed. Pressing the continue softkey {•} displays the remaining softkey functions on softkey level 1.

(softkey level 1)

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→
--	----------------	-----------------	---------------	-----------------	--	---

The exit softkey is not displayed since the softkeys are already on softkey level 1.

The softkey functions for level 1 and level 2 are explained in appendix A. Softkey functions for level 3, or higher, are explained in the sections that apply to their specific operations.

To use a softkey function, press the plain, unmarked button directly below the description of the softkey function.

Important: Some of the softkey functions are purchased as optional features. This manual assumes that all available optional features have been purchased for the machine. If an option is not purchased, the softkey is blank.

2.1.4
CRT

The control can be purchased with a 9-inch monochrome monitor or a 12-inch color monitor.

Certain lines of the screen are dedicated to displaying specific information:

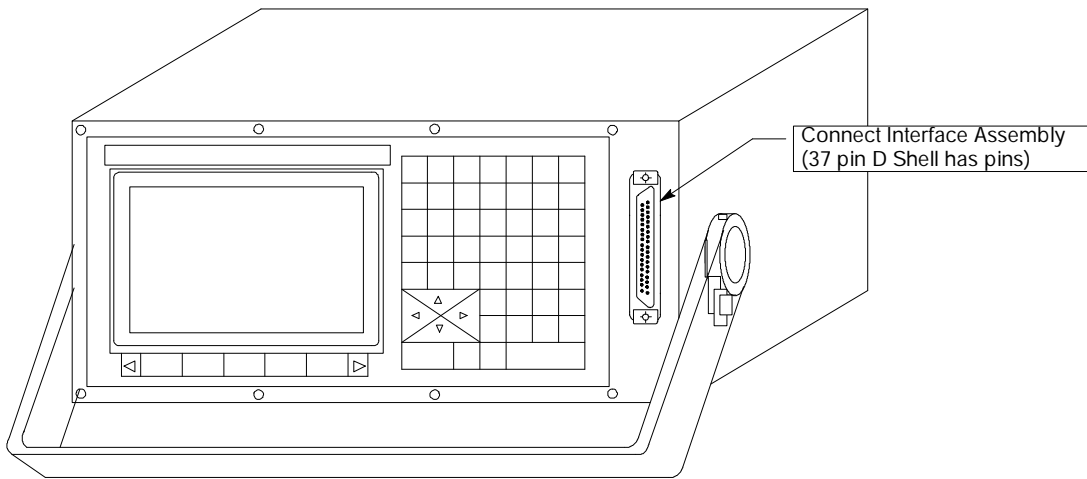
Lines:	Display information:
line 1 machine/ system message area	If an error occurs or a message is generated for any reason during machine operation or program execution, the control displays the corresponding machine/system message in this area. Only the highest priority, most current message is displayed here.
lines 2-3 input lines	When you enter data using the keyboard, the control displays the characters corresponding to the keys pressed until you press the [TRANSMIT] key. Some screens can only have line 2 as an input line.
lines 4-20 data display area	The control displays axis position data, listing of the part program, tool offset data, G-, M-, H-, T-, F-, S-, and D-codes, graphics, and other data, as determined by the selected display. See chapter 8.
lines 21-22 PAL message area	The control displays any messages generated by the control's PAL program in this area
lines 23-25 softkey display area	The control displays the currently available softkey functions in this area.

2.1.5 Portable Operator Panel

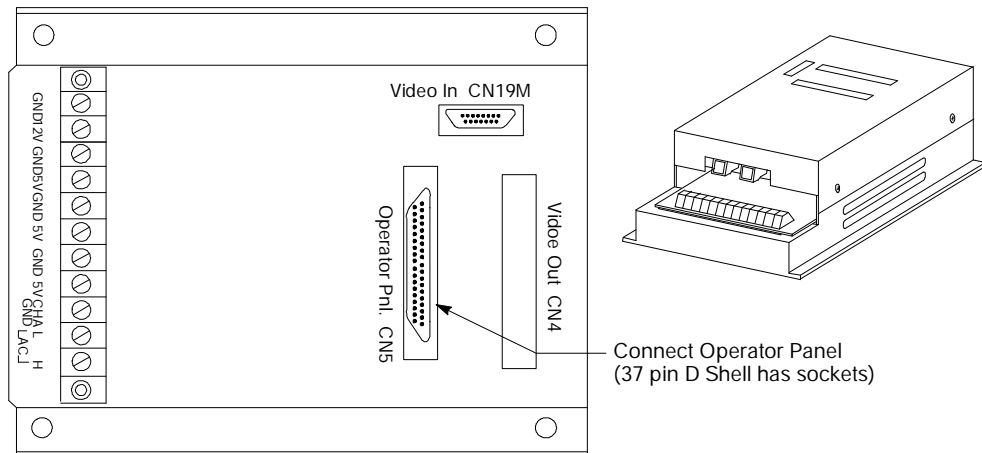
The control can be purchased with a 9-inch monochrome portable operator panel. This panel can be attached or detached to the 9/Series I/O ring operator panel interface assembly at any time without disrupting control operation.

The portable operator panel is attached through a 10 ft portable operator panel interface cable with a 3-pin D-shell connector at each end. One end of the cable attaches to the front of the portable operator panel. The other end attaches to the portable operator panel interface assembly. Refer to your system installer's documentation for details on attaching and detaching the portable operator panel and the locations of the operator panel interface assembly.

Removable Operator Panel



Interface Assembly



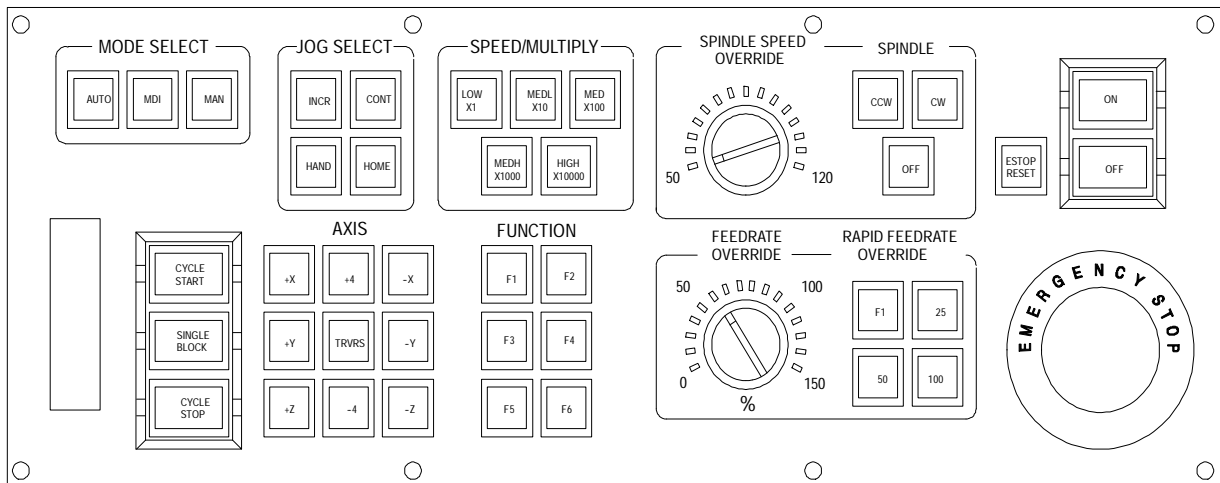
2.2
The MTB Panel

Figure 2.3 shows the push-button MTB panel. Table 2.D explains the functions of the switches and buttons on the MTB panel. Other optional or custom MTB panels may be used. Refer to the documentation prepared by your system installer for details.

We show button names found on the push-button MTB panel between the < > symbols throughout this manual. The push-button MTB panel uses defaults when you turn on power to the control. Table 2.D contains these defaults.

Most of the switches or buttons on the MTB panel are configured by your system installer's PAL program. We assume that PAL has been written as intended for normal operation. If a switch or button does not work the way it is described in this manual, refer to documentation prepared by your system installer.

Figure 2.3
Push-Button MTB Panel



19930

Table 2.D
Functions of the Buttons on the Push-Button MTB Panel

Switch or Button Name	How It Works ☐ = Default for Push-Button MTB Panel
MODE SELECT	<p>Selects the operation mode</p> <p>AUTO -- automatic mode</p> <p>MANUAL -- manual mode ☐</p> <p>MDI -- manual data input mode</p>
JOG SELECT	<p>Selects the jog method to be active in manual mode</p> <p>HANDWHEEL -- HPG (hand pulse generator) jog</p> <p>INCREMENTAL -- incremental jog</p> <p>CONTINUOUS -- continuous jog ☐</p> <p>HOME -- machine home</p>
SPEED/MULTIPLY	<p>Selects an axis feedrate or axis feed amount multiplication ratio used in the manual mode. Each selection modifies the active feedrate by a value set in AMP. Modification also depends on the setting of <JOG SELECT> as described below:</p> <ul style="list-style-type: none"> • HANDWHEEL When in handwheel jog mode, SPEED/MULTIPLY alters the handwheel resolution by a factor determined in AMP. Your system installer sets the value for: ☐ - LOW X1 - MEDL X10 - MEDL X100 You cannot use the values listed below for handwheel jog: - MEDH X1000 - HIGH X10000 • INCREMENTAL When in incremental jog mode, SPEED/MULTIPLY alters the incremental jog <u>distance</u> by a factor set in AMP by your system installer. Your system installer sets a value for the selections. The incremental jog <u>speed</u> is fixed to medium but can still be controlled by <FEEDRATE OVERRIDE>. • CONTINUOUS When in continuous jog mode, SPEED/MULTIPLY acts as a feedrate selection switch which has values set in AMP by your system installer. Your system installer sets a value for all 5 selections independently for each axis. <FEEDRATE OVERRIDE> can be used for speed adjustments. <p>Important: The values for the different <SPEED/MULTIPLY> selections are configured by your system installer.</p>

Table 2.D
Functions of the Buttons on the Push-Button MTB Panel

Switch or Button Name	How It Works ☐ = Default for Push-Button MTB Panel
SPINDLE SPEED OVERRIDE	Selects the override for programmed spindle speeds in 5% increments within a range of 50% to 120%.
SPINDLE or SPINDLE DIRECTION	Selects spindle rotation, clockwise (CW), spindle stop (OFF), counterclockwise (CCW). Can be overridden by any programmed spindle direction command.
FEEDRATE OVERRIDE	Selects a feedrate override percentage for the feedrate programmed with an F-word in any of the feedrates modes (G93/G94/G95) and the reciprocation feedrate programmed with an E-word. <FEEDRATE OVERRI DE> has a range of 0% to 150% of the programmed feedrate and alters the programmed feedrate in 10% increments. When set to 0%, the control is effectively in feedhold.
RAPID FEEDRATE OVERRIDE	Selects the override for rapid feedrates. Select from [F1] 25%, 50%, and 100% where F1 is a rapid feedrate override setting established in AMP by the system installer.
EMERGENCY STOP	This button stops machine operation and disables the spindle and axis drives when pressed.
E-STOP RESET	This button resets an emergency stop condition when pressed. Before pressing this button the condition that caused the E-Stop should be resolved.
CYCLE START	The control begins or resumes part program execution, MDI program execution, or program check when this button is pressed.
CYCLE STOP	The control stops part program execution, MDI execution, or program check when this button is pressed. If pressed during the execution of a program block a cycle suspend state occurs.
SINGLE BLOCK	The control executes or checks one block of a part program or MDI entry each time the <CYCLE START> button is pressed when single block is active.
AXIS/DIRECTION	These buttons are used for manual operations. They select an axis and direction when <JOG SELECT> is set for continuous, incremental, or home. If <JOG SELECT> is set for handwheel, these buttons select an axis only. Direction is then determined by handwheel rotation.
TRVRS	Hold this button down while executing a continuous jog move to override the active feedrate and jog an axis in rapid traverse.
F1 - F4	The functions for these buttons are assigned by the system installer. Refer to the documentation prepared by the system installer for details.
JOG RETRACT	Use jog retract to jog a cutting tool away from the workpiece during automatic or MDI program execution. The control can retrace the jog moves and return the cutting tool automatically to the workpiece by pressing <CYCLE START>. Refer to chapter 7 for more on this.
BLOCK RETRACE	To retrace the tool path in a part program already executed (up to 15 blocks), press this button.
ON	Turns on power to the control.
OFF	Turns off power to the control.

Important: Many of the override switch settings may be disabled by programming the correct M-code or setting a particular paramacro parameter. Refer to their respective sections for details on these features.

2.3
Software MTB Panel
{FRONT PANEL}

The 9/Series control offers a software MTB panel that performs many of the functions of an MTB panel. This feature uses softkeys instead of the normal switches and buttons of a panel. If the control uses a standard MTB panel (described on page 2-12), or some other custom panel, the requests for operations from the panel takes priority. This means that requests of the software MTB panel are ignored if another request is sent from a standard or custom MTB panel.

The software MTB panel's operation depends on PAL, especially if the control uses either the standard or a custom MTB panel. Refer to documentation prepared by your system installer for details on using the software MTB panel. Your system installer uses PAL to disable the {FRONT PANEL} softkey.

Important: By using the PAL flags \$\$WF1 and \$\$WF2, you can enable and disable features available on the software front panel.

Refer to your PAL reference manual for additional information.

The software MTB panel can control these features:

Feature	Description
Mode Select	Select either Automatic, MDI, or Manual modes as the current operating mode of the control.
Rapid Traverse	This feature replaces the feedrate when executing a continuous jog move with the rapid feedrate.
Feedrate Override	Selects a feedrate override percentage for feedrates programmed with an F-word, in 10% increments within a range of 0% to 150%. When set to 0%, the control is effectively in feedhold.
Rapid Feedrate Override	Selects the override for rapid feedrates. Select from F1, 25%, 50%, and 100%, where F1 is a rapid feedrate override setting established in AMP by the system installer.
Spindle Direction	Selects spindle rotation, clockwise (CW), spindle stop (OFF), or counterclockwise (CCW). Is overridden by any programmed spindle direction command.
Spindle Speed Override	Selects the override for programmed spindle speeds in 5% increments within a range of 50% to 120%.
Dry Run Mode	Places the control in Dry Run Mode. This replaces feedrates with the Dry Run feedrates. See chapter 7.
Block Delete	Allows the activation of the block delete feature "I or /I" (See chapter 7). Block delete 2 - 9 are not available with the { FRONT PANEL }.
M-Function Lock	Allows select M-, S-, T-, and B-codes to be ignored. See chapter 7.
Optional Stop	This enables or disables the M01 optional stop code. When this feature is "ON," an M01 in a part program stops automatic execution. When this feature is "OFF," an M01 in a part program is ignored. See chapter 10 for details on M01.
Single Block	The control executes or checks one block of a part program or MDI entry each time the <CYCLE START> button is pressed when this feature is on.
Mirror Image	Mirrors the axis commands in the part program around the selected axis. See chapter 14.
Axis Inhibit	Prevents axis commands from being executed. The control simulates axis motion on inhibited axes by using Acc/Dec and feedrates; however, no actual axis motion for the inhibited axes is generated. See chapter 7.
Jog the Axes	Allows manual motions to be performed in any one of the jogging modes. See section 4.1. Multi-axis jogs may not be performed by using the Software MTB feature.
Set Zero	Change the tool's current position in the work coordinate system to 0 for the selected axis. This is done by shifting the work coordinate system. See chapter 11.
Block Retrace	This allows up to 15 part program blocks to be retraced during program execution and allow the tool to automatically re-execute these blocks. See chapter 7.
Jog Retract	Allow the tool to be manually jogged away from the part and then automatically return the tool to the part by retracing the jogged moves. Up to 15 jog moves can be remembered. See chapter 7.
Cycle Start	The control begins or resumes part program execution, MDI program execution, or program check when you press this button.
Cycle Stop	The control stops part program execution, or program check when you press this button. If pressed during the execution of a program block, a cycle suspend state occurs.

Software MTB Panel Screen

To use the software MTB panel feature, follow these steps:

1. From the main menu screen, press the {FRONT PANEL} softkey.
(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWIT CH LANG		→

The Software MTB Panel screen displays the current status of the alterable features.

SOFTWARE FRONT PANEL						
	MODE SELECT:		MDI			
	RAPID TRAVERSE:		OFF			
	FEDRATE OVR:		0%			
	RAPID FEEDRATE OVR:		100%			
	SPI NDL E DI RECTI ON:		CW			
	SPI NDL E SPEED OVR:		50%			
	DRY RUN MODE:		OFF			
	BLOCK DELETE:		OFF			
	M-FUNC LOCK:		OFF			
	OPTIONAL STOP:		OFF			
	SINGLE BLOCK:		OFF			
	MIRROR IMAGE:					
	AXIS INHIBIT:					
	USE CURSOR FOR SELECTION					
↑	JOG AXI S	PRGRAM EXEC				

2. Press the up or down cursor keys to select the function you want to change. The selected function appears in reverse video.
3. Alter the value of the function by pressing the left or right cursor keys (except for the Mirror Image and Axis Inhibit features). When you select the Mirror Image or Axis Inhibit features, the softkey names change to the axis names. Press the softkey that corresponds to the axis (or axes) that you want to assign to these features.

Jog Screen

We assume that you have performed the steps to display the Software Front Panel screen. Make sure that the function selected on the Software Front Panel screen is not the Mirror Image or the Axis Inhibit features.

1. Press the {JOG AXIS} softkey.

(softkey level 2)

↑	JOG AXIS	PROGRAM EXEC				
---	-------------	-----------------	--	--	--	--

This screen appears:

		E-STOP	
PROGRAM [mm]	F	0.000	MMPM
Z 0.000	S	0.0	
R X 0.000	T	0	
C 359.99	FILENAME		
		SUB_NAME	
MEMORY	MAN	STOP	
AXIS SELECT:	X		
JOG SELECT:	CONTINUOUS		
SPEED/MULTIPLY	LOW		
HPG NUMBER	0		
↑	SET ZERO	JOG AXIS +	JOG AXIS -

2. Select the axis, jog type, speed multiply value (see chapter 4), and HPG number (if HPG has been selected as the type of jog) by using the up and down cursor keys, and the left and right cursor keys to alter the value assigned to that function.
3. If the jog type is not HPG, then press the softkey that corresponds to the direction on the selected axis to jog (if HPG is the jog type then the direction of HPG rotation determines the direction to jog).
4. If you want to shift the current work coordinate system to force the current tool position to be the zero point of the work coordinate system, press the {SET ZERO} softkey. The {SET ZERO} softkey does not function when the control is in handwheel mode.

Program Execute Screen

The following assumes that the steps have been performed to display the Software Front Panel screen (see page 2-17). Make sure that the function selected on the Software Front Panel screen is not the Mirror Image nor the Axis Inhibit feature.

1. Press the {PROGRAM EXEC} softkey.

(softkey level 2)

↑	JOG AXIS	PROGRAM EXEC				
---	-------------	-----------------	--	--	--	--

This screen appears. Before you press this softkey, a program should have already been selected for automatic execution as described in chapter 7.

		E-STOP				
PROGRAM[MM]	F	0.000 MPM				
Z 0.000	S	0.0				
R X 0.000	T	0				
C 359.99	FILENAME					
		SUB_NAME				
MEMORY	MAN	STOP				
M100; G90; Z10; G70;						
↑	BLOCK RETRCE	JOG RETRCT	CYCLE START	CYCLE STOP		

2. Select one of these softkey options:

- block retrace
- jog retract
- cycle start
- cycle stop

To Perform a:	Press:
Cycle Start	the softkey that corresponds to the desired feature. Details on these features are described in chapter 7.
Cycle Stop	the softkey that corresponds to the desired feature. Details on these features are described in chapter 7.
Block Retrace	the { BLOCK RETRCE } softkey. The control automatically retraces the previously executed block each time you press. The tool returns to the start point of Block Retrace when you press the { CYCLE START } softkey. Details on Block Retrace can be found in chapter 7.
Jog Retract	the { JOG RETRACT } softkey. This calls up the jog retract screen. Select: the axis, the type of jog, the speed multiply value (see chapter 4), or the HPG number (if HPG has been selected as the type of jog). Use the up and down cursor keys to select the function, and the left and right cursor keys to alter the value assigned to that function.

3. If you selected JOG RETRACT, press the {**JOG AXES +**} or {**JOG AXES -**} softkeys to select the direction to jog the axis. Press the exit {**•**} softkey when you complete the jog retract. The cutting tool automatically retraces the jog moves you made when you pressed the {**CYCLE START**}. See chapter 7.

(softkey level 4)

↑	SET ZERO	JOG AXES +	JOG AXES-			
---	-------------	---------------	--------------	--	--	--

Figure 2.4
Jog Retract Software MTB Panel Screen

		E- STOP	
PROGRAM[MM]		F	00000.000 MPM
Z	00000.000	S	0
R X	00000.000	T	12
C	359.99	FILENAME	
		SUB_NAME	
MEMORY	MAN	STOP	
↑	JOG AXES+	JOG AXES-	

2.4 Power Procedures

The basic procedure for turning power on and off is described in this section. Refer to the documentation prepared by your system installer for more specific procedures.

2.4.1 Turning Power On

Follow this procedure to turn on power to the control:

1. Visually check to make sure that the control and the machine are in normal operating conditions.
2. Press the power <ON> button.

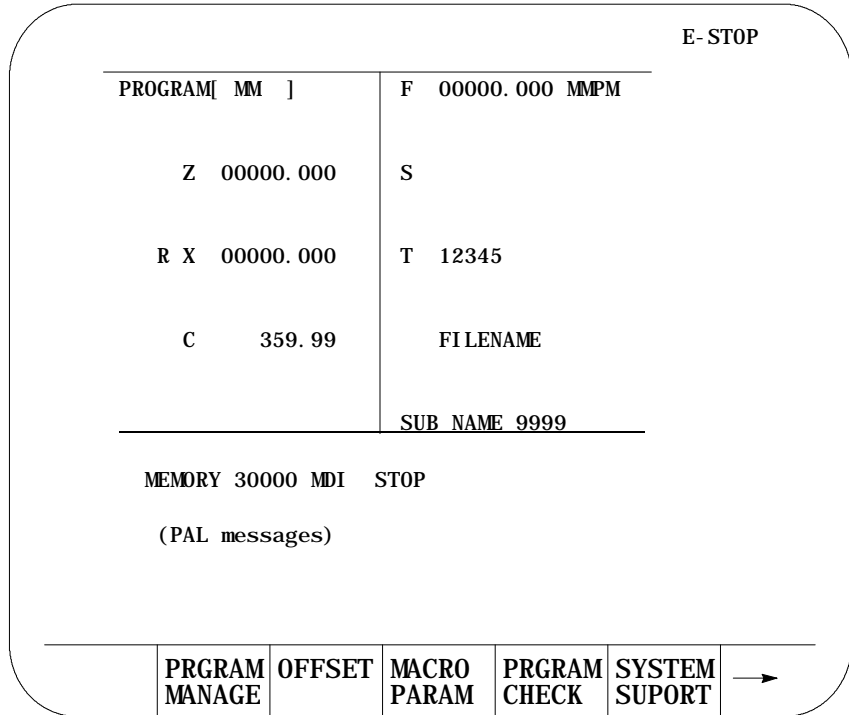
Important: Do not touch any other buttons after you press the power <ON> button until characters appear and remain on the screen.

3. If characters are not displayed on the screen within 15 seconds, press the power <OFF> button immediately.

Important: If the screen does not display characters after you press the power <ON> button within a reasonable warm up period (about 15 seconds), immediately press the power <OFF> button and contact service personnel.

To activate the main menu, press the [TRANSMIT] key.

You see the main menu screen:



The softkeys available on the main menu screen are referred to as “level 1” softkey functions. Some of the softkey functions are purchased as optional and may not appear exactly as shown.

2.4.2 Turning Power OFF

Follow this procedure to turn off power to the control:



CAUTION: To prevent damage to the machine, never turn off power while a part program is being executed. Before turning off power, make sure that the control is in CYCLE STOP.

1. Before you turn the power off, make sure that the control is in CYCLE STOP.
2. Press the <EMERGENCY STOP> button.
3. Make sure that power is turned off to all peripheral devices (tape reader, tape punch, etc.) that are connected to the control.
4. Press the power <OFF> button.

2.5
Control Conditions at
Power-Up

After powering up the control or performing a control reset operation (see page 2-4), the control assumes a number of initial operating conditions. These are listed below:

- **Initial Password Access** is assigned to the level that was active when power was turned off (provided that level is a power-up level selected in access control). If the active level when power is turned off is not a power-up level, then the control defaults to the next lower level that is a power-up level. See page 2-25 on access control.
- **The control is placed in E-Stop.** The control is not allowed to come out of E-Stop if the default AMP is loaded at power-up, or if there is no PAL program loaded in the system. An appropriate error message is displayed.
- **The control defaults to one G-code from each of these groups (as set in AMP):**

Modal Group:	G-code
1	G00 Rapid traverse G01 Linear interpolation
2	G17 Plane Selected G18 Plane Selected G19 Plane Selected
3	G90 Absolute G91 Incremental
4	G22 Programmable Zone 2 and 3 (On) G22.1 Programmable Zone 2 (Off) 3 (On) G23 Programmable Zone 2 and 3 (Off) G23.1 Programmable Zone 2 (On) 3 (Off)
5	G94 Feed per minute G95 Feed per revolution
6*	G70 Inch mode G71 Metric mode
18	G07 Radius Programming Mode G08 Diameter Programming Mode

* This G code group is only established at power up. A control reset will not change the last programmed state of this modal G code group.

To show the current operating conditions at any time, access the G-code status screen as described in chapter 8. If you do this immediately after power-up, it shows the initial operating conditions selected in AMP along with other control power-up default conditions.

2.6 Emergency Stop Operations

Press the red <EMERGENCY STOP> button on the MTB panel (or any other E-Stop switches installed on the machine) to stop operations regardless of the condition of the control and the machine.



WARNING: To avoid damage to equipment or hazard to personnel, the system installer should connect the <EMERGENCY STOP> button, so that pressing the button opens the circuit connected to the E-STOP STATUS terminal on the control. This should disable the axis drives and the spindle drive circuits, which should both be connected to this terminal. Refer to the integration manual or the documentation prepared by your system installer for details.

If equipped with the standard MTB panel, the following occurs automatically after you press the <EMERGENCY STOP> button:

- The control displays “E-STOP” in the message area. This indicates that the control is in the emergency stop state.
- The red light in the <CYCLE STOP> button lights up to indicate that the control is in the feedhold state.
- Power to all axis drive motors is turned off.

Important: If you press the <EMERGENCY STOP> button while a part program is running, program execution can resume at the point of interruption. Refer to the mid-program start feature described in chapter 7.

2.6.1 Emergency Stop Reset

Before resetting the emergency stop state, first locate and eliminate the cause of the emergency stop.

If the <EMERGENCY STOP> button is locked in the pressed position, it must be released before the emergency stop state can be reset. The locked button can be released in different ways depending on its type. With the standard MTB panel, turn the button clockwise until it pops out.

To reset the emergency stop state, press the <E-STOP RESET> button. Once you push the E-Stop Reset button to clear the E-Stop state, the message, “RESETTING E-STOP” displays to alert you that the control is attempting to come out of E-Stop. After the cause of the E-Stop is resolved, the control clears the “E-STOP” message. If the error condition is not cleared, the “RESETTING E-STOP” message clears, but the “E-STOP” message continues to flash as the control remains in E-Stop state.

If the E-Stop occurred during program execution, the control may reset the program when E-Stop reset is performed provided AMP is configured to do so. Assuming that a control reset is performed, program execution begins from the first block of the program when <CYCLE START> is pressed. If the current axis position prohibits this, the operator can manually jog the axes clear, or consider executing a Mid-Program Start. See chapter 7. If no control reset is performed, the remainder of the program block being executed when E-Stop took place is aborted, and a <CYCLE START> begins program execution at the next block.

Important: If the cause of the E-Stop is not eliminated, the circuit connected to the E-STOP STATUS terminals remains open, and the emergency stop state is not reset even when the <E-STOP RESET> button is pressed.

2.7 Access Control

Access control lets the system installer assign different functions of the control to different users by means of a password. See section 2.7.2 for a list of the functions that may be protected on the 9/Series control.

Each protectable function is assigned an access level that is made active when the operator enters the password. When an access level is made active, all functions that are assigned to that access level become available. Access levels range between 1 and 8 where 1 is the highest level and 8 is the lowest. A different password is assigned to each of the different access levels. Eight passwords can be assigned.

Access control only applies to the front panel and softkey inputs. It cannot control inputs from outside the system. For instance, if you control access to the delete function, the user can't delete a file, but a file can be deleted by Mini-DNC software and by Data Highway with the 9/Series Data Highway Plus Communication Module.

Important: If you do not want to use password protection, simply select all functions as accessible for access level 8. Since access level 8 is automatically available at power up, no password is necessary to access any of the functions of the control. Password protection can also be disabled by assigning a level at the power-up level by using the "POWER UP LEVEL" parameter as described in section 2.7.2.

2.7.1 Assigning Access Levels and Passwords

This section describes setting or changing the functions assigned to a particular access level, and changing the password used to activate that access level.

Important: Functions or passwords can be assigned to another access level only if:

- If you have a higher access level than the access level you are attempting to change, this means that if your password is assigned to access level 6, you can only change the functions or passwords for access levels 7 and 8. Functions, or a password, cannot be assigned to access level 6 with a level 6 password.
- Functions that are not available to the current user cannot be assigned to other levels. If a user with access level 6 is changing a lower access level function, access level 6 must have access to any functions that are changed. For example, if you are an access level 6 user, you do not have access to {SYSTEM SUPORT}, you cannot assign or remove {SYSTEM SUPORT} to access level 7.
- The current user must have access to the {ACCESS CONTROL} function.

To change the functions or password of a lower user number, follow these steps:

1. Press the {PASSWORD} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {ACCESS CONTRL} softkey. If the {ACCESS CONTRL} softkey does not appear on the screen, the currently active access level is not allowed to use the {ACCESS CONTRL} function. Enter a password that has access to {ACCESS CONTRL}.

(softkey level 2)

↑				ACCESS CONTRL		
---	--	--	--	------------------	--	--

This screen appears.

PASSWORD NAME -- (current password)			
1	ACTIVE PROGRAM	18	ACCESS CONTROL
2	PROGRAM MANAGE	19	SWITCH LANGUAGE
3	EDIT PROGRAM	20	POWER UP LEVEL
4	COPY PROGRAM	21	TIME PARTS
5	RENAME PROGRAM	22	SI/OEM MESSAGE
6	DELETE PROGRAM	23	SCALING
7	DELETE ALL PROGRAM	24	CHANGE DIRECTORY
8	OFFSETS	25	AUX COMM
9	TOOL WEAR	26	SEARCH MONITOR
10	TOOL GEOMETRY	27	
11	ACTIVE OFFSET	28	
12	BACKUP OFFSET	29	
13	QUICK CHECK	30	
14	SYSTEM SUPPORT	31	
15	PRGRAM PARAMETER	32	
16	AMP	33	
17	PARAMACRO PROG	34	
↑	UPDATE & EXIT	01	02
		03	04
			→

NOTE: Some of these protected features are not available on 9/240 controls.

The softkey names change to display the 8 access levels along with their corresponding passwords (provided that a password has been assigned to that access level). Only the password names of access levels that are lower than the currently active access level are shown.

- Press the softkey that corresponds to the access level that you want to change. The pressed softkey appears in reverse video, and the password name assigned to that access level is moved to the "PASSWORD NAME."

Important: If you attempt to change the functions available to an access level that is equal to or higher than your the current access level, the error message "ACCESS TO THIS LEVEL IS NOT ALLOWED." You cannot change the features that are assigned to your current access level or any level that is higher than your own.

(softkey level 3)

↑	UPDATE & EXIT	01 (NAME)	02 (NAME)	03 (NAME)	04 (NAME)	→
---	------------------	--------------	--------------	--------------	--------------	---

- If you want to enter or change the password for the selected level, edit the password next to the "PASSWORD NAME" prompt by using the input cursor as described on page 2-41 . If you also want to change the functions for this password, move on to step 5. To save the change made to the password and leave the access control screen press the {UPDATE & EXIT} softkey.

Functions that are currently available to the selected level are shown in reverse video on the access level screen.

- Use the up, down, right, and left cursor keys to select the functions to change for that access level. The selected function is shown with a flashing > to the left of the function.
- Pressing the [TRANSMIT] key toggles the function between accessible and inaccessible for that access level.

Important: If you attempt to activate or deactivate a function that is not accessible to the current user's access level, the message "ACCESS TO THIS FUNCTION NOT ALLOWED" is displayed. Only features that are accessible to your the current access level can be selected as accessible or inaccessible to a lower access level.

- Press the {UPDATE & EXIT} softkey to store the changes made to accessible functions for the user levels and return the control to softkey level 1.

(softkey level 3)

↑	UPDATE & EXIT	01 (NAME)	02 (NAME)	03 (NAME)	04 (NAME)	→
---	------------------	--------------	--------------	--------------	--------------	---

2.7.2
Password Protectable
Functions

The following section describes the functions on the 9/Series control that can be protected from an operator by the use of a password. If a user has access to a function, the parameter associated with that function is shown in reverse video on the access control screen.

Access to these functions can be controlled by passwords. Table 2.E describes the function that is enabled (the operator can perform them) if the parameter name is shown in reverse video. If the function is not shown in reverse video, the function is protected and cannot be accessed.

Some parameters enable more than one function. If a parameter that enables multiple functions is not selected as accessible, some of the functions that would be enabled by the parameter can be enabled individually by using other parameters.

Table 2.E
Password Protectable Functions

Parameter Name:	Function becomes accessible when parameter name is in reverse video:
1) ACTIVE PROGRAM	To access these features, both ACTIVE PROGRAM and PROGRAM MANAGE (number 2 below) must be assigned to the user. <ul style="list-style-type: none"> • {SELECT PROGRAM} – Select a program for automatic operation. • {SEARCH} – Search a part program for a character string or sequence number to begin program execution at. • {MID ST PROGRAM} – Start program execution from some location other than the beginning and still set all of the parameters previously defined in the program active. • {T PATH GRAPH} – Enable the active graphics feature. • {CANCEL PROGRAM} – Disable the currently active program from execution. • {SEQ STOP} – Choose a sequence number for program automatic program execution to stop at.
2) PROGRAM MANAGE	<ul style="list-style-type: none"> • {ACTIVE PROGRAM} – All of the functions in 1 above provided ACTIVE PROGRAM (number 1 above) is also selected. • {EDIT PROGRAM} – Edit an existing program or create a new program. • {DISPLY PROGRAM} – Display a program using the display function. • {COPY PROGRAM} – Copy a program to or from memory. • {VERIFY PROGRAM} – Verify that two programs are identical using the verify function. • {COMMENT} – Add comments to a program name in the directory. • {RENAME PROGRAM} – Change a program name. • {DELETE PROGRAM} – Delete a single program stored in memory. • {REFORM MEMORY} – Delete all programs currently stored in memory.
3) EDIT PROGRAM	{EDIT PROGRAM} – Edit an existing program or create a new program.
4) COPY PROGRAM	{COPY PROGRAM} – Copy a program to or from memory or storage device.
5) RENAME PROGRAM	{RENAME PROGRAM} – Not rename a program name.
6) DELETE PROGRAM	{DELETE PROGRAM} – Delete a single program stored in memory
7) DELETE ALL PROG	{REFORM MEMORY} – Delete all programs currently stored in memory.

Table 2.E
Password Protectable Functions

Parameter Name:	Function becomes accessible when parameter name is in reverse video:
8) OFFSETS	<ul style="list-style-type: none"> • { WORK CO- ORD } – Display and alter the preset work coordinate system zero locations and the fixture offset value. • { TOOL WEAR } Display and alter the tool wear amount tables for the different tools. • { TOOL GEOMET } – Display and alter the tool geometry tables. • { ACTIVE OFFSET } – Change the currently active offset number without requiring the programming of a different offset number. • { TOOL MANAGE } – Alter the tool life indicators and other machine specific tool functions. • { RANDOM TOOL } – Allow the use of the random tool tables used to keep track of different tools in different tool pocket (see chapter 20). • { COORD ROTATE } – Rotate all of the work coordinate systems from the machine coordinate system. • { BACKUP OFFSET } – Make a copy of the current tool offset data.
9) TOOL WEAR	{ TOOL WEAR } – Display and alter the tool wear amount table for the different tools.
10) TOOL GEOMETRY	{ TOOL GEOMET } – Display and alter the tool geometry table.
11) ACTIVE OFFSET	{ ACTIVE OFFSET } – Change the currently active offset number without requiring the programming of a different offset number.
12) BACKUP OFFSET	{ BACKUP OFFSET } – Make a copy of the current tool offset data.
13) QUICK CHECK	{ QUICK CHECK } – Use the syntax and format checker or the { QUICK CHECK } with graphics function.
14) SYSTEM SUPORT	<ul style="list-style-type: none"> • { PRGRAM PARAM } – Display and change the tables for programmable zones 1 and 2, the single-digit feedrates, and the fixed-cycle operating parameters. • { AMP } – Change any of the online AMP features. • { DEVICE SETUP } – Display and change any of the communication parameters for the peripheral devices. • { MONI - TOR } – Display the current I/O configurations and the axis monitor for following error, distance to marker, etc. System suport does not protect hotkey access to the search monitor page.
15) PRGRAM PARAMETERS	{ PRGRAM PARAM } – Display and change the tables for programmable zones 1 and 2, the single-digit feedrates, and the fixed-cycle operating parameters.
16) ONLINE AMP	{ AMP } – Display and change the online adjustable machine parameters.
17) PARAMACRO PARAM	{ MACRO PARAM } – Display or change any of the values in the paramacro tables without using programming commands.
18) ACCESS CONTROL	{ ACCESS CONTRL } – Assign different functions to different access levels, change the current password, or view the functions assigned to the different access levels.
19) SWITCH LANGUAGE	{ SWITCH LANG. } – Change the current displays from one language to another.
20) POWER-UP LEVEL	When POWER-UP LEVEL is shown in reverse video, it indicates that if power is turned off when this level is active, this level automatically becomes active when power is turned back on. If this is not in reverse video, it indicates that the control defaults to level 8 access control at next power-up.
21) TIME PARTS	<ul style="list-style-type: none"> • { SET TIME } – Change or reset time parameters • { SET DATE } – Change or reset data parameters <p>When TIME PARTS is not in reverse video, the operator can only perform the following functions on the time and parts screen: RUN TIME, CYCLE TIME, and LOT SIZE.</p>
22) S/OEM MESSAGE	<ul style="list-style-type: none"> • { ENTER MESSAGE } – Enter a new message to be displayed on the control's power-up screen. • { STORE BACKUP } – Store an entered message for the power-up screen to backup memory.

Parameter Name:	Function becomes accessible when parameter name is in reverse video:
23) SCALING	When SCALING is not in reverse video, the operator still has access to the {SCALNG} softkey; however values on the screen may not be modified.
24) CHANGE DIRECTORY	Allows access to the protectable directory for file edit, direct execution selection, and encrypted output.
25) AUX COMM	Allows access to the {AUX COMM} softkey that sets Data Highway Communication Module Parameters.
26) SEARCH MONITOR	Allows access to the {SEARCH MONITOR} page that allows searching and monitoring of the control's PAL program. This parameter also protects hotkey access to the search monitor page.
27) PASS THRU ENABLE	Allows access to the {PT ENABLE} softkey that enables remote I/O pass thru of AMP, PAL, and other ODS features. The {PT ENABLE} softkey is only available when AMP is configured to disallow block transfers and the system contains remote I/O hardware that has been enabled through PAL.

2.7.3 Entering Passwords

When you power-up, only functions that are not protectable and functions that are assigned to access level 8 are available (provided that the active level when power was turned off was not assigned the POWER UP LEVEL feature). To access the functions that are assigned to a specific access level, you must enter the password that corresponds to that access level. To enter a password, follow these steps:

1. Press the {PASSWORD} softkey.
(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	PROGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

E- STOP

ENTER PASSWORD:

PROGRAM [INCH]	F	0. 000 MPPM
Z 00000. 000	S	0
R X 00000. 000	T	1
C 359. 99		

MEMORY MAN STOP

↑				ACCESS CONTRL		
---	--	--	--	------------------	--	--

2. Enter the password you want to activate by typing it in on the input line with the keys on the operator panel. The control displays * for the characters you entered. If you make an error entering the password, edit the input line as described on page 2-41.
3. When the password is correct, press the [TRANSMIT] key. The access level that the password is assigned to is made active, and the control enables all of the functions that are assigned to that access level.

2.8 Changing Operating Modes

The control provides 3 basic operation modes:

- manual (MAN or MANUAL)
- manual data input (MDI)
- automatic (AUTO)

You can select a mode by using <MODE SELECT> on the MTB panel, or using the {FRONT PANEL} softkey. This is configurable by your system installer. Both means of selection cannot be available. Details on using the {FRONT PANEL} softkey are given on page 2-15.

Depending on the current control status, a mode change request cannot be honored. Operating modes may not be changed if any of these are true:

- The control is in E-Stop.
- The control is in the cycle-suspend state. This results when a program is halted during the execution of a block.
- The control is executing a threading- or multiple-pass turning cycle.

Important: Your system installer may have written PAL to disable the use of the {FRONT PANEL} softkey to change modes. If this is the case, then changing modes can be performed by using only <MODE SELECT> on the MTB panel.

Manual mode

To operate the machine manually,

- select MAN or MANUAL under <MODE SELECT>

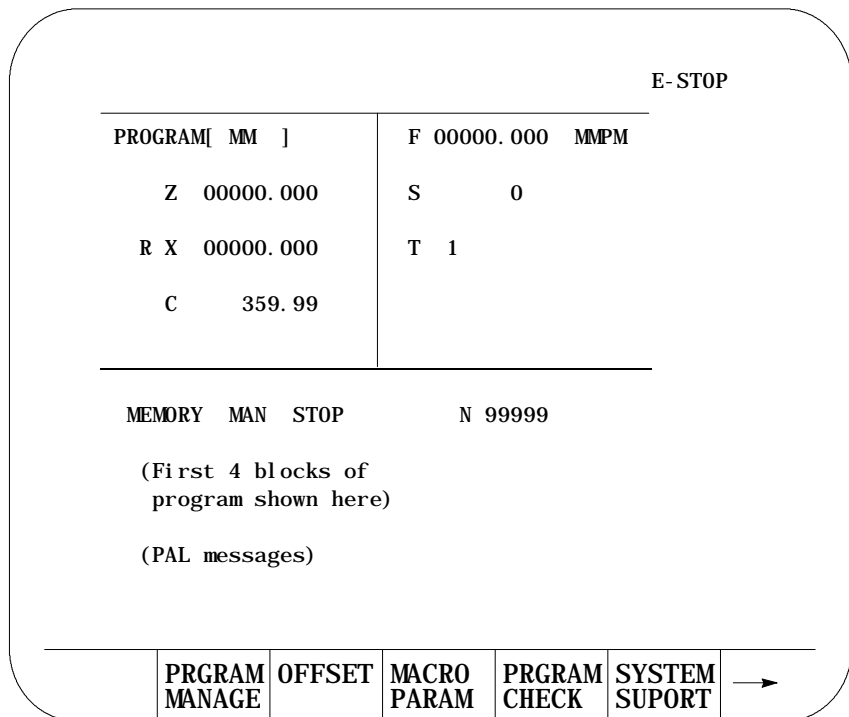
or

- press the {FRONT PANEL} softkey.

Use the left/right arrow keys to change the mode select options if using {FRONT PANEL}.

For details on Manual Mode operation, see chapter 4.

Figure 2.5
Manual Mode Screen



MDI mode

To operate the machine in MDI mode,

- select MDI under <MODE SELECT>

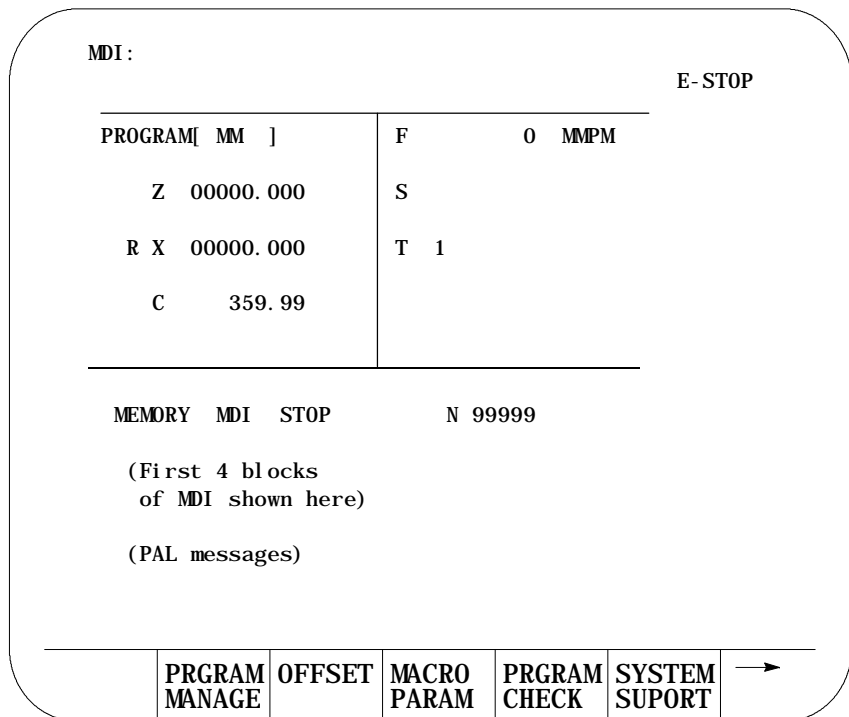
or

- press the {FRONT PANEL} softkey

Use left/right arrow keys to change mode select options if using {FRONT PANEL}.

For details on MDI operation, see page 4-11.

Figure 2.6
MDI Mode Screen



Automatic mode

To operate the machine automatically,

- select AUTO under <MODE SELECT>

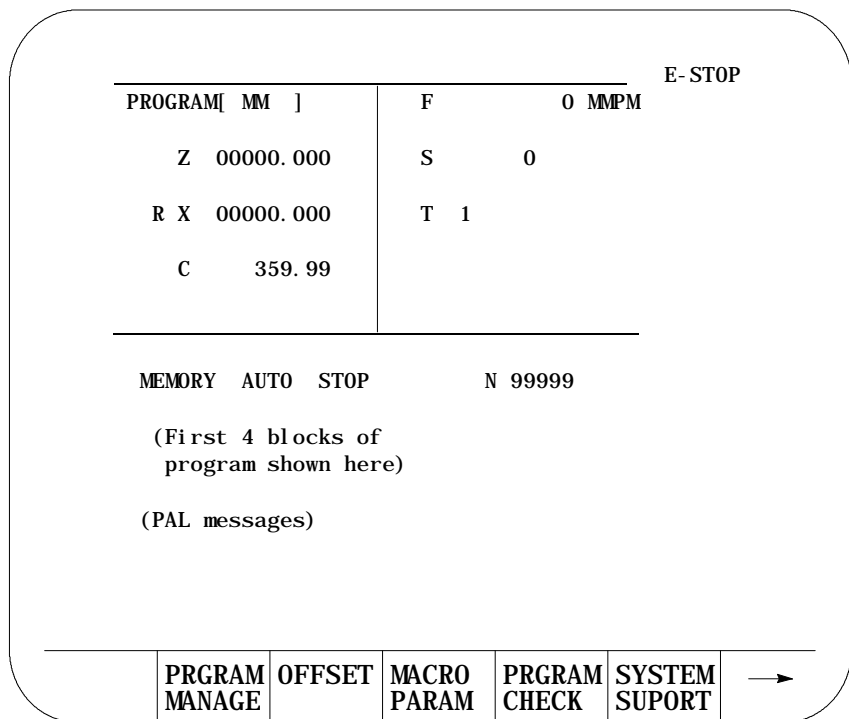
or

- press the {FRONT PANEL} softkey

Use left/right arrow keys to select mode options if using {FRONT PANEL}.

For details on automatic operation, see chapter 7.

Figure 2.7
Automatic Operation Screen



2.9
Displaying System and
Machine Messages

The control has two screens dedicated to displaying messages. The MESSAGE ACTIVE screen displays up to nine of the most current system messages and ten of the most current machine (logic generated) messages at a time. The MESSAGE LOG screen displays a log of up to 99 system messages and a separate log of up to 99 machine messages that occurred since the last time memory was cleared. For single process controls, this log displays up to 11 pages of messages (99 messages total) and up to six pages per process for dual process systems.

Important: The control automatically displays the highest priority, single, active message on all screens (other than the message screen) on line 1 of the CRT. If more than one message occurs with the same priority, the control displays the most recent message (provided no other message is active with a higher priority).

Use the MESSAGE ACTIVE screen to display all the messages that are currently active, or the MESSAGE LOG screen to display a log of the recorded messages. To access these message screens, follow these steps:

1. From the main menu press the continue {• } softkey to change the softkey functions.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
--	------------------	--------	----------------	-----------------	------------------	---

2. Press the {ERROR MESSAGE} softkey to enter the MESSAGE ACTIVE screen shown in Figure 2.8.

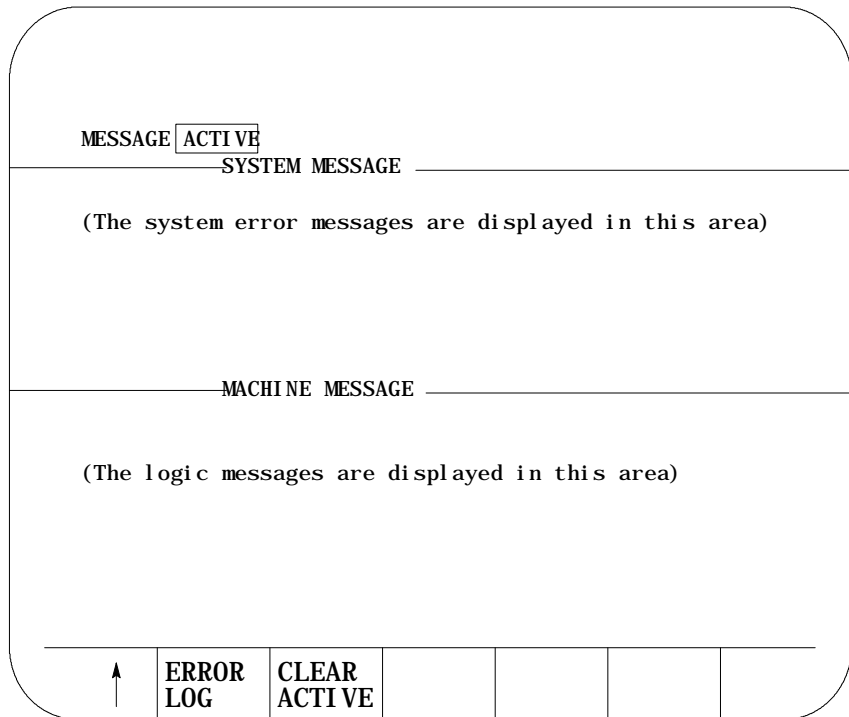
(softkey level 1)

	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→
--	----------------	------------------	---------------	-----------------	--	---

The control displays the currently active messages in sections dedicated to:

- system messages in the top half of the screen
- machine messages (logic generated) in the bottom half of the screen

Figure 2.8
Message Active Display Screen



This is the information displayed on the MESSAGE ACTIVE screen. The control displays up to 9 active system messages and up to 10 machine messages. The machine messages represent the currently active logic messages located on 13 lines through 22 of the 9/Series display screen.

Important: For a listing of system messages and a brief description, refer to appendix B. For a description of machine messages, refer to the documentation prepared by your system installer.

3. Press the {ERROR LOG} softkey to enter the MESSAGE LOG screen shown in Figure 2.9.

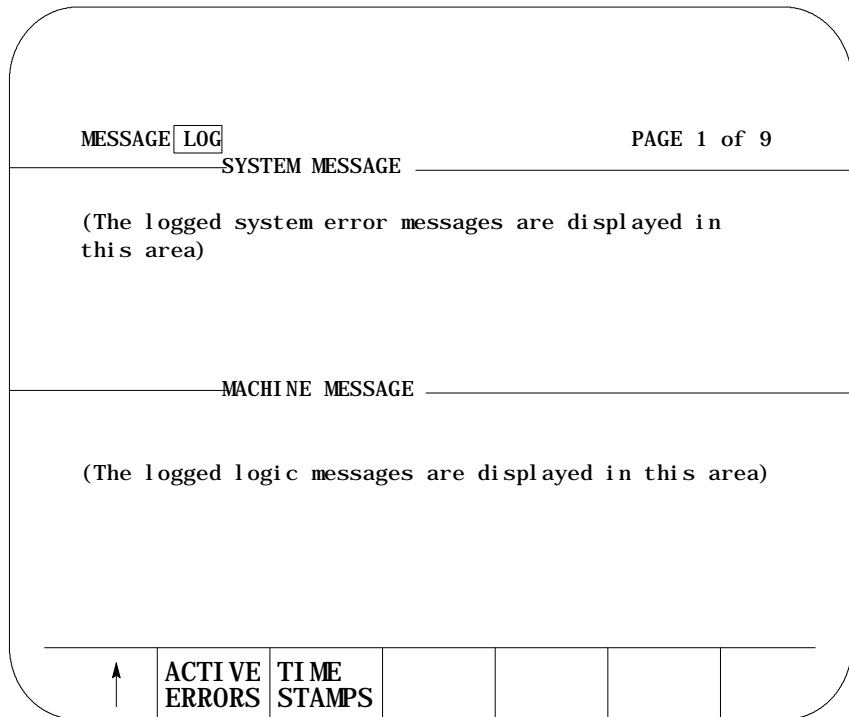
(softkey level 2)



The control displays the logged messages in sections dedicated to:

- system logged messages in the top half of the screen
- machine logged messages (logic generated) in the bottom half of the screen

Figure 2.9
Message Log Display Screen



This is the information displayed on the MESSAGE LOG screen. The control displays up to 99 system messages and up to 99 machine messages. These are the most recent system messages and machine messages that occurred on the control since memory was last cleared. To display more messages, press the [•] key while holding the [SHIFT] key. The next page of the logged messages displays.

Press the {TIME STAMPS} softkey to view the date and time of when each logged system message and machine message occurred. Pressing this softkey may cause part of the messages text to be overlaid by the date and time data. To display the full messages text, press the {FULL MESSAGE} softkey.

4. Press the {ACTIVE ERRORS} softkey to return to the MESSAGE ACTIVE screen.

(softkey level 2)



5. To return to softkey level 1 press the exit {•} softkey while holding the [SHIFT] key.

2.9.1
Clearing Active Messages
{CLEAR ACTIVE}

After the cause of a machine or system message has been resolved, some messages remain displayed on all screens until you clear them.



CAUTION: Not clearing the old messages from the screen can prevent messages that are generated later from being displayed. This occurs when the old resolved message has a higher priority than the newly generated message. The new message is still displayed on the message display screen as an active message, but does not appear in the message area of other screens.

Active messages are cleared from the screen in this way:

- Press the [CAN] key to clear the most recent active messages individually.
- Clear all active messages from the error message display screen by pressing the {CLEAR ACTIVE} softkey.

(softkey level 2)

↑	ERROR LOG	CLEAR ACTIVE				
---	--------------	-----------------	--	--	--	--

Important: Clearing active messages does not correct the problem that caused the error; it only clears the message from the active file.

2.10
The Input Cursor

The input cursor is the cursor located on lines 2 and 3 of the screen. It is available when you need to input data by using the operator panel (as needed in MDI mode, for example). The following section is a description of how to move the cursor and edit data on the input line by using the keys on the operator panel.

Cursor Operation:	Description:
Moving the input cursor	To move the cursor right and left in the input area, press and hold the [SHIFT] key while pressing the left and right cursor keys. To move the cursor between line 2 and 3 use the left and right cursor keys while holding the [SHIFT] key. The up and down cursor keys cannot be used. The input cursor wraps around when it reaches the end of the screen.
Inserting characters	To insert characters into existing data on the input lines, move the cursor as described above. Place the cursor just to the right of the location to insert data. Any data typed in by using the operator panel is inserted just to the left of the cursor.
Deleting Characters	To delete characters on the input lines move the cursor to the right of the character to delete. Pressing the [DEL] key deletes the character to the left of the cursor in the input line.
Deleting all characters on the input line	To delete all entered characters on the input lines press the [DEL] key while holding down the [SHIFT] key. All characters on the input line are deleted.
Sending information	To send information to the control from input line press the [TRANSMIT] key. All information on the input line is sent to the control.

2.11
{REFORM MEMORY}

It is sometimes necessary to perform a Reform Memory operation to clear memory. This is often necessary after one of the following occurs:

- the amount of RAM memory that can be used by PAL is changed in AMP
- a new PAL program has been sent to the control (downloading PAL does not always make it necessary to format)
- volatile memory has failed either from the super cap losing power or from the battery backup losing power.

You must format any time the control displays the error message "MEMORY CRASH - REFORMAT."

This operation can also be performed when you want to delete all part programs in control memory.

After a format operation is performed, it is not necessary to re-install AMP or PAL.



CAUTION: The {REFORM MEMORY} function erases all part programs that are stored in control memory.

To reformat control memory and delete all programs stored in memory, follow these steps:

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	PROGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {REFORM MEMORY} softkey.

(softkey level 2)

↑	ACTI VE PROGRAM	EDI T PROGRAM	RESTR T PROGRAM	DI SPLY PROGRAM	COPY PROGRAM	→
↑	DELETE PROGRAM	VERFY PROGRAM	PROGRAM COMENT	RENAME PROGRAM	I NPUT DEVI CE	→
↑	REFORM MEMORY	CHANGE DI R				→

3. Press the {REFORM YES} softkey. All programs that are stored in control memory are deleted. To abort the operation, press the {REFORM NO} softkey.

(softkey level 3)

↑		REFORM YES	REFORM NO			
---	--	---------------	--------------	--	--	--

It can take several seconds for the control to complete the operation. During this period, the softkeys on the operator panel are rendered inoperative.

2.12
Removing an Axis
(Axis Detach)

This feature allows the removal of a rotary table or other axis attachment from a machine. When activated, the control ignores messages that may occur resulting from the loss of feedback from a removed axis such as servo errors, etc.

Important: This feature removes the selected axis from the control as an active axis. Any attempt to move the removed axis results in an error. This means that part programs that use the removed axis name cannot be executed. Jog moves and MDI commands that attempt to move the removed axis also result in an error.

This feature can be enabled in AMP. The axis must be selected as “Detached” to be considered removed. Refer to the documentation supplied by your system installer for the necessary steps involved in detaching an axis or physically removing axis hardware from your machine.

2.13
Time Parts Count
Display Feature

The Time Parts Count Display logs data relevant to part program execution such as number of workpieces cut, cycle time, lot size, and remaining workpieces. This data can be displayed and altered through the Time Parts screen.

There are 3 levels of access to the Time Parts screen. They are listed below in order of most restrictive to least restrictive. Refer to page 2-25 for details on password protection and access control.

Access:	Protection:
No	Restricts operator from Time Parts screen entirely (softkey {TIME PARTS} not accessible). Accomplished by denying access to “Active Program.”
Operator	Restricts operator from setting “Date” or “Time” (softkeys {SET TIME} and {SET DATE} do not appear). Also restricts operator from setting “Power-on time/overall” and “Workpieces cut/overall.” Accomplished by denying access to “Time Parts.”
Supervisor	Full access to all features of the Time Parts screen.

To access the Time Parts screen, follow these steps:

1. Press the {PROGRAM MANAGE} softkey.
(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	PROGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {ACTIVE PROGRAM} softkey.

(softkey level 2)

↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTR PROGRAM	DISPLY PROGRAM	COPY PROGRAM	→
↑	DELETE PROGRAM	VERFY PROGRAM	PRGRAM COMENT	RENAME PROGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY	CHANGE DIR				→

3. Press the {TIME PARTS} softkey. This generates the screen shown in Figure 2.10.

(softkey level 3)

↑	DE- ACT PROGRAM	SEARCH	MID ST PROGRAM	T PATH GRAPH	T PATH DISABL	→
↑	SEQ STOP				TIME PARTS	→

You see the Time Parts screen:

Figure 2.10
Time Parts Screen

PROGRAM XXXXXXXX	DATE MM/DD/YY	TIME HH: MM: SS				
POWER- ON TIME/OVERALL	:	99999: 59: 59				
WORKPIECES CUT/OVERALL	:	9999999999				
RUN TIME	:	99999: 59: 59				
POWER- ON TIME/AFTER RESET	:	99999: 59: 59				
CYCLE TIME	:	99999: 59: 59				
WORKPIECES CUT/AFTER RESET	:	9999999999				
LOT SIZE	:	9999999999				
REMAINING WORKPIECES	:	9999999999				
LOT COMPLETION FLAG	:	0				
(PAL messages)						
↑	SET TIME	SET DATE	ED PRT INFO			

Important: Some softkeys shown in Figure 2.10 might not appear on your system due to restricted access. Refer to the beginning of this section and page 2-25 for details.

You can modify the values on this screen. Press the {ED PRT INFO}, {SET DATE}, or the {SET TIME} softkeys as explained in the Screen Field Definitions that follow.

Press the exit softkey {•} to save changes and return to the “Active Program” screen.

Time Part Screen Field Definitions

Program - is the currently active part program, displayed automatically by the control.

Date - is the current date setting. To change this setting:

1. Press the {SET DATE} softkey, provided that you have supervisor-level access.

You are prompted for a new date with a line that displays the current date setting.

2. Press the [DEL] key to erase the characters displayed.
3. Type in the new date.

You must use one of these formats:

Format:	Language:
Month / Day / Year	English
Year / Month / Day	Japanese
Day / Month / Year	German

Time - is the current 24-hour time setting. It can be changed by pressing the {SET TIME} softkey, provided that you have supervisor-level access. You are prompted for a new time with a line that displays the current time setting. Change the current setting by pressing the [DEL] key to erase the characters displayed and then typing in the new time. You must enter using the format Hours:Minutes:Seconds. For example, 2:00 pm should be entered as 14:00:00.

Power-on Time/Overall - indicates the total accumulated time that the control has been ON. This value is saved in backup memory each time the control is powered off, so it is restored at its previous value each time the control is turned ON. To clear this field to zero:

1. Press the {ED PRT INFO} softkey, provided that you have supervisor-level access.
2. Press the up or down cursor keys to move to this field or the next field without changing the current value.
3. Enter a Y at the prompt for this field.
4. Press [TRANSMIT] to clear the current value.

Workpieces Cut/Overall - indicates the total number of part programs executed to completion by the control. Use this field to determine the need for periodic checkups or as a statement of warranty. This counter is incremented by one each time the control encounters an M02, M30, or an M99 in a main part program (M99 in a subprogram does not increment this counter, though M02 or M30 does). To clear this field to zero:

1. Press the {ED PRT INFO} softkey, provided that you have supervisor-level access.
2. Press the up or down cursor keys to move to this field or the next field without changing the current value.
3. Enter a Y at the prompt for this field.
4. Press [TRANSMIT] to clear the current value.

Run Time - indicates the total accumulated time that part programs were executing with the control in automatic mode. Use this field with "Power-on Time/After Reset" to estimate the utilization ratio of the machine. To clear this field to zero:

1. Press the {ED PRT INFO} softkey if you have either operator-level or supervisor-level access.
2. Press the up or down cursor keys to move to this field or the next field without changing the current value.
3. Enter a Y at the prompt for this field.
4. Press [TRANSMIT] to clear the current value.

Power-on Time/After Reset - indicates the total accumulated time that the control has been ON. This value is saved in backup memory each time the control is powered off, so it is restored at its previous value each time the control is turned ON. Use this field with "Run Time" to estimate the utilization ratio of the machine. The value for this field is cleared to zero when the "Run Time" field is cleared to zero; it cannot be changed independently.

Cycle Time - indicates the elapsed execution time for each individual part program. Cycle time begins counting when the cycle-start button is pressed and ends when an M02 reset or M30 is encountered. To reset this field to zero, use one of three methods:

- press the cycle-start button to initiate program execution
- turn off the control power
- follow these steps:
 1. Press the {ED PRT INFO} softkey if you have either operator-level or supervisor-level access.
 2. Press the up or down cursor keys to move to this field or the next field without changing the current value.
 3. Enter a Y at the prompt for this field.
 4. Press [TRANSMIT] to clear the current value.

Workpieces Cut/After Reset - indicates the total number of part programs executed to completion by the control since the last time “Run Time” was reset. This counter is incremented by one each time the control encounters an M02, M30 or an M99 in a main part program (M99 in a subprogram does not increment this counter, though M02 or M30 does). The value for this field is cleared to zero when the “Run Time” field is cleared to zero; it cannot be changed independently.

Lot Size - is the number of times you need to execute this particular part program. To enter a new number:

1. Press the {ED PRT INFO} softkey if you have either operator-level or supervisor-level access.
2. Press the up or down cursor keys to move to this field or the next field without changing the current value.
3. Enter a numeric value at the prompt for this field.
4. Press [TRANSMIT] to change the current value.

Remaining Workpieces - indicates the number of workpieces that still need to be cut in the lot. The value for this field is automatically set equal to the lot size each time the "Lot Size" value is changed. When the control encounters an M02, M30, or M99 in a main part program, the remaining workpieces field is decremented by one. The control tells the system installers PAL program when the lot remaining size is zero. At this point, press <CYCLE START> to automatically set the field back to the "Lot Size" value. Complete operation of this feature is somewhat PAL dependant. Refer to the documentation supplied by your system installer.

Lot Completion Flag - is automatically set to zero by the control whenever a non-zero value is entered for "Lot Size." It is set to one when the "Remaining Workpieces" field reaches zero. It is again reset to zero when the next cycle start occurs after the remaining workpieces field has reached zero. Complete operation of this feature is somewhat PAL dependant. See the documentation supplied by your system installer.

Press the exit softkey {• } to save changes and return to the "Active Program" screen.

END OF CHAPTER

Offset Tables and Setup

3.0 Chapter Overview

In this chapter we describe the basics of job setup. Major topics include how to:

- use the offset table
- set and display offset data
- set and display work coordinate systems
- set and display communication parameters

3.1 Tool Offset Table {TOOL GEOMET} and {TOOL WEAR}

The offset tables are broken in to two major tables: the tool geometry offset table and the wear offset table. Use the tool geometry offset table to enter measured values that compensate for differences in tool mounting and differences in tool dimensions. Use the wear offset table to enter measured values that compensate for the amount of wear on a tool under normal use. “Wear” means a non-reversible tool dimensional change.

This data can be entered into the offset tables:

- Tool length offset data (TOOL GEOMETRY and TOOL WEAR.)
- Cutter radius data (TOOL GEOMETRY and TOOL WEAR)

Parameters for the resolution of the offset data are determined by the system installer in AMP. For more AMP information see your:

- 9/Series CNC AMP Reference Manual
- Offline Development System (ODS) Software User Manual

Any setting smaller than the minimum unit set for axis motion is not acceptable as offset data. The minimum value available to the system installer is 0.00001 mm (0.000001 inch) with a maximum value of 99999999 (8 digits).

The D-word in a program will call values from the offset tables corresponding to the tools diameter. An H-word in a program will call values from the offset tables for tool length. When the H- or D-word is programmed the control pulls data from both the tool the geometry table and the tool wear table. The value from the tool wear table is subtracted from the value from the tool geometry table. The result is used by the control as tool length offset or cutter compensation data.

For details on using tool offsets see chapter 20. For information on cutter compensation see chapter 21.

Figure 3.1
Offset Table Screen for Wear

TOOL OFFSET NUMBER:						
TOOL	WEAR TABLE	X	PAGE 1 OF 4			
NO.	LENGTH	(DIAMETER)				
1	.5321	.0234	[INCH]			
2	.4421	.0142	[INCH]			
3	.0243	.0888	[INCH]			
4	.0156	.0791	[INCH]			
5	.0265	.0532	[INCH]			
6	.081	.043	[MM]			
7	.032	.022	[MM]			
8	.0000	.0000	[INCH]			
9	.0000	.0000	[INCH]			
10	.0000	.0000	[INCH]			
11	.0000	.0000	[INCH]			
12	.0000	.0000	[INCH]			
13	.0000	.0000	[INCH]			

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTI VE OFFSET	MORE OFFSET	→
---	------------------	-----------------	-----------------	-------------------	----------------	---

3.1.1 Tool Offset Dimensional Parameters

Tool Offset Numbers (Geometry and Wear Table)

Tool offset numbers are called out in a part program through use of D- and H-words. D- and H-words specify a one, two, or three digit offset number. The control then accesses the value assigned to that offset number in the offset table. The offset number is in the far left column on the offset screen. Offset number "00" is not a valid offset number to enter data under, but can be used to cancel tool offsets.

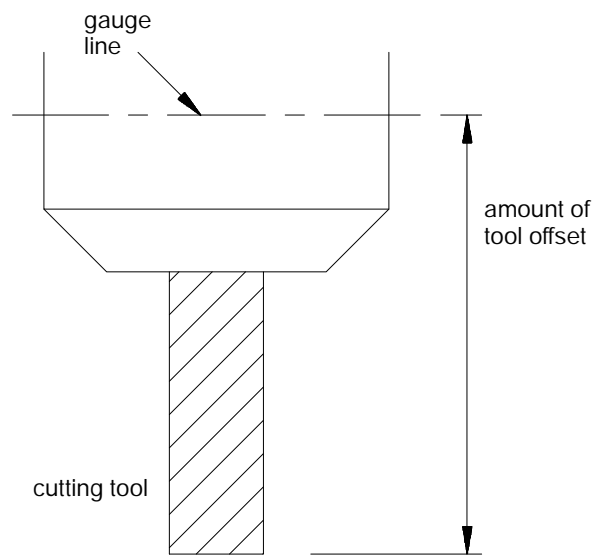
For more on calling offset numbers refer to chapter 20.

Tool Length Offset Data (Geometry Table)

The tool length offset function is used to compensate for the difference between the tool position (or tool length) as mounted in the spindle and the tool length assumed in writing a part program. By using the tool length offset function, a programmer can write a part program without further concern for tool mounting.

The system installer determines in AMP which axis (or axes) are used by the control as the tool length axis. Refer to documentation prepared by the system installer for details on what axes have been selected for the tool length axis. This manual assumes that the Z axis is used as the tool length axis.

Figure 3.2
Tool Length Offset



The term “gauge line” is used to define the precise point on the spindle or tool holder from which all programmed tool paths originate. Offsets refer to the distance from this gauge line to the end of the tool that contacts the part being cut.

For tool length offset data, measure the value for the Z axis from the gauge line to the end of the cutting tool. The values may be entered as either positive or negative values depending on which G-code is used to activate the offset (G43 or G44).

Important: For a typical end milling machine (with tool/chuck configuration similar to Figure 3.2), keep the following in mind. When programming tool offsets with a G43, the geometry offset value should be positive and the wear offset data should be negative when entered in the offset tables. When programming tool length offsets with a G44, the geometry offset value should be negative and the wear offset data should be positive when entered into the offset tables.

Tool Diameter Compensation Data (Geometry Table)

To cut a workpiece using the side face of the cutting tool, it is more convenient to write the part program so that the center of the tool moves along the shape of the workpiece. Since all cutting tools have a diameter, a program written for moving the center of the tool must somehow “compensate” for the tool’s radius.

The system installer determines if radius or diameter values are entered in the offset table. This manual assumes that the system installer requires diameter values to be entered.

The control can compensate for this difference using the cutter compensation feature discussed in chapter 21.

Cutter compensation require that the diameter of the cutting tool be entered. Call the tool diameter value from the offset tables by programming a D-word in a part program. Note that the control will automatically convert the tool diameter to a radius value when necessary.

Tool Length Wear Data (Wear Table)

The tool length wear feature takes into account the wear (change in length) that the end of a tool will incur from normal usage. Enter a value in the wear table that is equal to the difference in tool length as entered in the geometry table and the actual tool length. The value entered as tool length wear is subtracted from the current value for tool length taken from the geometry offset table when an H-word is designated in a part program.

This feature allows the compensation for slight changes in tool length without the need to change the tool geometry table. Then when a worn tool is replaced with an identical tool the wear offset needs to be reset to zero. The tool geometry value should never need to be altered once the initial value is entered as long as identical tools are always used when replacing tools.

Important: For a typical end milling machine (with tool/chuck configuration similar to Figure 3.2), keep the following in mind. When programming tool offsets with a G43, the geometry offset value should be positive and the wear offset data should be negative when entered in the offset tables. When programming tool length offsets with a G44, the geometry offset value should be negative and the wear offset data should be positive when entered into the offset tables.

Tool Diameter Wear Compensation Data (Wear Table)

The tool diameter wear compensation feature takes into account the wear that a tool diameter will incur from normal usage. Enter a value in the wear table that is equal to the difference in tool diameter as entered in the geometry table and the actual tool diameter. The value entered as tool diameter wear is subtracted from the current value for the tool diameter taken from the geometry offset table when a D-word is designated in a part program.

The system installer determines if radius or diameter values are entered in the offset table. This manual assumes that the system installer requires diameter values to be entered.

This feature allows the compensation for slight changes in tool diameter without the need to change the tool geometry table. Then when a worn tool is replaced with an identical tool the wear offset needs to be reset to zero. The tool geometry value should never need to be altered once the initial value is entered as long as identical tools are always used when replacing tools.

3.1.2 Setting Tool Offset Tables

There are six methods for modifying tool offset tables. These are discussed in the following chapters and sections:

- Using {MEASURE} (section 3.1.3)
- Programming G10s (chapter 20)
- Skip functions using a probe (chapter 27)
- Setting Paramacro System Parameters (chapter 28)
- Altering through the PAL program

When PAL is used to modify either the work coordinate system tables or the tool offset tables, cutter compensation should not be active (G40 mode). If cutter compensation is active, be aware that the new offset will not be placed in part program set-up buffers that have already been read into control memory. This will result in the offset not being activated until several program blocks after the current block. The number of setup buffers is dependent on the number of block retrace steps configured in AMP and what software features are currently being used.

The sixth method, and the one discussed here, lets you directly key in to the offset table offset data that is manually measured.

Important: In order for newly modified tool offsets to become immediately active, cutter compensation must be off (G40 mode). If it is on (G41/G42 mode), the control generates the error message “CHANGE NOT MADE IN BUFFERED BLOCKS”. This indicates that the control is still using the old offset values and must first run several program blocks before using the new offsets values. The new offsets may then be activated too late for your particular application.

To manually display or alter the offset tables follow the directions below:

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Display either the tool geometry offsets or the tool wear offsets.

To display the geometry offsets (tool length offsets and the tool diameter data), press the {TOOL GEOMET} softkey. An example of a tool offset geometry screen is shown in Figure 3.3.

To display the wear offsets (tool length and diameter wear data), press the {TOOL WEAR} softkey. An example of a tool offset wear screen is shown in Figure 3.4.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

Figure 3.3
Tool Offset (Geometry) Screen

TOOL OFFSET NUMBER:						
TOOL GEOMETRY TABLE			X	PAGE 1 OF 4		
NO.	LENGTH	(DIAMETER)				
1	1.6396	1.6000	[INCH]			
2	1.4537	.8000	[INCH]			
3	.6312	.9000	[INCH]			
4	5.7931	.5000	[INCH]			
5	7.8432	.6000	[INCH]			
6	0.000	.000	[MM]			
7	0.000	.000	[MM]			
8	0.0000	.0000	[INCH]			
9	0.0000	.0000	[INCH]			
10	0.0000	.0000	[INCH]			
11	0.0000	.0000	[INCH]			
12	0.0000	.0000	[INCH]			
13	0.0000	.0000	[INCH]			

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
---	---------------	--------------	--------------	---------------	-------------	---

3. Move the cursor to the offset data to be modified. Use the up, down, left, or right cursor keys to move the cursor to the tool offset data on the current page. Press the {MORE OFFSET} softkey to change pages. The tool offset data located at the cursor will be shown in reverse video.

4. Select the units using {INCH/METRIC}

To select units of “mm” or “inch” for the offset data, press the {INCH/METRIC} softkey. The units used for the currently selected offset number will change each time the softkey is pressed.

When the units are altered, all data previously entered is converted to the newly selected units (Inch or Metric) for that offset number.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
---	---------------	--------------	--------------	---------------	-------------	---

↑	MEASURE	INCH/METRIC	NEXT AXIS	PREV AXIS	COPY OFFSET	→
---	---------	-------------	-----------	-----------	-------------	---

Figure 3.4
Tool Offset (TOOL WEAR) Screen

TOOL OFFSET NUMBER:						
TOOL WEAR TABLE X				PAGE 1 OF 4		
NO.	LENGTH	(DIAMETER)				
1	.5321	.0234	[INCH]			
2	.4421	.0142	[INCH]			
3	.0243	.0888	[INCH]			
4	.0156	.0791	[INCH]			
5	.0265	.0532	[INCH]			
6	.081	.043	[MM]			
7	.032	.022	[MM]			
8	.0000	.0000	[INCH]			
9	.0000	.0000	[INCH]			
10	.0000	.0000	[INCH]			
11	.0000	.0000	[INCH]			
12	.0000	.0000	[INCH]			
13	.0000	.0000	[INCH]			

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
---	---------------	--------------	--------------	---------------	-------------	---

5. Replace or add data as follows:

- To replace stored offset data with new data, key-in the new data, and press the {REPLCE VALUE} softkey.
- To add to previously stored offset data, key-in the amount to be added, and press the {ADD TO VALUE} softkey.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
↑	MEASURE	INCH/METRIC	NEXT AXIS	PREV AXIS	COPY OFFSET	→

6. Change to the next or previous axis by pressing {NEXT AXIS} or {PREVIOUS AXIS}.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
↑	MEASURE	INCH/METRIC	NEXT AXIS	PREV AXIS	COPY OFFSET	→

3.1.3 Setting Offset Data Using {MEASURE}

The measure feature offers an easier method of establishing tool offsets. The control, not the user, computes the tool length offsets and enters the value into the tool offset table. Note the measure feature is used to measure tool length offset values for the wear or geometry tables. It is typically not very effective at measuring tool diameters unless special attention is paid to tool orientation. To use the measure feature, follow these steps:

1. Establish a fixed machine position without a tool in the chuck. This position may be any fixed, non-movable location on the machine that the tool may be jogged against consistently, using a variety of different tools.
 - If entering a tool length in the geometry offset table, jog the machine gauge line (on the axis being updated) to this position. No tool offsets should be active and no tool should be in the chuck. The value of this position, located in the work coordinate system, must be recorded. The user keys in this value in steps 6 and 7.
 - If entering a tool length wear in the wear offset table, jog the machine gauge line to the fixed position. No tool offsets should be active, and no tool should be in the chuck. The value of this position, located in the work coordinate system, must be recorded. Add the original tool length offset from the tool geometry table to the fixed machine location. The user keys in this value in step 7.
2. Access the tool geometry or wear offset table as discussed in section 3.1.2.
3. Cursor down to the offset that is to be changed. Note that the offset can be displayed in either inch or metric measurements.
4. Load the tool that is to be measured into the chuck.
5. Using incremental, continuous or handwheel mode, jog the tool tip to the fixed location determined in step 1.
6. Press the {MEASURE} softkey.
7. Key in the coordinate value of the fixed location determined in step 1.
8. Press the [TRANSMIT] key.

The control now subtracts the keyed in position from the current axis position and enters this difference as the offset value into the table.

3.1.4 Tool Offset Range Verification

Tool offset range verification checks:

- the maximum values entering the tool offset tables
- the maximum change that can occur in either table

To use tool offset range verification, follow this softkey sequence:

9. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

10. Press the {AMP} softkey.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVI CE SETUP	MONI - TOR	TI ME PARTS	→
↑	PTOM SI /OEM		SYSTEM TI MI NG			→

11. Press the {AXIS PARAM} softkey.

(softkey level 3)

↑	AXIS PARAM	PATCH AMP	UPDATE BACKUP	UPLD/ DWNLD	BACKUP AMP	→
---	---------------	--------------	------------------	----------------	---------------	---

12. Press the {RANGE PARAM} softkey.

(softkey level 4)

↑	SPNDL PARAM	SERVO PARAM	AXIS CALI B	HOME CALI B	REVERS ERROR	→
↑	RANGE PARAM					→

Your system installer initially sets these values in AMP. You can modify them with online AMP by using this screen:

OFFSET RANGE VERIFICATION

		[inch]
Per table values	}	MAXIMUM +/- WEAR RADIUS .12345
		MAXIMUM +/- GEOM RADIUS 10.12345
		[inch]
Per axis values	}	MAXIMUM WEAR OFFSET CHANGE .01000
		MAXIMUM GEOM OFFSET CHANGE 1.00000
		MAXIMUM +/- WEAR OFFSET .10000
		MAXIMUM +/- GEOM OFFSET 10.00000

softkey level 5
↑
REPLCE VALUE
ADD TO VALUE
NEXT AXIS
UPDATE & EXIT
QUIT
→

About the Offset Range Verification Screen

- **display format is fixed**

Mode	Places to the left of the decimal point	Places to the right of the decimal point
inch	3	5
metric	4	5

- **data entry is bounded by the programming resolution of the axes**

When Does Verification Occur

Verification occurs when a value enters the table from:

- data entry screens
- PAL
- paramacros

Important: The control does not perform the verification if the value, old or new, is zero, nor does it check G10 data-setting codes.

Verify for Maximum Value

This value represents the absolute maximum value per table for all tool offsets in that table.

If you enter:	Then:
a positive number greater than the maximum value	the control generates the error message: "OFFSET EXCEEDS MAX VALUE"
a negative number less than the negative of the maximum value	The control does not modify the value in the table.

Verify for Maximum Change

This change represents the amount an offset may change from its current value. If you exceed the amount set by the system installer in AMP, the change is not allowed. The control generates the error message "OFFSET EXCEEDS MAX CHANGE."

3.2 Changing the Active Tool Offset {ACTIVE OFFSET}

Use this feature to allow the manual activation of tool offsets without the need to program a D- or H-word to call the corresponding offset number. This may be necessary when a broken tool has been replaced using the Jog Retract feature, or if a program is to start execution with a tool active in the chuck and no tool offsets programmed, etc.

Important: The control must be in either cycle stop or E-Stop states before an attempt is made to change the active offset using this method.

The axis that is selected as the length axis (the axis that length offsets are applied to) is shown in reverse video. The length axis is selected in AMP or through programming as discussed on page 20-9.

If it is necessary to change the current tool offset values or to activate tool offset numbers without programming an H- or D-word, follow these steps:

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

- Press the {TOOL GEOMET} or the {TOOL WEAR} softkey. It does not matter which softkey is pressed. Any changes made to the active offset number on the tool geometry screen also activates the same offset number on the tool wear screen as well and vice versa.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

The tool offset table is displayed. Currently active offset values (if any) are indicated with an * to the left of the offset value.

- Move the cursor on the offset table until the desired offset is shown in reverse video. If it is desired to activate a length offset (normally programmed with an H-word), make sure the selected offset value is in the "LENGTH" column. If it is desired to activate a diameter or radius offset (normally programmed with a D-word), make sure the selected offset value is in the "DIAMETER" column. Only one length offset and one diameter offset may be active at the same time.
- Press the {ACTIVE OFFSET} softkey when the desired offset is selected. The offset will be made active provided that that offset mode is active on the control G41 or G42 for tool diameter offsets and G43 or G44 for tool length offsets. See chapter 20 and 21 for details on programming these tool offsets.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
↑	MEAS- URE	INCH/ METRIC	NEXT AXIS	PREV AXIS	COPY OFFSET	→

If the tool length offset is currently active (G43 or G44), then the new tool length offset will go into effect immediately (the coordinate system is shifted). The next time cycle start is pressed, the control will generate and execute a block that will move the cutting tool back to the coordinate location in the work coordinate system that it was at before the shift to the coordinate system took place. If the control is currently in G49 mode, then only the H-word is activated until a G43 or G44 is programmed.

If cutter compensation is active (G41 or G42) and the control is not currently in MDI mode, then the new radius is activated as discussed in chapter 21. If G40 is currently active, then only the D-word is activated until a G41 or G42 is programmed.

3.3 Work Coordinate System Offset Tables {WORK CO-ORD}

There are two types of data that are entered in the work coordinate system table. One is the initial work coordinate system zero point locations that are called when programming G54-G59.3. The other is the external offset, used to offset all of the G54-G59.3 zero points to make the same set of work coordinate systems fit a variety of applications.

Zero Point Parameters

The work coordinate system parameters refer to the zero point locations of all of the work coordinate systems called out by G54-G59.3. Enter positions for these zero points as machine coordinate values. The specified machine coordinate position is then used by the control as the work coordinate system zero point.

Enter a machine coordinate system position for each of the work coordinate systems as described in section 3.3.3.

External Offset

The external offset is used to modify all of the work coordinate system zero points. Use of the external offset is optional. The value entered here will offset all of the work coordinate systems by the specified amount. Enter external offsets in the work coordinate system tables as the external offset value.

This offset is used to allow a programmer to use the same set of work coordinate system values in a variety of applications. Adjusting this value, for example, will allow for use of the same work coordinate systems and programs after a different part or tool mounting fixture has been installed on the machine. It can also be used to offset all work coordinate systems when part programs are transferred from different machines with different mechanical features.

3.3.1 Setting Work Coordinate System Tables

There are four methods for modifying work coordinate values. Three methods are discussed in the following chapters:

- Programming G10s (chapter 11)
- Setting paramacro system parameters (chapter 28)
- Modify offsets through PAL (see the system installer’s documentation)

The fourth method, and the one discussed in this section, lets you modify the work coordinate values immediately by using the keyboard.

Important: In order for newly modified work coordinate offsets to become immediately active, cutter compensation must be off (G40 mode). If it is on (G41/G42 mode), the control generates the error message “CHANGE NOT MADE IN BUFFERED BLOCKS”. This indicates that the control is still using the old offset values and must first run several program blocks before using the new offsets values. It is possible, therefore, that the new offsets may be activated too late for your particular application.

To display or change the initial setups for the work coordinate system and external offset follow these steps.

1. Press the {OFFSET} softkey on the main menu screen.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the {WORK CO-ORD} softkey to display the offset values for the work coordinate systems and the external offset. See Figure 3.5.

(softkey level 2)

	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
	COORD ROTATE	BACKUP OFFSET	SCALNG			→

Figure 3.5
Work Coordinate System Setting

WORK COORDINATE TABLES						
G54 [INCH]		G55 [MM]		G56 [MM]		
X	-9999.9999	X	-9999.9999	X	-9999.9999	
Y	-9999.9999	Y	-9999.9999	Y	-9999.9999	
Z	-9999.9999	Z	-9999.9999	Z	-9999.9999	
U	-9999.9999	U	-9999.9999	U	-9999.9999	
G57 [INCH]		G58 [MM]		G59 [MM]		
X	-9999.9999	X	-9999.9999	X	-9999.9999	
Y	-9999.9999	Y	-9999.9999	Y	-9999.9999	
Z	-9999.9999	Z	-9999.9999	Z	-9999.9999	
U	-9999.9999	U	-9999.9999	U	-9999.9999	
↑	REPLCE VALUE	ADD TO VALUE	INCH/ METRIC	RADI/ DIAM	MORE OFFSET	→

3. Move the cursor to the offset data to be modified. Use the up, down, left, or right cursor keys to move the block cursor to the offset data on the current page. Press the {MORE OFFSET} softkey to change pages. The selected item will be shown in reversed video.

Important: To modify the active work coordinate system, the control must be in ESTOP, or CYCLE-STOP and END OF BLOCK. If it is not, the control, along with generating the error message “ACTIVE OFFSET CAN NOT CHANGE”, will not update the work coordinate table.

4. Units selection {INCH/METRIC}

To select units of “mm” or “inch”, press the {INCH/METRIC} softkey.

(softkey level 3)

↑	REPLCE VALUE	ADD TO VALUE	INCH/ METRIC	RADI/ DIAM	MORE OFFSET	
---	-----------------	-----------------	-----------------	---------------	----------------	--

When the units are altered, all data previously entered for a particular coordinate system or offset is converted to the newly selected units (Inch or Metric). The current units are displayed to the right of the work coordinate system or the offset that is being changed.

Data can be replaced or added to as follows:

- To replace stored data with new data, key-in the new data and press the {REPLCE VALUE} softkey.
- To add to previously stored data, key-in the amount to be added and press the {ADD TO VALUE} softkey.

(softkey level 3)

↑	REPLCE VALUE	ADD TO VALUE	INCH/ METRIC	MORE OFFSET
---	-----------------	-----------------	-----------------	----------------

5. Replace or add data.

3.4
Backing Up Offset Tables

The control is capable of saving all of the information that is entered in the tool offset tables and the work coordinate system tables as a backup. This is done by the control generating a program consisting of G10 blocks. These G10 blocks contain the offset numbers and their respective wear and geometry values. Any time that this program is run, the set of values contained in these G10 blocks replace the current values in the offset tables. The G10 program can be saved in either control memory or it can be sent to a peripheral device.

The backup format includes a G43.1 block with the axis name of the currently active tool length offset axis. The axis name replaces the R-word in the L10/L11 blocks for the non-AMPed tool length offset axes.

This feature is very useful if the same tool or coordinate system offsets are to be used on different machines. The same offset tables can be easily set up by running this G10 program on other machines.

The offset table program can be saved in either control memory as a program or it can be sent to a peripheral device. This is very useful if the same tools are to be used on different machines. The offset tables can be easily set up by executing the same offset table program on all the other machines.

Important: Once the control begins executing a G10 program that has been previously generated, it will clear any data that already exists in the offset table being updated by that G10 command. This makes it impossible for a G10 block to simply add a few offset values. A G10 program must load the entire offset table each time it is run. Note that tool geometry and tool wear tables are separate offset tables. Loading data into one does not clear the other.

To backup the offset tables follow the directions below:

1. Press the {OFFSET} softkey.

(softkey level 1)

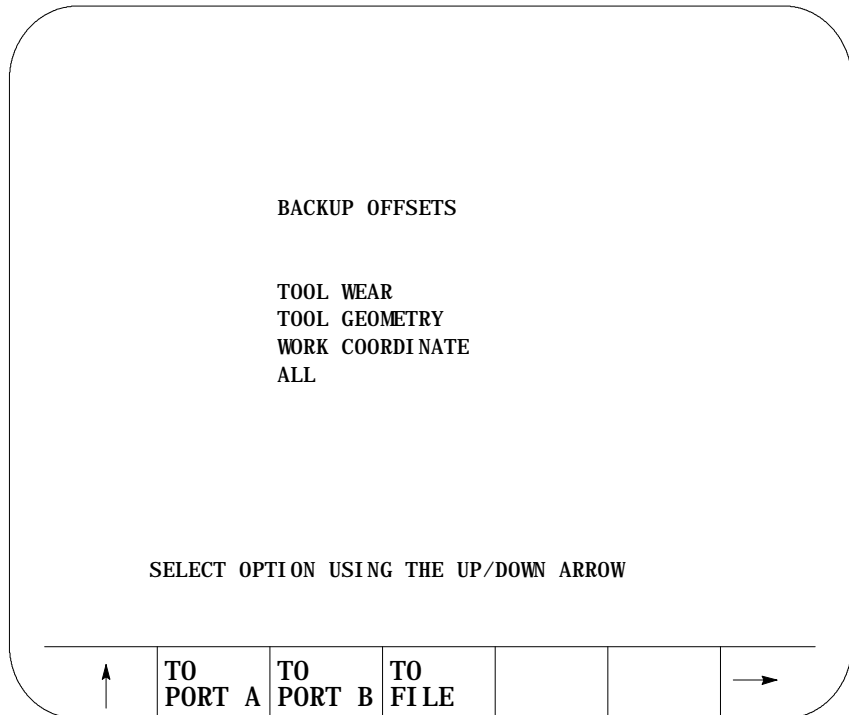
	PRGRAM MANAGE	FRONT PANEL	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	OFFSET	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {BACKUP OFFSET} softkey. This softkey backs up the currently active tool length offset axis. If you're using offsets on more than one axis, each axis must be selected and backed up separately. The backup offset screen shown in Figure 3.6 is displayed.

(softkey level 2)

↑	WORK CO- ORD	TOOL WEAR	TOOL GEOMET	TOOL MANGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

Figure 3.6
Backup Offset Screen



3. Select the offsets to be backed up by moving the cursor to the desired offset using the up and down cursor keys. The selected offset will be shown in reverse video. There are four options here:
 - **TOOL WEAR** -- When wear is selected all data from the tool offset wear tables is stored as a G10 program.
 - **TOOL GEOMETRY** -- When geometry is selected all data from the tool offset geometry tables is stored as a G10 program.
 - **WORK COORDINATE** -- When work coordinate systems is selected the work coordinate offset information for the G codes G54 - G59.3 and the external offset value are stored as a G10 program.
 - **ALL** -- When all is selected all data from the tool offset geometry and wear tables and work coordinate offset tables is stored as a G10 program.

4. Once the data to save has been selected, determine the destination for the G10 program from these three options:
 - Press the {TO PORT A} softkey to send the G10 program to a peripheral attached to port A.
 - Press the {TO PORT B} softkey to send the G10 program to a peripheral attached to port B.
 - Press the {TO FILE} softkey to send the G10 program to control memory.
5. When the {TO FILE} softkey is pressed, the control will ask for a program name to store the program as. Enter the program name using the alphanumeric keys on the operator panel and press the [TRANSMIT] key (See the section on program names in chapter 10). The G10 program will be saved under the file name just entered.

3.5 Programmable Zone Table

The programmable zone feature provides a means to prevent tool motion from entering or exiting a designated area. For details on programmable zones see chapter 12.

This table contains the values for programmable zones 2 and 3. These values define the boundaries for the programmable zones and are referenced from the machine coordinate system.

Important: These values may also be entered in AMP by the system installer. Programmable zone 3 table values may also be modified by programming a G22 command (see chapter 12).

1. To display or alter the values in the programmable zone table follow the steps below:

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {PROGRAM PARAM} softkey.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVI CE SETUP	MONI - TOR	TI ME PARTS	→
↑	PTOM SI /OEM					→

3. Press the {ZONE LIMITS} softkey to display the programmable zone table as shown in Figure 3.7.

(softkey level 3)

↑	ZONE LI MI TS	F1 - F9	MI LCYC PARAM	PROBE PARAM		
---	------------------	---------	------------------	----------------	--	--

Figure 3.7
Programmable Zone Table

ENTER VALUE:						
PROGRAMMABLE ZONE						
		LOWER LIMIT	UPPER LIMIT			
LIMIT 2						
X	AXIS	0.0000	0.0000	[MM]
Y	AXIS	0.0000	0.0000	[MM]
Z	AXIS	0.0000	0.0000	[MM]
U	AXIS	0.0000	0.0000	[MM]

↑	REPLCE VALUE	ADD TO VALUE	MORE LIMITS	UPDATE & EXIT	QUIT	
---	-----------------	-----------------	----------------	------------------	------	--

Important: Programmable zone coordinates are displayed in inch or metric units for a liner axis, depending on which is the currently active program mode. Rotary axes are shown in units of degrees.

- Use the up or down cursor keys to move the block cursor to the data to be changed. Data located at the cursor-will be shown in reverse video.

5. Data can be replaced or added to as follows:

- To replace stored travel data with new data, key-in the new data and press the {REPLCE VALUE} softkey.
- To add to previously stored travel data, key-in the amount to be added and press the {ADD TO VALUE} softkey.

(softkey level 4)

	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	
--	-----------------	-----------------	--	------------------	------	--

6. To end editing the programmable zone parameters there are two choices.

- Press the {UPDATE & EXIT} softkey to store the changes made to the parameters and leave the programmable zone screen.
- Press the {QUIT} softkey to delete all changes made to the programmable zones and leave the programmable zone screen.

(softkey level 4)

	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	
--	-----------------	-----------------	--	------------------	------	--

3.6 Single Digit Feedrate Table

Use this feature to change the values set for the single digit feedrates. When a single digit F-word is encountered during block execution, the control looks to the single digit feedrate table for a feedrate. The feedrate in this table corresponding to the single digit then becomes the active feedrate. For details on programming single digit F-words see chapter 28.

Important: Feedrates must always be entered into the single digit feedrate table in units of inches per minute (IPM) or millimeters per minute (MMPM) dependent on the controls current feedrate mode.

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
--	------------------	--------	----------------	----------------	------------------	---

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITC H LANG		→
--	----------------	-----------------	---------------	--------------------	--	---

2. Press the {PROGRAM PARAM} softkey.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVI CE SETUP	MONI - TOR	TI ME PARTS	→
↑	PTOM SI /OEM					→

3. Press the {F1 - F9} softkey to display the single digit feedrate table as shown in Figure 3.8.

(softkey level 3)

↑	ZONE LI MITS	F1 - F9	MI L CYC PARAM	PROBE PARAM		
---	-----------------	---------	-------------------	----------------	--	--

Figure 3.8
Single Digit Feedrate Table

ENTER VALUE:					
1- DIGIT F- WORD			FEEDRATE [MMPM]		
	F1				. 01000
	F2				. 02000
	F3				. 03000
	F4				. 04000
	F5				. 05000
	F6				. 06000
	F7				. 07000
	F8				. 08000
	F9				. 09000
↑	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT

Important: Feedrates are displayed in inch or metric depending on which is the active mode on the system at the current time. IPM (inches per minute) or MMPM (millimeters per minute) is displayed at the top of the screen, indicating the currently active mode.

4. Use the up, or down cursor keys to move the block cursor to the feedrate parameter to be changed. The selected feedrate will be shown in reverse video.
5. There are two choices for changing feedrate values.
 - Type in a new value for the selected feedrate by using the keys on the operator panel. Then press the {REPLCE VALUE} softkey. The value typed in will replace the old value for that feedrate.
 - Type in a value to add to the old value for the selected feedrate by using the keys on the operator panel. Then press the {ADD TO VALUE} softkey. The value typed in will be added to the old value for that feedrate.

(softkey level 4)

	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	
--	-----------------	-----------------	--	------------------	------	--

6. To leave the feedrate parameter screen there are two choices.
 - To save the changes just made and leave the feedrate parameter screen press the {UPDATE & EXIT} softkey.
 - To leave the feedrate parameter screen without saving the changes press the {QUIT} softkey.

(softkey level 4)

	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	
--	-----------------	-----------------	--	------------------	------	--

END OF CHAPTER

Manual/MDI Operation Modes

4.0 Chapter Overview

This chapter describes the manual and MDI operating modes. Major topics include:

Topic:	On page:
Mechanical handle feed	4-8
Removing an axis	4-8
Manual machine homing	4-8
MDI mode	4-11

Important: This manual assumes that the standard MTB is being used and standard PAL to run that MTB panel has been installed. For applications that do not use the standard MTB panel or that do not use standard PAL to run the MTB panel, refer to documentation prepared by your system installer.

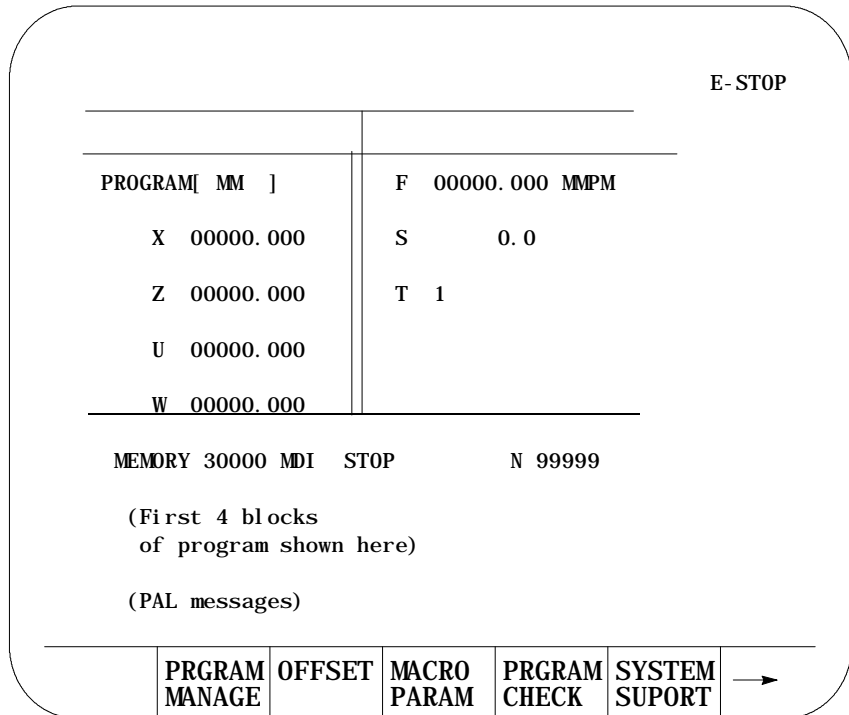
4.1 Manual Operating Mode

To go into the manual mode, select MANUAL under <MODE SELECT>. If you are not equipped with a <MODE SELECT> button or switch, press the {FRONT PANEL} softkey followed by the left/right arrow key to select the mode.

When you select the manual mode, both the axis position data and the part program currently active are displayed in the data display area if the normal display is used for softkey level 1.

Press <CYCLE STOP> to abort manual operations. However, the system installer has the option to designate some other switch to abort manual operations in the PAL program. Refer to the documentation provided by your system installer for details.

Figure 4.1
Data Display in MANUAL Mode



4.1.1 Jogging an Axis

In the jog modes, the motion of the cutting tool is controlled by the use of pushbuttons, switches, or hand pulse generators (HPGs). Typically these are mounted on or near the MTB panel.

The cutting tool can be jogged by using three different methods:

- continuous jog -- the axes move continuously while a pushbutton on the MTB panel is held.
- incremental jog -- the axes move a predetermined amount each time a pushbutton on the MTB panel is pressed.
- HPG jog -- the axes move when an HPG is turned by hand and an axis has been selected on the MTB panel.

Normally, the axes can only be jogged in manual mode. Your system installer can write PAL to allow jogging in the automatic and MDI modes. Refer to page 4-6.

The control can be equipped with an optional offset jogging feature, activated by a switch installed by the system installer. When this feature is active, all jog moves are used to offset the current work coordinate system and no position registers are changed. Refer to page 4-6 for details.

Only normal single-axis jogs (one axis at a time in the continuous, incremental, or HPG modes) are permitted during a jog retract operation.

Important: S-Curve Acc/Dec is not available during manual jogged motion.

4.1.2 Continuous Jog

To continuously jog an axis:

1. Select CONTINUOUS under <JOG SELECT>.
2. Select the feedrate for continuous jog under <SPEED/MULTIPLY>.
3. Press the <AXIS/DIRECTION> button for the axis and direction to jog. The axis moves while the button is held down.

If you want to:	Then:
alter the feedrate selected with the <SPEED/MULTIPLY> switch or button	select a <FEEDRATE OVERRIDE> %
jog the axis at a special AMP assigned traverse feedrate and ignore the setting of the <SPEED/MULTIPLY> switch or button	press and hold the <TRVRS> when jogging
jog moves that use the traverse feedrate	select a <RAPID FEEDRATE OVERRIDE> %

Important: It is possible to jog more than one axis at a time. To jog multiple axes, press and hold more than one axis direction button. The selected axes will drive at the feedrate chosen under <SPEED/MULTIPLY>. If the selected feedrate is above a specific axis maximum allowable feedrate, that axis drives at its maximum feedrate. The feedrate for the other selected axes is not affected.

4.1.3 Incremental Jog

Incremental jog manually moves an axis a predetermined amount each time an <AXIS/DIRECTION> button is pressed. To use incremental jog:

1. Select INCREMENTAL under <JOG SELECT> .
2. Select the jog increment under <SPEED/MULTIPLY>. The jog increment is equal to an amount specified in AMP for each selection under <SPEED/MULTIPLY>.

3. Press the <AXIS/DIRECTION> button for the axis and direction to jog. The control makes one incremental move each time the <AXIS/DIRECTION> button is recognized. Until the control completes the execution of the incremental move, no other jog moves are recognized on that axis. This includes attempts to perform other incremental moves on that axis.

The control will normally jog the axes the selected distance and direction at the feedrate set in AMP for the MED feedrate. It is possible for the system installer to select a different feedrate with a specific PAL Program. Refer to documentation prepared by the system installer for details.

Important: You can jog more than one axis at a time. To jog multiple axes, press more than one axis direction button. The selected axes drive at the feedrate chosen under <SPEED/MULTIPLY>. If the selected feedrate is above a specific axis maximum allowable feedrate, that axis drives at its maximum feedrate. The feedrate for the other selected axes is not affected.

4.1.4 HPG Jog

This section describes the procedure for jogging an axis with a hand pulse generator (HPG). It is assumed here that the system is configured for only one HPG. If the system installer has chosen to install more than one HPG, they may not function as described here. For details, refer to documentation provided by the system installer.

To jog an axis with a hand pulse generator (HPG):

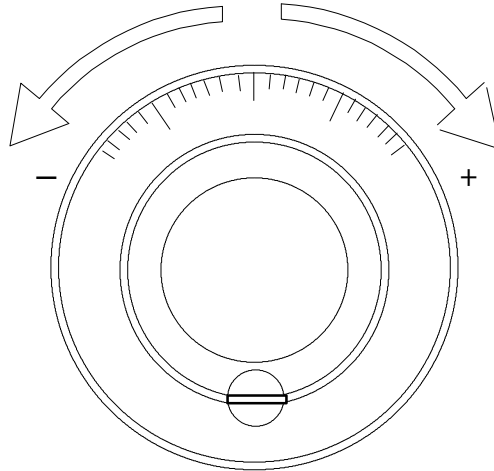
1. Select HANDWHEEL under <JOG SELECT> .
2. Select the distance that the axis moves for one division of the HPG with <SPEED/MULTIPLY>. The distance the axis moves per division is equal to an amount specified in AMP for each selection under <SPEED/MULTIPLY>.

Important: The x1000 and x10000 selections under <SPEED/MULTIPLY> are not usable in HPG jog mode.

3. Press the <AXIS/DIRECTION> button for the axis to jog. The direction the axis will move is determined by the direction the HPG is rotated and is unaffected by the <AXIS/DIRECTION> button.

Important: It is possible to jog more than one axis at a time with the HPG if the system installer has written PAL to allow it. Refer to documentation prepared by the system installer for details.

4. Typical HPG configuration will result in:
 - clockwise -- plus (+) direction for the axis
 - counterclockwise -- minus (-) direction for the axis

Figure 4.2
HPG Feed

4.1.5 Arbitrary Angle Jog

If desired, the system installer can enable a feature that allows control over the angle in which a multiaxis jog move will take through the installation of some optional switches.

When this feature is activated, the operator selects two different axes to define a plane for the arbitrary angle jog to take place. Then, an angle is selected (between 0° and 360°) to define a vector for the jog to take place. This angle is referenced from the first axis chosen to define the current plane. The system installer has the option to predefine this plane and vector in PAL. Refer to the documentation prepared by the system installer on selecting an arbitrary angle jog.

Once this vector has been defined, a jog move can be made along the defined vector by selecting HPG incremental or continuous jogging modes in the normal fashion. Refer to page 4-1. The direction that the tool will travel along the defined vector is determined by your system installer in PAL or by the operator by the position of another optional switch.

The feedrate of the tool is equal to the current feedrate as determined by the `<SPEED/MULTIPLY>` and `<FEEDRATE OVERRIDE>`. This feedrate is the vectorial feedrate for the cutting tool. Individual axis feedrates automatically adjust to produce the designated vectorial feedrate. To jog at the maximum feedrate as set in AMP, hold the `<TRVS>` button down while jogging in continuous mode.

4.1.6 Jog Offset

The control may be equipped with an optional jog offset feature, activated by a switch installed by the system installer. When this function is active, all jog moves made are added as offsets to the current work coordinate system.

Normally, jogging occurs in the manual mode. The system installer has the option to enable a “Jog on the Fly” feature that will allow jogging in automatic or MDI mode for the purpose of jogging an offset. To jog in automatic or MDI mode both the “Jog on the Fly” and jog offset features must be active. Normally, the system installer will enable both of these features with the same switch. Refer to documentation provided by the system installer for details. “Jog on the Fly” can be performed at any time during automatic operation, even while blocks are being executed.

To use this feature, follow these directions:

1. Turn on the switch to activate the jog offset function. Refer to documentation provided by the system installer.
2. Change to manual mode unless the control is equipped for the “Jog on the Fly” feature which allows jogging in MDI and Automatic modes. If equipped with “Jog on the Fly,” turn on the switch to activate it. For details, refer to documentation prepared by the system installer.
3. Jog the axis by using any of the available jog types, with the exception of homing, as described on page 4-8. The control adds the amount of the jog move as offsets to each jogged axis immediately when the jog takes place.

Important: When the jog move is made, the axis position displays do not change on the screen unless the currently active screen is the absolute screen. This is because the value is being added to the work coordinate system offset and the control does not recognize any tool motion on the coordinate system.

4.1.7 Resetting Overtravels

The control stops tool motion during overtravel conditions. Overtravel conditions can occur from 3 causes:

- **Hardware Overtravel** -- the axes reach a travel limit, usually set by a limit switch or sensor mounted on the axis. Hardware overtravels are always active.
- **Software Overtravel** -- commands cause the cutting tool to pass a software travel limit. Software overtravels are active only after the axis has been homed provided the feature has been activated in AMP by the system installer.

- **Programmable Zone Overtravel** -- the axes reach a travel limit established by independent programmable areas. Programmable Zones are activated through programming the appropriate G-code.

These 3 causes of overtravel are described in detail in chapter 12.

When an overtravel condition occurs, all axis motion stops, the control is placed in cycle stop, and one of the following error messages is displayed.

Message:	Description:
HARDWARE OVERTRAVEL (-) BY AXIS (X)	indicates that the specified axis has tripped either the + or - hardware limit switch mounted on the machine.
SOFTWARE OVERTRAVEL (+) BY AXIS (X)	indicates that an attempt was made by the specified axis to enter the overtravel area defined by the softlimits in either a positive or negative direction.
VIOLATION OF ZONE (2) BY AXIS (X)	This message indicates that an attempt was made to enter the overtravel area defined by programmable zone 2 or 3.

When a software or zone overtravel has taken place, you cannot move the axis in the same direction as the overtravel. Only axis motion in the reverse direction is possible.

Reset a hardware overtravel condition depending on the E-Stop circuit design and the way PAL was programmed by your system installer.

To reset a software or programmable zone overtravel condition:

1. Determine whether the control is in E-Stop. If it is not, go to step 4.
2. Look for and eliminate any other possible conditions that may have caused emergency stop, then make sure that it is safe to reset the emergency stop condition.
3. Press the <E- STOP RESET> button to reset the emergency stop condition. If the E-Stop does not reset, it is a result of some cause other than overtravel causing E-Stop.
4. Make sure it is safe to move the axis away from the overtravel limit.
5. Use any of the jog features described on page 4-1, except homing and jog offset, to manually move the axis away from the limit. Any attempt to jog the axis in the direction of the overtravel will not be allowed.

4.2 Mechanical Handle Feed (Servo Off)

This feature lets you disable the servo drives, and allows the axes to be moved by external means (such as a hand crank attached to the ball screw) without requiring the control to be in E-Stop. When this feature is enabled, all position displays get updated as the axes are moved.

Use this feature in conjunction with the digitize feature described in chapter 5.

This feature only enables when the control is in the Cycle Stop state and the axes are not being jogged at the time of request. To use this feature, it must be enabled in PAL by your system installer. Refer to your system installer's documentation for details on how the "Mechanical Handle Feed" feature is activated and used.

4.3 Removing an Axis (Axis Detach)

Use this feature to allow the removal of a rotary table or other axis attachment from a machine without requiring the system to be re-configured. When activated, the control ignores messages that may occur resulting from the loss of feedback from a removed axis such as servo errors.

Important: This feature removes the selected axis from the control as an active axis. Any attempt to move the removed axis results in an error. This means that part programs that use the removed axis name cannot be executed. Jog moves and MDI commands that attempt to move the removed axis also results in an error.

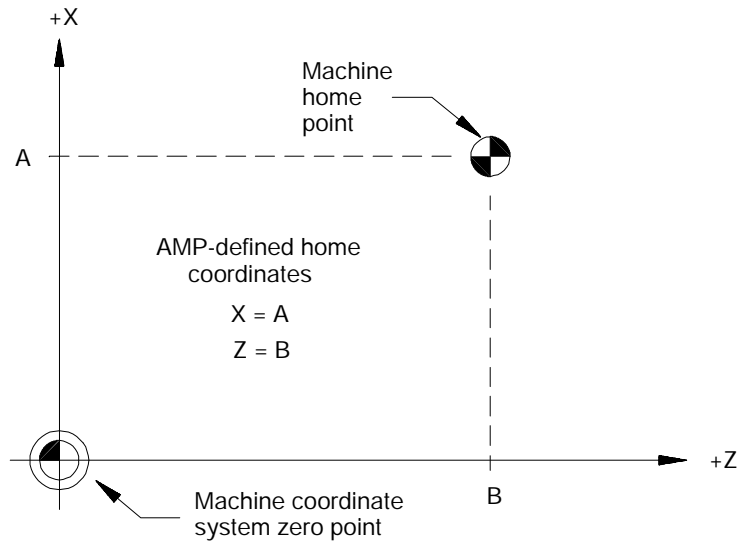
This feature can only be enabled in AMP. The axis must be selected as "Detached" to be considered removed. Refer to your system installers documentation for the necessary steps involved in actually physically removing axis hardware from a specific machine.

4.4 Manual Machine Homing

The machine home return operation means the positioning of a specified linear or rotary axis to a machine-dependent fixed position, which is called the machine home. This position is established via a home limit switch mounted on the machine and the closest marker on the feedback device.

The execution of machine home establishes the machine coordinate system. Since all of the AMP-assigned work coordinate systems and all of the programmable zones are referenced from the zero point of the machine coordinate system, none of these features are available until the machine homing operation has been conducted. Homing the axis should be the first operation done on the control after power-up.

Figure 4.3
Machine Home

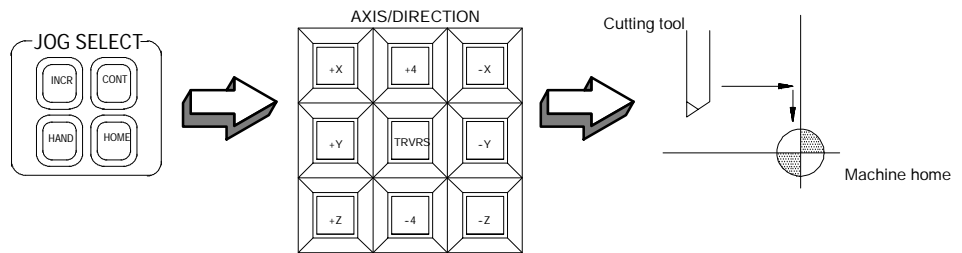


The following procedure describes how the control is homed manually by using the pushbuttons on the standard MTB panel. Manual homing may be different for some machines depending on the PAL program written by your system installer.

Important: When a homing request is made the feedback device for the axis (typically an encoder) must encounter at least one marker before tripping the homing limit switch. If the axis is close to the home limit switch you should jog the axis away from this switch before attempting a homing operation.

Important: Automatic homing is available.

Figure 4.4
Manual Machine Home



To execute the manual return to machine home position:

1. Select HOME under <JOG SELECT>.

2. Place the control in manual mode. Refer to page 4-1.
3. Determine the direction that each axis must travel to reach the home limit switch. Refer to your system installer on the location of the home limit switch on a specific machine.
4. Press the <AXIS/DIRECTION> button for the axis and direction to home. You can select more than one axis at one time. The axis selected moves at the feedrate under <SPEED/MULTIPLY>.

Important: If you choose the wrong direction for an axis, it will continue to travel in the selected direction until it contacts a hard limit and an overtravel will occur. Refer to chapter 12. Your system installer has the option to enable some button or switch (typically Cycle Stop) through the PAL program to abort a jog operation or prevent the user from homing the axis in the wrong direction. Refer to your system installer's documentation for details.

The axis homes by:

- The axis moves until it trips its home limit switch, then the axis decelerates to a stop.
- The axis then reverses direction and moves off the home limit switch at a feedrate specified in AMP.
- The controller records the distance to the nearest encoder marker or null position.
- The control then moves in a direction specified in AMP, an amount equal to the home calibration value, specified in AMP, plus the distance from the encoder marker or null position.

This locates the machine home position. When the axis reaches this position, the control resets the position registers to a machine coordinate value specified in AMP. This establishes the zero point of the machine coordinate system.

Important: During the machine home operation, softlimits and programmable zones are not active. All active coordinate offsets are cancelled.

4.5 MDI Mode

In manual data input (MDI) mode, machine operations can be controlled by entering program blocks directly by using the keys on the operator panel.

To begin MDI operations, select MDI under <MODE SELECT> or press the {FRONT PANEL} softkey followed by the left and right cursor keys to select the mode if not equipped with a mode select switch or button.

Important: If desired, your system installer has the option of disabling G- or M-code AMP-defined paramacro calls in MDI mode. For details on paramacros, refer to chapter 28.

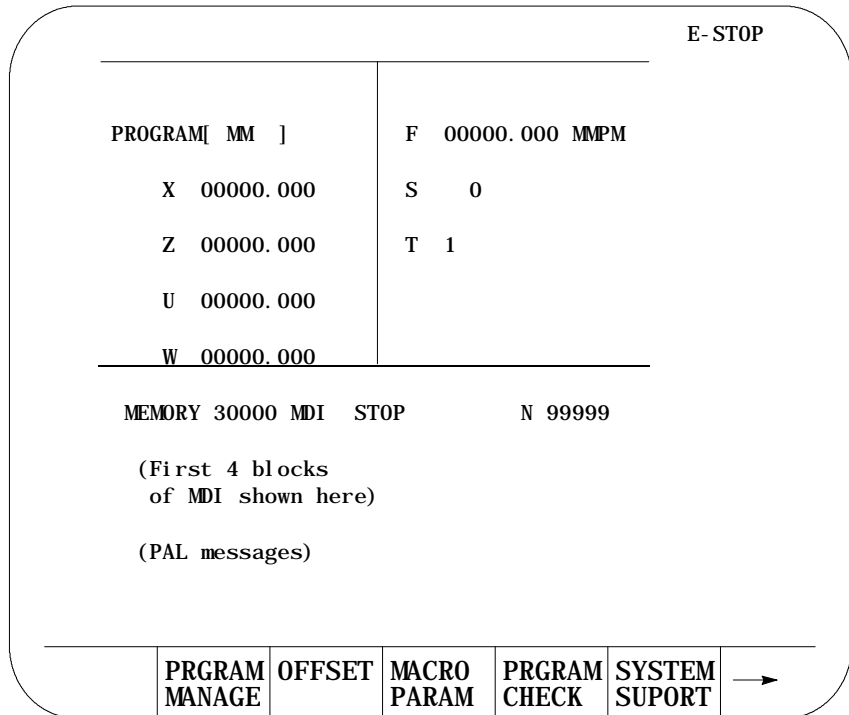
To insert blocks in an active, executing, program by using MDI, the control must be in the end of block state to allow the selection of MDI mode. If a program is interrupted while executing in automatic mode by pressing cycle stop, the control will not allow the selection of MDI since the control is in cycle suspended state not end of block state, and a mode change is not accepted.



CAUTION: When program blocks are executed in MDI, no tool tip radius compensation (TTRC) is allowed. If TTRC was previously active before the MDI blocks are executed, it is temporarily canceled for the execution of the MDI blocks. Refer to chapter 21 for details on the effect of MDI on TTRC. Any TTRC G-codes that are programmed in MDI mode affect the cutter compensation mode (G41, G42, or G40) when compensation is reactivated.

Important: It is possible to call subprograms or paramacros within an MDI program, however, there are limitations to the allowable commands. Refer to chapter 28 on paramacros for details on illegal MDI commands for these features.

Figure 4.5
Program Display Screen in MDI Mode



4.5.1 MDI Basic Operation

Operating procedures in the MDI mode include:

1. When it is in MDI mode, the control accepts standard programming blocks.
2. Key in programming blocks (refer chapter 10). Each block, up to a maximum of 62 characters, is separated with an end of block statement. The blocks entered appear in the input area of the screen (lines 2 and 3). The complete MDI program should be entered on these lines since once you send the blocks to control memory, they cannot be edited or added to.

The input cursor is the cursor shown on the input lines (lines 2 and 3 on the screen). To move the cursor right and left in the input area, press and hold the [SHIFT] key while pressing the left and right cursor keys. A new character will be inserted to the left of the cursor automatically when you press any character key.

If you make a mistake keying in a character before it is sent, that character can be edited by using the input cursor described on page 2-41.

3. Pressing the [TRANSMIT] key transmits the blocks to control memory. Once the blocks have been sent to control memory, you cannot send any more MDI blocks until all of the previous set has been executed.

The control displays the first 4 blocks of the MDI program entered on lines 17-20 with an ! (exclamation point) just to the left of the blocks. If you insert lines by using MDI within a program selected for automatic execution, the control inserts the MDI blocks just before the next block to be executed.

If you need to abort the MDI program due to an error in the MDI program or any other reason, discard the MDI program by executing a control reset operation.

4. The MDI blocks can then be executed continuously by pressing the <CYCLE START> button in either the AUTO or MDI mode. The single block, block retrace, and jog retract features are also available for MDI programs (refer to sections 7.1.4, 7.10, and 7.9 respectively for details on these features).

The control displays an “@” symbol next to any MDI blocks that have been executed.

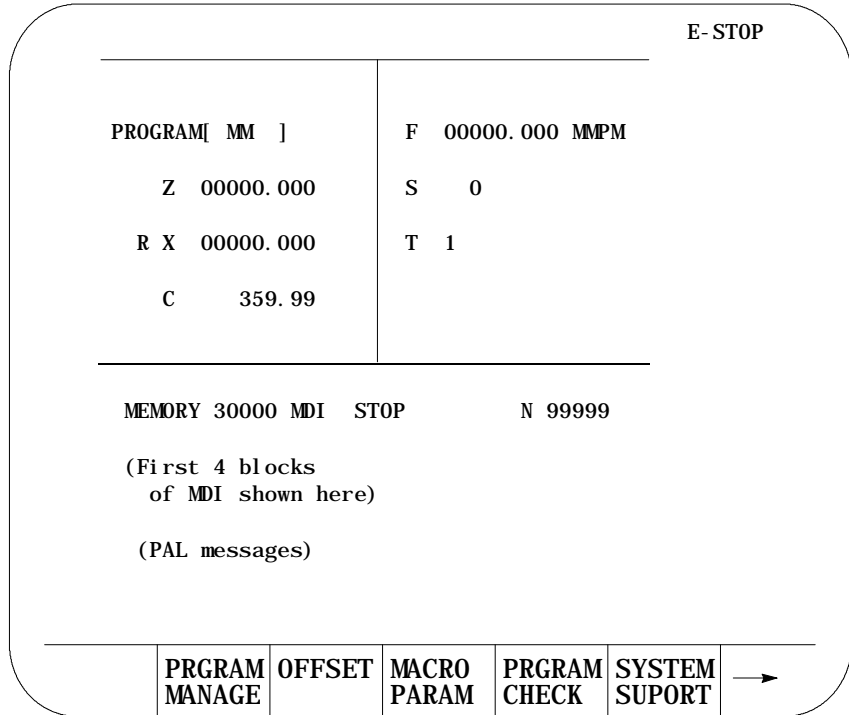
The error message:

“NO MORE MDI BLOCKS”

appears if you press cycle start in the MDI mode when there are no more MDI blocks remaining in memory to be executed.

If:	Then:
the MDI blocks were entered into an executing part program	the control returns to automatic mode and continues executing the part program.
you execute the MDI program in the MDI mode	execution halts when the control encounters the first block of the part program.

Figure 4.6
MDI Mode Program Screen



Important: Performing a block reset operation causes the control to abort the current MDI program block or skip the following MDI program block. Refer to page 2-4 for details. By performing a control reset operation as described, the control erases all MDI blocks that have not been executed in the MDI program.

END OF CHAPTER

Editing Programs Online

5.0 Chapter Overview

This chapter describes the basics of editing programs on line (at the keyboard) including:

- Selecting the program to edit
- Editing programs
- Programming aids {QUICKVIEW}
- Digitizing a program (Teach)
- Deleting program {DELETE}
- Renaming programs {RENAME}
- Displaying a program {DISPLAY}
- Comment display {COMMENT}
- Copying programs {COPY PROGRAM}

Programs may also be edited off line (at a personal computer). Other operations may also be performed on a PC including uploading, downloading, and copying. These offline operations are described in chapter 6. An extension of these offline editing features, including many file management features, may be purchased from Allen-Bradley in a Mini-DNC package (see your Allen-Bradley sales representative).

5.1
Selecting the Program To
Edit

This section discusses how to select a part program for editing. Note that only part programs that are stored in control memory may be edited online. If a part program is on tape or other storage device and must be edited online, copy this program to memory as described in chapter 9.

Important: You can edit both active and inactive programs. However, keep in mind, that edit operations being performed on an active program must be exited before that program may actually be executed in automatic mode. Note that currently active part programs are shown with an A to the left of the part program name. Programs currently open for editing display an E to the left of the program name.

To begin an edit operation on either an active or inactive part program:

1. Press the {PROGRAM MANAGE} softkey. The program directory screen is displayed (see Figure 5.1).

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITC LANG		→

Figure 5.1
Part Program Directory

SELECTED PROGRAM		
DI RECTORY		PAGE 1 OF 1
NAME	SIZE	COMMENT
MAIN	2. 3	
012345	14. 3	
RRR	9. 3	THIS IS A TEST PROG
TEST	3. 9	
4 FILES 120.2 METERS FREE		
↑	ACTIVE PROGRAM	EDIT PROGRAM
	RESTRT PROGRAM	DISPLY PROGRAM
	COPY PROGRAM	→

2. The part program to be edited can be selected using two methods:

- Keying-in the program name of the part program to edit or create.

or

- Moving the cursor to the program name on the program directory screen by using the up or down cursor keys.

Important: If you are creating a new program and using it as a subprogram, see chapter 10 and its section on program names. Programs used as subprograms must have the letter O as the first character in the program name, followed by up to 5 numeric characters.

3. Press the {EDIT PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

5.2 Editing Programs at the Control (Online)

This section covers how to edit part programs after a program has already been selected to edit as described in section 5.1.

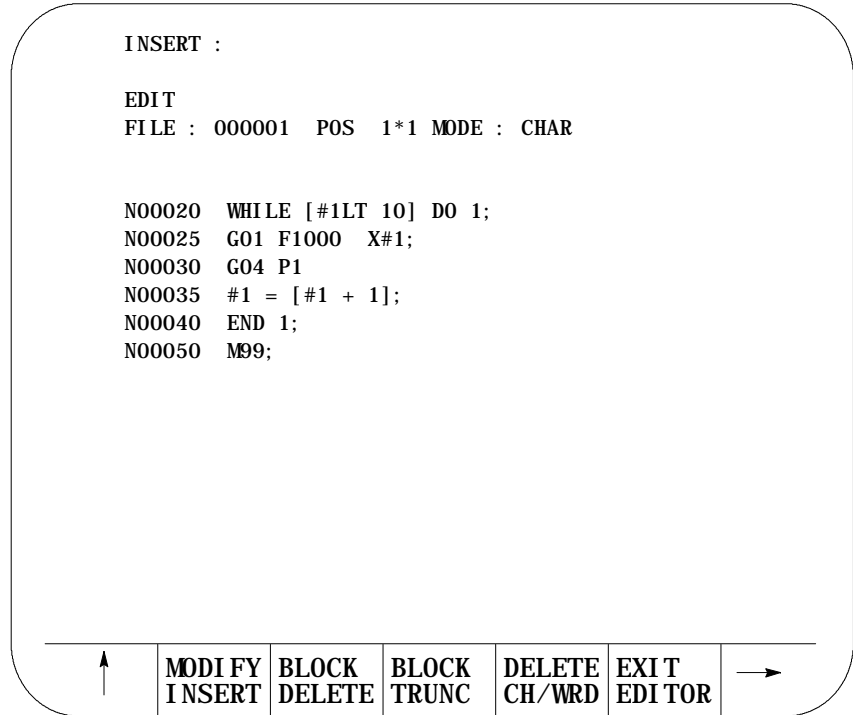


CAUTION: Any edit operation that is performed on a part program is permanent. There is no way to discard any changes that have been made to a program. Programs are saved in memory at the same time they are edited.

There are two major sections to the edit screen. They are:

- The input area. This area is in lines 2 and 3 of the screen and is where the characters appear when typed in on the keyboard.
- The program display area. This area is in lines 7-20 of the screen and is where the control displays the part program blocks as they have currently been edited.

Figure 5.2
Program Edit Screen



The maximum number of programs that you can have is 328. To store a program, it needs to occupy at least 1.3 meters of memory. Use this table to find out how much part program space there is on your system:

If your system has	this is your part program storage
64K	150 meters
128K	300 meters
192K	450 meters
Extended Part Program Storage (EPPS)	5235 meters

Extended Part Program Storage is not available on the 9/230.

The following subsections describe how to use the features available when editing a part program.

5.2.1
Moving the Cursor {STRING
SEARCH} and Cursor Size
{CHAR/WORD}

The following section discusses moving the cursor in the program display area (lines 7-20 of the CRT). It assumes that a program has already been selected to edit as discussed in section 5.1.

Important: The input cursor is the cursor shown on the input lines (2 and 3 on the screen). Details on the input cursor are given in chapter 2.

- To move the block cursor right and left, press the right and left cursor keys.
- To move the block cursor up and down, press the up and down cursor keys.
- To search for a character or character string press the continue {• } softkey to change the softkey functions and follow these steps.

1. Press the {STRING SEARCH} softkey.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PRGRAM	MERGE PRGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DI G I T Z E					→

2. To locate the cursor at the beginning or end of the file:

Press either the:

{TOP OF PRGRAM} softkey -- to move the cursor to the beginning of the part program.

{BOT OF PROGRAM} softkey -- to move the cursor to the end of the part program.

(softkey level 4)

↑	FORWRD	REVRSE		TOP OF PRGRAM	BOT OF PRGRAM
---	--------	--------	--	------------------	------------------

3. Key in the character or character string to search for, and press either the:

{FORWRD} softkey -- to search in the forward direction in the part program

{REVRSE} softkey -- to search in the reverse direction in the part program

(softkey level 4)

↑	FORWRD	REVRSE		TOP OF PRGRAM	BOT OF PRGRAM	
---	--------	--------	--	------------------	------------------	--

If the control cannot find the character or character string, it issues the error message "SEARCH STRING NOT FOUND"

4. To end the search operation press the exit [•] softkey.

- Changing Cursor Size

Sometimes it is desirable to change the cursor size for editing operations such as changing, inserting, or erasing. The control has two cursor sizes available. There is the single character cursor size, automatically assigned to the cursor when the edit screen is accessed, and there is the word cursor size, used to encompass a word and its value for using erasing, inserting, or changing operations.

Modify the cursor size in the following manner:

1. Press the continue {•} softkey to change the softkey functions.
2. Press the {CHAR/WORD} softkey to select the cursor size (character or word). The cursor size is changed each time that the {CHAR/WORD} softkey is pressed.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
---	------------------	-----------------	----------------	------------------	----------------	---

↑	STRING SEARCH	RENUM PRGRAM	MERGE PRGRAM	QUICK VIEW	CHAR/ WORD	→
---	------------------	-----------------	-----------------	---------------	---------------	---

↑	DIGITZ E					→
---	-------------	--	--	--	--	---

5.2.2 Entering Characters and Blocks

After selecting a part program to be edited, use the following method to add lines, blocks, or characters to the part program. The control should be in the edit mode at this point with EDIT: displayed in the input area of the screen (lines 2-3).

To enter blocks in a program:

1. Move the block cursor to the location in which program blocks or characters are to be added using the up, down, left and right cursor keys.
2. Type the program characters to be entered in the input area. Press the [EOB] key (end of block) at the end of each block.

If a mistake is made keying in a character before it is sent from the input area the input lines may be edited as described in chapter 2.

3. Press the [TRANSMIT] key to send data from the input lines to the program display area.

Important: After data has been sent to the program display area of the screen (lines 7-22) it can no longer be edited using the input cursor.

4. Repeat these steps until the entire program has been entered.

5.2.3 Changing and Inserting {MODIFY INSERT}

Changing

It is possible to replace characters or words with one or several characters or words by using the change feature. Follow these steps to change data that has already been entered into the program display area.

1. From the edit menu, press the {MODIFY INSERT} softkey until the MODIFY: prompt is displayed on the input line. The control toggles between modify and insert each time the {MODIFY INSERT} softkey is pressed.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PRGRAM	MERGE PRGRAM	QUI CK VI EW	CHAR/ WORD	→
↑	DI G I T Z E					→

2. Locate the block cursor in the program display area at the character(s) that need to be changed by pressing the up, down, left, and right cursor keys. Characters shown in reverse video on the screen will be the characters changed.
3. Key in a new character or word to replace data located by the cursor in the input area, then press the [TRANSMIT] key.

Important: Only the data that is within the cursor will be changed. If more data is typed in than is contained in the cursor the extra data will be inserted to the right of the cursor. If less data is typed in than is contained in the cursor, remaining data in the cursor will be deleted. The cursor size may be changed as discussed in section 5.2.1.

Example 5.1
Changing Characters

To change Z93 to W93 in the following block:

Program Block (Program Display Area)	Enter (Input Area)	Notes
G01X93Z93;		Move the block cursor to the Z in the program display area and toggle the {MODIFY/INSERT} softkey to "MODIFY:".
G01X93Z93;	W	Type this data into the input area, then press the [TRANSMIT] key.
G01X93W93;		This is the block of altered commands shown in the program display area.

Example 5.2
Changing Characters and Inserting At the Same Time

To change X97 to X96.5 in the following block:

Program Block (Program Display Area)	Enter (Input Area)	Notes
G01X97Z93;		Move the block cursor to the character 7 in the program display area and toggle the {MODIFY/INSERT} softkey to "MODIFY:".
G01X97Z93;	6.5	Type this data into the input area, then press the [TRANSMIT] key
G01X96.5Z93;		This is the block of altered commands shown in the program display area.

Example 5.3
Changing Words

To change X97 to X42 in the following block first select the word cursor size (see section 5.2.1):

Program Block (Program Display Area)	Enter (Input Area)	Notes
G01X97Z93;		Move the block cursor to the word X97 in the program display area and toggle the {MODIFY/INSERT} softkey to "MODIFY:".
G01X97Z93;	X42	Type this data into the input area, then press the the [TRANSMIT] key
G01X42Z93;		This is the block of altered commands shown in the program display area.

Inserting

It is possible to insert characters, words, and blocks to the left of the program display cursor within an already existing or newly created part program. Follow these steps to use the insert function.

1. From the edit menu, press the {MODIFY INSERT} softkey until the INSERT: prompt is displayed on the input line. The control toggles between modify and insert each time the {MODIFY INSERT} softkey is pressed.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DIGITZ E					→

2. Move the cursor in the program display area to the position that is just right of the location to insert the new data by pressing the cursor keys.
3. Key in the characters, words, or blocks of commands to be inserted into the input area, then press the [TRANSMIT] key.

Example 5.4
Inserting Characters

To change G01X97Z93; to two separate blocks:

Program Block (Program Display Area)	Enter (Input Area)	Notes
G01X97Z93;		Move the block cursor to the Z in the program display area and toggle the {MODIFY/INSERT} softkey to 'INSERT:'.
G01X97Z93;	;	Type this data into the input area, then press the [TRANSMIT] key. Note: Entering the EOB in the step above is not necessary.
G01X97; Z93;		Result

Example 5.5
Inserting Characters

To change "X123.0" to "X123.034" when the following is displayed on the input line:

Program Block (Program Display Area)	Enter (Input Area)	Notes
N1000X123.0Z45.0;		move the cursor to "Z" and toggle the {MODIFY/INSERT} softkey to 'INSERT:'.
N1000X123.0Z45.0;	34	Type this data into the input area, then press the [TRANSMIT] key.
N1000X123.034Z45.0;		Result

Example 5.6
Inserting Words

To change X93.Z20.; to X93.W31.Z20.;

Program Block (Program Display Area)	Enter (Input Area)	Notes
X93.Z20.;		move the cursor to the Z and toggle the {MODIFY/INSERT} softkey to 'INSERT:'.
X93.Z20.;	W31.	Type this into the input area, then press the [TRANSMIT] key.
X93.W31.Z20.;		Result

5.2.4 Erasing Characters and Blocks

The control can erase part program data in 3 ways:

- Erase a character or a word
- Erase all the characters from the current location of the cursor to the EOB code (:).
- Erase an entire block.

Erasing a Character or Word

1. First choose whether to erase a character or a word by pressing the {CHAR/WORD} softkey.
2. From the edit menu move the cursor until the character or word to be erased is in reverse video.
3. Press the {DELETE CH/WRD} softkey.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DI G I T Z E					→

Erasing Commands to the EOB

1. From the edit menu move the cursor until the first character or word to be erased is in reverse video.
2. Press the {BLOCK TRUNC} softkey. All information located from the cursor to the End of Block character is erased. Note that the End of Block character cannot be erased using the {BLOCK TRUNC} softkey.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DI G I T Z E					→

Example 5.7
Erasing to the End of the Block Character

To erase Z20. from the block below:

Program Block (Program Display Area)	Enter (Input Area)	Notes
X93Z20;		move the cursor to the Z
X93Z20;		Press the {BLOCK TRUNC} softkey.
X93;		Result

Erasing An Entire Block:

1. From the edit menu move the cursor to any character that is in the block to be erased.
2. Press the {BLOCK DELETE} softkey. The selected entire block is erased including the end of block character.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DI G I T Z E					→

Example 5.8
Erasing An Entire Block

Program Block (Program Display Area)	Enter (Input Area)	Notes
X93M01Z10;		Position the cursor any where in the block
X93M01Z10;		Press the {BLOCK DELETE} softkey.
		Result -- the block will be completely deleted

Important: If the block consist of more than one line on the CRT the entire block is deleted, not just the line that contains the cursor.

Erasing a character still in the input area:

Characters can also be erased in the input area (lines 1-2 on the screen) before being sent to the program display area. Delete the character to the left of the cursor on the input line by pressing the [DEL] key. Delete all information on the input line by processing the [DEL] key while holding down the [SHIFT] key. A new character will be inserted to the left of the cursor automatically when any character key is pressed. The cursor can be moved on the input line by holding down the [SHIFT] key and pressing the left or right cursor keys.

5.2.5
Sequence Numbers {RENUM
PRGRAM}

Each block in a part program can be assigned up to a five-digit numeric value following an N address. These numbers are referred to as sequence numbers and are used to distinguish one block from another.

Sequence numbers can be assigned at random to specific blocks or to all blocks if desired. Blocks assigned sequence numbers can be called later by designating their sequence number.

Sequence numbers can be renumbered or assigned in 3 ways:

- Manually type in a sequence number following an N word for each block individually.
- assign new sequence numbers only to blocks that already have sequence numbers
- assign new sequence numbers to all blocks

Follow these steps to assign or renumber sequence numbers:

1. From the edit menu, press the continue softkey {• } to change the softkey functions.
2. Press the {RENUM PROGRAM} softkey

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DI G I T Z E					→

3. Key in an initial sequence number (the number for the first sequence number), a comma, and an incremental value for the control to add to each new sequence number. The format to this command is:

RENUM: initial-sequence-number, incremental-value

For example

RENUM 5, 10

Would make the first sequence number 5, the next 15, 25, 35....

Important: Both the initial sequence number and the incremental value must be entered as integer values. If a decimal point is used the control will issue the error message "ILLEGAL VALUE".

If no initial sequence number or no incremental-value is given or both are not entered the control will use default values of one.

4. Here are two choices:

- To assign sequence numbers or to resign sequence numbers to all blocks from the beginning of the part program, press the {ALL} softkey.
- To assign new sequence numbers to only the blocks that already have sequence numbers, press the {ONLY N} softkey.

(softkey level 4)

↑	A L L	O N L Y N				
---	-------	-----------	--	--	--	--

Important: Any sequence numbers in a block that are referenced in the current program by a paramacro “GOTO” or “WHILE” will also be renumbered. In order to maintain proper program execution it will be necessary to manually alter the “GOTO” or “WHILE” blocks target block sequence number.

5.2.6 Merging Part Programs {MERGE PRGRAM}

It is possible to merge a complete part program within another part program while one of the programs is in the edit mode. To merge part programs follow these steps:

1. Move the cursor in the program display area just after the location that another program is to be inserted. The merged program will be inserted to the left of the cursor with each end of block statement creating a new line.
2. Press the continue softkey {• } to change the softkey functions.
3. Press the {MERGE PRGRAM} softkey. The prompt “MERGE:” will appear on the input line of the screen.

(softkey level 3)

↑	M O D I F Y I N S E R T	B L O C K D E L E T E	B L O C K T R U N C	D E L E T E C H / W R D	E X I T E D I T O R	→
---	----------------------------	--------------------------	------------------------	----------------------------	------------------------	---

↑	S T R I N G S E A R C H	R E N U M P R G R A M	M E R G E P R G R A M	Q U I C K V I E W	C H A R / W O R D	→
---	----------------------------	--------------------------	--------------------------	----------------------	----------------------	---

↑	D I G I T Z E					→
---	------------------	--	--	--	--	---

- Key-in the program name of the part program to insert, then press either the [TRANSMIT] key or the {EXEC} softkey.

(softkey level 1)

↑	EXEC					
---	------	--	--	--	--	--

5.2.7 Exiting Edit Mode {EXIT EDITOR}

When you edit a program, all changes and additions that you make are saved immediately in the control's memory. No formal "save" command is executed.

You cannot quit, abandon or abort an edit session and restore the original version of the program you have been editing. For that reason, we recommend that you copy the program prior to editing.

To exit the edit mode from the edit menu, press the {EXIT EDITOR} softkey.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
---	------------------	-----------------	----------------	------------------	----------------	---

↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
---	------------------	------------------	------------------	---------------	---------------	---

↑	DI G I T Z E					→
---	-----------------	--	--	--	--	---

Important: Do not press the Exit {•} softkey to leave a program being edited. The {EXIT EDITOR} softkey must be used to exit the program. Exiting using the Exit {•} softkey (shown on the CRT as a up arrow) will cause the program editor to remain open for that program. This results in not allowing any other program to be selected for editing.

5.3
Programming Aids {QUICK
VIEW}

The QuickView features display sample patterns or the G-code prompts to help in writing part programs. By keying in data corresponding to prompted messages, the control will automatically generate the required block(s) to insert into an existing part program.

The QuickView feature can aid the programmer by giving access to:

- QuickPath Plus Prompts -- a selection of commonly used sample patterns representing a series of machining steps with prompts for the necessary words to program it using QuickPath Plus (see chapter 15 for details on QuickPath Plus).
- G-code Prompting -- a selection of the more commonly used G-codes with prompts for the necessary words to program it.
- Milling Cycle Prompting -- a selection of the more commonly used fixed cycles with prompts for the necessary words to program it.
- Plane Select -- a selection of the planes in which the different QuickView are programmed.

Use the following steps to select the QuickView features:

1. Select a program for editing as discussed in section 5.1.
2. From the edit menu, press the {QUICK VIEW} softkey.
(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
---	------------------	-----------------	----------------	------------------	----------------	---

↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
---	------------------	------------------	------------------	---------------	---------------	---

↑	DI G I T Z E					→
---	-----------------	--	--	--	--	---

3. The softkey functions will change to those indicated below.
(softkey level 4)

↑	QPATH+ PROMPT	GCODE PROMPT	M I L L PROMPT		PLANE SELECT
---	------------------	-----------------	-------------------	--	-----------------

4. See the following subsections on using the different QuickView functions.

Axis Selection

The selection of the axes that can be programmed using QuickView is determined by the type of QuickView prompt you are using. The two factors the control uses to determine the axes for QuickView are based on if the QuickView prompt is for a planer G-code or a non-planer G-code.

- **Planar G-codes** - Planar G-codes are any feature that is plane dependant (such as G02, G41, Cycles, etc...). The first two axes are selected with the {PLANE SELECT} QuickView softkey discussed on page 5-19. The third axis displayed is the axis not in the current plane but in both of the other planes defined. For example if G17=XY, G18=ZX , G19=YZ and G18 is selected as the QuickView plane than Y would be the third axis since it is in both G17 and G19 planes but not in the G18 plane. If there is no common axis between these two planes than the next axis defined in the process that is not already in the QuickView plane is used.
- **Non-Planar G-codes** - Non-planar G-codes are any feature that is not plane dependant (such as G01, G04, G92, etc...). The axes used for QuickView prompts for these features are independent of the QuickView plane you have selected. The control uses the first three axes configured in the current process.

Important: Two digit axis names are not compatible with the QuickView feature (typically only used on systems with more than 9 axes and consist of a dollar sign "\$" followed by a letter). When an attempt is made to display one of these axis names on a QuickView prompt, the axis name is displayed as a \$ only. QuickView can not be used to create part program blocks with \$ axis names.

On quickview screens that display more axis then currently configured in the system (as configured in AMP), the graphics and prompts will display asterisks for the un-available axes names. No data can be entered on these prompts.

5.3.1 Selecting a QuickView Plane

This feature is used to select the plane that is used to program the different QuickView features in. This will determine what plane is displayed for the prompting and their axis names displayed for the prompts. It is not possible to select any parallel planes with the QuickView feature, only primary planes may be used.

The system installer determines the primary planes established by G17, G18, and G19 in AMP. See chapter 13 for details on plane selection. To set the plane to program in with the QuickView feature follow these steps:

1. From the QuickView menu press the {PLANE SELECT} softkey.
(softkey level 4)

↑	QPATH+ PROMPT	GCODE PROMPT	MILL PROMPT	PLANE SELECT
---	------------------	-----------------	----------------	-----------------

The current plane selected is displayed. The right side of the screen will list the primary and secondary axes for the plane. "AXIS 1" is the first axis in the plane (primary axis), and "AXIS 2" is the second axis in the plane (secondary axis).

2. Change the plane by pressing the softkey that corresponds to the plane desired to program in (G17, G18, or G19). Refer to documentation prepared by the system installer for details on the planes selected by these G-codes.

The display will change to show the selected plane.

(softkey level 5)

↑	SET	G17	G18	G19
---	-----	-----	-----	-----

3. If the plane displayed is the plane it is desired to program the QuickView feature in press the {SET} softkey. All of the prompts for the QuickView feature are changed to display the newly selected plane.

Important: This does not select the current operating plane on the control, it only modifies the plane being programmed in. To change the currently active plane on the control, it is necessary to execute a program block containing a G17, G18, or G19 block. This block is automatically generated when necessary by the QuickView feature and inserted into the program being edited so when this program is executed the correct planes will be established.

5.3.2 Using {QPATH+ PROMPT} Sample Patterns

With the QuickView functions and the QuickPath Plus section, dimensions from part drawings can be used directly to create a part program. The sample patterns available with the QuickPath Plus prompts are summarized below.

{CIR ANG PT}	The arc radius and the taper angle of a line are known for the geometry from an arc to the line. R (Radius), A (Angle in degrees), G (02 for clockwise 03 for counter clockwise) and the coordinates of the endpoint.
{CIR CIR}	The coordinates of the arc centers are known for the geometry from an arc to another arc.
{ANG CIR PT}	The taper angle of a line and the arc radius are known for the geometry from the line to an arc.
{1ANG PT}	The taper angle and either of the end point coordinate values are known for the definition of a line.
{2ANG 1PT}	The angles of two tapered lines are known for defining two successive tapered lines.
{2ANG 1PT R}	The angles of two tapered lines are known for defining two lines joined by an arc.
{2PT R}	The intersection point of two lines are known for defining two lines joined by an arc.
{2ANG 1PT C}	The angles of two tapered lines are known for defining two lines joined by a chamfer.
{2PT C}	The intersection point of two lines are known for defining two lines joined by a chamfer.
{2ANG 2PT R}	The taper angles of two tapered lines and two arc radii are known for defining three lines joined by and tangent to two arcs.
{3PT R}	Two intersection points are known for defining three lines joined by an arc between two adjacent lines.
{2ANG 2PT C}	The Taper angles of two lines and one intersection point are known for defining three lines joined by chamfers.
{3PT C}	Two intersection points are known for defining three lines joined by chamfers.
{2ANG 2PT RC}	The taper angles of two tapered lines and one intersection point are known for defining three lines joined by an arc and a chamfer.
{3PT RC}	Two intersection points are known for defining three lines joined by an arc and a chamfer.
{2ANG 2PT CR}	The taper angles of two lines and one intersection point are known for defining three lines joined by an arc and a chamfer.
{3PT CR}	Two intersection points are known for defining three lines joined by an arc and a chamfer.

Angle of a line, corner radius, and chamfer size is often necessary for a sample pattern in QuickPath Plus prompting. The following prompts in QuickPath Plus prompting refer to the following drawing dimensions:

,A Angle
,R Corner radius
,C Chamfer size
L Length of line

For more information regarding these designations see chapter 15 on programming QuickPath Plus or chapter 16 on programming chamfers and corner radius.

The system installer may select a different address for angle ,A in AMP if desired, refer to system installers documentation.

Axis words followed by a (1), (2), or (3) are prompting for the first, second, or third coordinate position respectively. The location of the first, second, or third axis word is shown on the drawing accompanying the prompt screen.

Prompts that are shown in reverse video indicate that this parameter is optional and need not be entered in this block. Prompts that are shown with an OR to the left of the prompt indicate that there are two or more optional parameters for this prompt, only one of them must have a value assigned to it.

Follow these steps to use the QuickPath Plus QuickView feature.

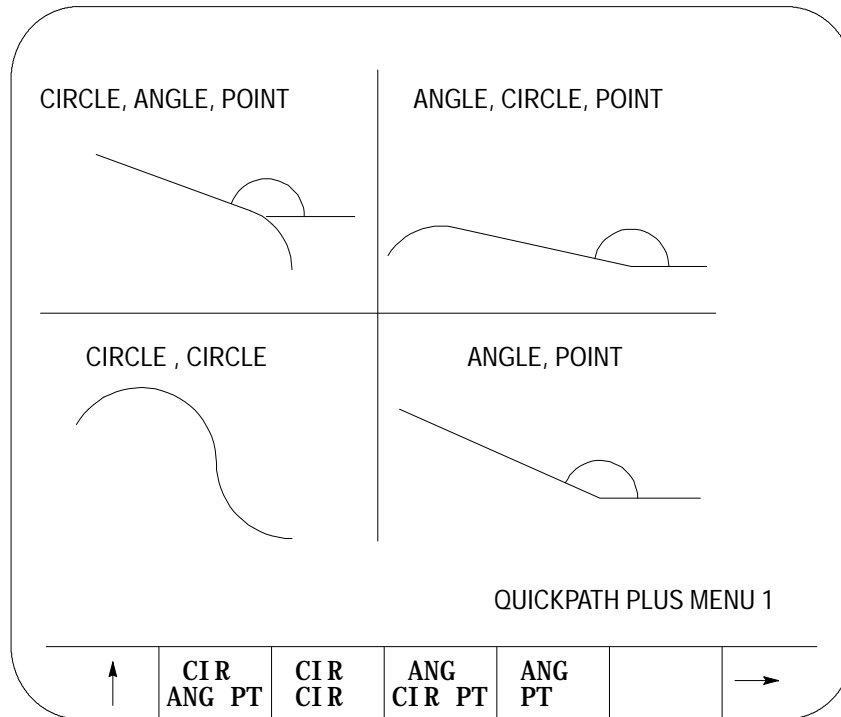
1. From the QuickView menu, press the {QPATH+ PROMPT} softkey.
(softkey level 4)

↑	QPATH+ PROMPT	GCODE PROMPT	MILL PROMPT	PLANE SELECT
---	------------------	-----------------	----------------	-----------------

2. Select a sample pattern matching the part geometry to be programmed from the previous list of sample patterns and press the corresponding softkey.

To change the QuickPath Plus softkey menu, press the continue softkey {• }.

Figure 5.3
QuickPath Plus Menu Screen



3. After selecting the desired sample pattern enter values for the parameters in the following way.

Use the up and down cursor keys to select the parameter to change or enter. The selected item will be shown in reverse video.

Type in the desired data on the input line of the CRT and press the [TRANSMIT] key. The data typed in will be entered as the data for the selected parameter.

4. After all data for the pattern has been entered store the data by pressing the {STORE} softkey.

(softkey level 6)



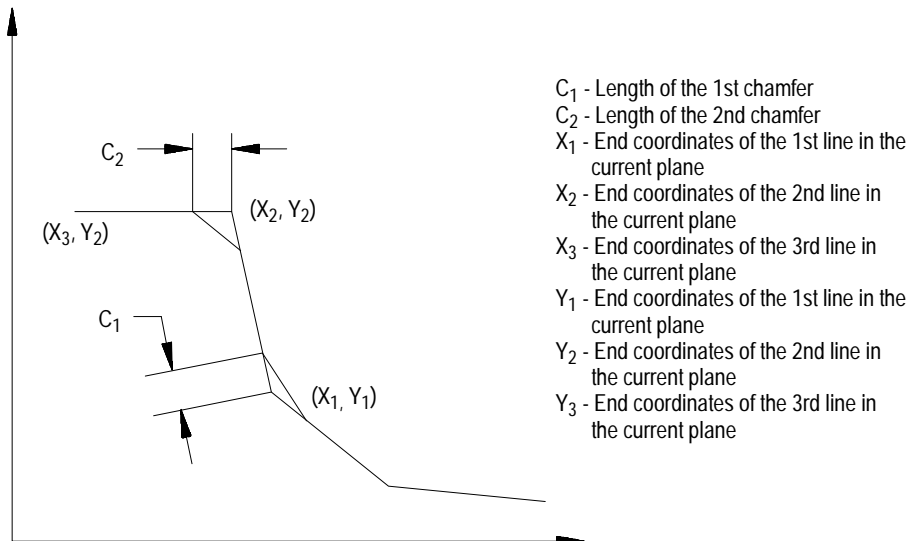
The control will generate the necessary block(s) to create the desired axis moves. These blocks are displayed in the input area next to the EDIT: prompt. These blocks may be edited in the input area as discussed in chapter 2.

5. To enter the blocks in the program being edited, move the block cursor in the program display area just past the location in the program where it is desired to insert the new blocks. Then press the [TRANSMIT] key. The generated blocks will be entered to the left of the cursor.
6. Press the exit {•} softkey to return to the main edit menu or press a different QuickView key for more prompting.

The following is an example of using one of the QuickPath Plus prompting screens:

After the {3PT C} softkey is pressed the prompt screen for that sample pattern becomes available. The following figure shows what data must be entered for that prompted screen to generate the necessary tool paths correctly.

Figure 5.4
Entering Data in the 3PT C Screen



5.3.3
G-code Format Prompting
{GCODE PROMPT}

G-code format prompting aids the operator in programming different G-codes by prompting the programmer for the necessary parameters. A graphical representation is usually provided also to show the programmer a sample of what the G-code parameters are used for.

Milling fixed-cycle G-codes are available under fixed-cycle prompting, section 5.3.4.

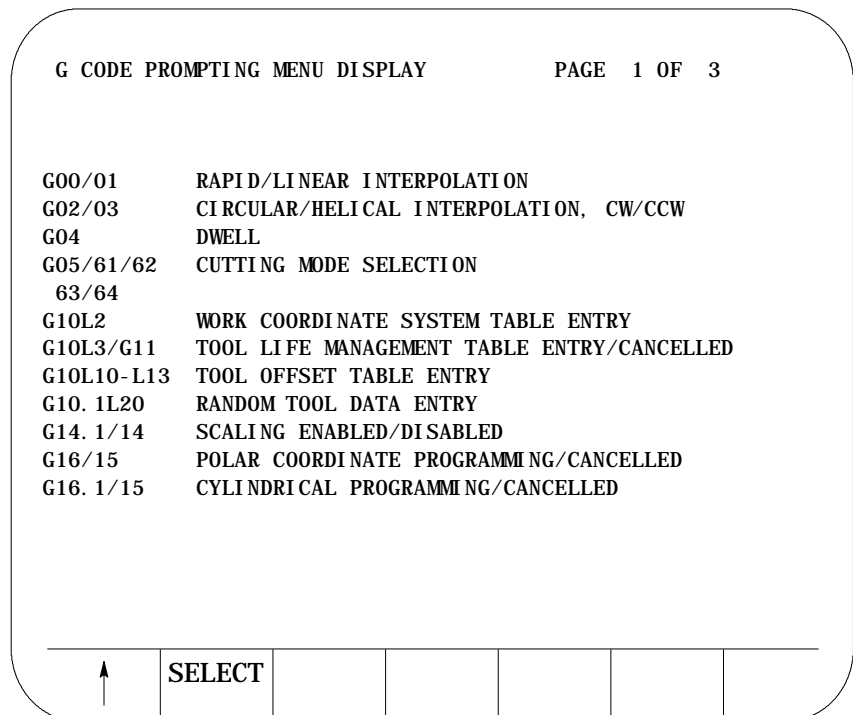
The following is a description of how to use the G-code prompting menus.

1. From the QuickView menu press the {G CODE PROMPT} softkey. The following G-code prompt select screen becomes available.

(softkey level 4)

↑	QPATH+ PROMPT	GCODE PROMPT	MILL PROMPT	PLANE SELECT
---	------------------	-----------------	----------------	-----------------

Figure 5.5
G-code Prompt Select Screen



Press the <SHIFT> and <•> keys to select the next page of G-code prompts. Press the <SHIFT> and <◀> keys to select the previous page of G-code prompts.

2. Position the cursor at the desired G-code to prompt by using the up and down cursor keys. The selected G-code is shown in reverse video.
3. Once the correct G-code is selected, press the {SELECT} softkey. A screen with prompts for that G-code is displayed.
4. Use the up and down cursor keys to select the parameters to be changed or entered. The selected item will be shown in reverse video.

Axis words followed by a (1), (2), or (3) are prompting for the first, second, or third coordinate position respectively. The location of the first, second, or third axis word is shown on the drawing accompanying the prompt screen.

Prompts that are shown in reverse video indicate that this parameter is optional and need not be entered in this block. Prompts that are shown with an OR to the left of the prompt indicate that there are two or more optional parameters for this prompt, only one of them must have a value assigned to it.

5. Type in the desired data for that parameter on the input line of the CRT and press the [TRANSMIT] key.
6. After all data for the G-code has been entered store the data by pressing the {STORE} softkey.

(softkey level 6)

↑	STORE					
---	-------	--	--	--	--	--

The control will generate the necessary G-code block. The generated block is displayed in the input area next to the EDIT: prompt. This block may be edited in the input area using the techniques discussed in chapter 2.

7. To enter the blocks in the program being edited, move the block cursor in the program display area just past the location in the program where it is desired to insert the new blocks. Pressing the [TRANSMIT] key will enter the generated block in the program to the left of the cursor.
8. Press the exit {•} softkey to return to the main edit menu or press a different QuickView key for more prompting.

5.3.4 Mill Cycle Format Prompting

Milling fixed cycle format prompting aids the programmer by prompting for the necessary parameters for the milling cycle. A graphical representation illustrating the fixed cycles operation and use of the parameters is also displayed.

For G-code prompts see section 5.3.3.

To use the MILL fixed cycle prompting function follow the steps below.

1. From the QuickView menu press the {MILL PROMPT} softkey. The mill prompt selection screen is made available.

(softkey level 4)

↑	QPATH+ PROMPT	GCODE PROMPT	MILL PROMPT	PLANE SELECT
---	------------------	-----------------	----------------	-----------------

Figure 5.6
Mill Prompt Selection Screen

MILL PROMPT MENU DISPLAY						
G80: CANCEL OR END FIXED CYCLE						
G81: DRILL CYCLE, NO DWELL/RAPID OUT						
G82: DRILL CYCLE DWELL/RAPID OUT						
G83: DEEP HOLE DRILL CYCLE						
G73: DEEP HOLE PECK DRILL CYCLE, DWELL						
G84: RIGHT HAND TAPPING CYCLE						
G74: LEFT HAND TAPPING CYCLE						
G85: BORING CYCLE, NO DWELL/FEED OUT						
G86: BORING CYCLE, SPINDLE STOP/RAPID OUT						
G76: BORING CYCLE, SPINDLE SHIFT						
G87: BACK BORING CYCLE						
G88: BORING CYCLE, SPINDLE STOP/MANUAL OUT						
G89: BORING CYCLE, DWELL/FEED OUT						
↑	SELECT					

2. Position the cursor at the desired cycle to prompt by using the up and down cursor keys. The selected cycle is shown in reverse video.
3. Once the correct cycle code is selected, press the {SELECT} softkey. A screen with prompts for that cycle is displayed along with a graphical representation of the cycle.

- Use the up and down cursor keys to select the parameters to be changed or entered. The selected parameter will be shown in reverse video.

Axis words followed by a (1), (2), or (3) are prompting for the first, second, or third coordinate position respectively. The location of the first, second, or third axis word is shown on the drawing accompanying the prompt screen.

Prompts that are shown in reverse video indicate that this parameter is optional and need not be entered in this block. Prompts that are shown with an OR to the left of the prompt indicate that there are two or more optional parameters for this prompt, only one of them must have a value assigned to it.

- Type in the desired data for that parameter on the input line of the CRT and press the [TRANSMIT] key.
- After all data for the G-code has been entered store the data by pressing the {STORE} softkey.

(softkey level 6)

↑	STORE					
---	-------	--	--	--	--	--

The control will generate the necessary G-code block. The generated block is displayed in the input area next to the EDIT: prompt. This block may be edited in the input area using the techniques discussed in chapter 2.

- To enter the blocks in the program being edited, move the block cursor in the program display area just past the location in the program where it is desired to insert the new blocks. Pressing the [TRANSMIT] key will enter the generated block in the program to the left of the cursor.
- Press the exit {•} softkey to return to the main edit menu or press a different QuickView key for more prompting.

For more details on the parameters used in a specific cycle refer to chapter 26 for details on milling cycle operation and parameters.

5.4 Digitizing a Program (Teach)

The digitize feature allows the programmer to generate blocks in a program based on the actual position of the cutting tool rather than typing in positions manually. The control records actual tool locations and uses them to generate program blocks.

The digitize feature can be used in any operating mode (auto, manual, or MDI). It is a function accessed through the regular edit menu which allows any of the other edit features discussed in section 5.2 to be used on blocks that are generated using the digitize feature.

To use the digitize feature select or create a part program for editing as described in section 5.2.

Important: The following description covers the use of softkeys to perform digitizing. It is possible for the system installer to have written PAL to allow some other method of digitizing. If this is the case refer to documentation provided by the system installer.

1. Select a part program to edit by pressing the {PRGRAM MANAGE} softkey. Enter a program name and press the {EDIT PRGRAM} softkey (see section 5.1 on selecting a program).
2. From the edit menu press the {DIGITZE} softkey.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DI G I T Z E					→

3. Position the cutting tool using one of the following methods. The cutting tool should be located at the desired start point of the new program.
 - Jog the Axes in manual mode.
 - Automatically move the axes by executing a part program or MDI program.
 - Manually move the axes as long as the encoder is still actively recording the tool position (see documentation prepared by the system installer).

4. Press the {MODE SELECT} softkey if it is necessary to change any of the following programming modes while digitizing a program:
- Inch/metric
 - Absolute programming/incremental programming.
 - Change planes G17, G18, or G19.

Press any of the softkeys corresponding to the desired mode to change. The control will display the mode that the next block will be programmed, in the upper right hand corner of the screen. The modes are abbreviated as discussed in Table 5.A.

(softkey level 5)

↑		INCH METRIC	ABS/ INCR	PLANE SELECT	
---	--	----------------	--------------	-----------------	--

Table 5.A
Changing Program Mode During Digitizing

MODE CHANGED TO:	ABBREVIATION	G CODE GENERATED	SOFTKEY
ABSOLUTE MODE	ABS	G90	{ABS/INCR}
INCREMENTAL MODE	INC	G91	{ABS/INCR}
PLANE SELECTED	G17,G18,G19	G17, G18, G19	{PLANE SELECT}
INCH MODE	INCH	G20	{INCH/METRIC}
METRIC MODE	METRIC	G21	{INCH/METRIC}

Each time the softkey corresponding to the mode is pressed the displayed G-code will change. When the desired mode is displayed exit the mode select screen by pressing the exit {•} softkey.

This will return the control to softkey level 4. The next block generated will have the necessary G-code(s) to change the mode inserted at the beginning of the block.

Important: This does not change the current operating mode of the control. The control only inserts the blocks to change the operating mode in the program. The mode does not change unless that block is executed or some other MDI or program block that changes modes is executed.

5. Determine if the next move will be linear or circular.
 - If the next move is to be linear press the {LINEAR} softkey (section 5.4.1).
 - If the next move is to be circular press either the:
 - {CIRCLE 3 PNT} softkey if 3 points on the arc are known. (section 5.4.2)
 - {CIRCLE TANGNT} softkeys if the endpoint of the arc and the line that is tangent to the start point of the arc is known. (section 5.4.3)

(softkey level 4)

↑	LINEAR	CIRCLE 3 PNT	CIRCLE TANGNT	MODE SELECT		
---	--------	-----------------	------------------	----------------	--	--

Press the exit {•} softkey to leave the digitize menu or press a different digitize softkey to continue.

5.4.1 Linear Digitizing

The following subsection assumes that steps 1-5 in section 5.4 have been completed to initiate a digitizing operation.

If the next move is to be linear press the {LINEAR} softkey . When the {LINEAR} softkey is pressed the control sets the current tool position as the start point of a linear move. The screen changes to display the current tool location in large display characters.

Figure 5.7
Linear Digitize Screen

DIGITIZE:		METRIC, ABS, G17	
ABSOLUTE [MM]		G00	
X		0.000	
Y		0.000	
Z		0.000	
F	0.000	MMPM S	00
↑		STORE END PT	EDIT & STORE

Reposition the tool at the desired end point of the linear move using any of the following methods.

- Jog the Axes in manual mode.
- Automatically move the axes by executing a part program or MDI program.
- Manually move the axes using any means as long as the encoder is still actively recording the tool position (see documentation prepared by the system installer).

After the axes have been positioned at the end point of the linear move press either the {STORE END PT} or the {EDIT & STORE} softkeys. This will record the current tool location as the final position for this digitize operation.

- The {STORE END PT} softkey does not return the control to the program display screen. Pressing this softkey inserts the generated block at whatever location the cursor was last at and allows the operator to immediately begin entering the next block using this same digitize feature.
- The {EDIT & STORE} softkey returns the control to the program display screen. The program block just generated is displayed on the input line of the CRT and may be edited as discussed in chapter 2. This block is then inserted at the location of the block cursor when the [TRANSMIT] key is pressed.

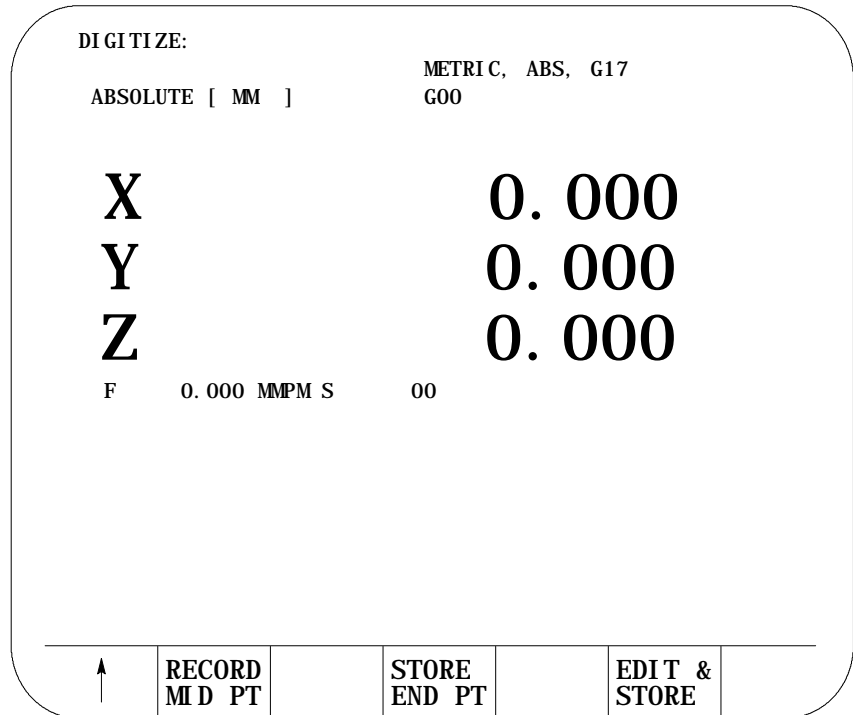
Important: To abort the linear digitize operation press the exit {• } softkey at any time before pressing the {STORE END PT} or {EDIT & STORE} softkeys. The control will be returned to softkey level 4. Then select either another digitize feature (by pressing one of the softkeys) or the control can be returned to the regular edit session (as discussed in section 5.2) by pressing the exit {• } softkey.

5.4.2 Digitizing an Arc (3 Points)

The following subsection assumes that steps 1-5 in section 5.4 have been completed to initiate a digitizing operation.

Press the {CIRCLE 3 PNT} softkey if 3 points on the circle are known. When the {CIRCLE 3 PNT} softkey is pressed the control sets the current tool position as the start point of a circular move (first point of 3 that are necessary to describe an arc). The screen changes to display the current tool location in large display characters.

Figure 5.8
CIRCLE 3 PNT Digitize Screen



Reposition the tool at any point on the arc between the start and the end point using any of the following methods.

- Jog the Axes in manual mode.
- Automatically move the axes by executing a part program or MDI program.
- Manually move the axes using any means as long as the encoder is still actively recording the tool position (see documentation prepared by the system installer).

After the axes have been positioned at a point on the arc press the {RECORD MID PT} softkey. The control will store the current tool position as the second point on the arc.

Important: If an axes is moved that is not in the currently defined plane (G17, G18, or G19) the control will interpret the move as a helical move and generate the block accordingly. (see helical interpolation in chapter 14).

After the second point on the arc has been stored reposition the axes at the end point of the arc. Store this block as a circular block by pressing either the {STORE END PT} or the {EDIT & STORE} softkeys. This will record the current tool location as the final position for this digitize operation.

- The {STORE END PT} softkey does not return the control to the program display screen. Pressing this softkey inserts the generated block at whatever location the cursor was last at and allows the operator to immediately begin entering the next block using this same digitize feature.
- The {EDIT & STORE} softkey returns the control to the program display screen. The program block just generated is displayed on the input line of the CRT and may be edited as discussed in chapter 2. This block is then inserted at the location of the block cursor when the [TRANSMIT] key is pressed.

Important: To abort the circular digitize operation press the exit {•} softkey at any time before pressing the {STORE END PT} or {EDIT & STORE} softkeys. The control will be returned to softkey level 4. Then select either another digitize feature (by pressing one of the softkeys) or the control can be returned to the regular edit session (as discussed in section 5.2) by pressing the exit {•} softkey.

5.4.3 Digitizing an Arc Tangent at End Points

The following subsection assumes that steps 1-5 in section 5.4 have been completed to initiate a digitizing operation.

Press the {CIRCLE TANGNT} softkeys if the circle is tangent at the endpoints of the previous cutting path. When the {CIRCLE TANGNT} softkey is pressed the control sets the current tool position as the start point of a circular move. The screen changes to display the current tool location in large display characters.

If the previous block was circular a tangent to the end point of the arc is used as the tangent point to the following block.

Important: The {CIRCLE TANGNT} feature can not be selected as the first motion block in a part program since there is no path for the start of the arc to be tangent to.

Figure 5.9
CIRCLE TANGNT Digitize Screen

DI G I T I Z E:		METRIC, ABS, G17	
ABSOLUTE [MM]		G00	
X	0.000		
Y	0.000		
Z	0.000		
F	0.000	M P M S	00
↑		STORE END PT	EDIT & STORE

Reposition the tool at the end point of the arc using any of the following methods.

- Jog the Axes in manual mode.
- Automatically move the axes by executing a part program or MDI program.
- Manually move the axes using any means as long as the encoder is still actively recording the tool position (see documentation prepared by the system installer).

Important: If an axis is moved that is not in the currently defined plane (G17, G18, or G19) the control will interpret the move as a helical move and generate the block accordingly. (see helical interpolation chapter 14)

After the axes have been positioned at the end point of the arc press either the {STORE END PT} or the {EDIT & STORE} softkeys. The control will store the current tool position as the end point of the arc.

- The {STORE END PT} softkey does not return the control to the program display screen. Pressing this softkey inserts the generated block at whatever location the cursor was last at on the program edit screen and allows the operator to immediately begin entering the next block using this same digitize feature.
- The {EDIT & STORE} softkey returns the control to the program display screen. The program block just generated is displayed on the input line of the CRT and may be edited as discussed in chapter 2. This block is then inserted at the location of the block cursor when the [TRANSMIT] key is pressed.

Important: To abort the circular digitize operation press the exit {• } softkey at any time before pressing the {STORE END PT} or {EDIT & STORE} softkeys. The control will be returned to softkey level 4. Then select either another digitize feature (by pressing one of the softkeys) or the control can be returned to the regular edit session (as discussed in section 5.2) by pressing the exit {• } softkey.

5.5
Deleting Program {DELETE
PROGRAM}

To delete part programs stored in memory:

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {DELETE} softkey.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→



CAUTION: Once a program has been deleted from memory it can not be recovered. Abort the delete program operation by pressing the {DELETE NO} softkey.

Here there are two choices:

- Key-in the the program name and press the {DELETE YES} softkey.
- Move the block cursor down until the desired program is in reverse video and press the {DELETE YES} softkey.

(softkey level 3)

↑	DELETE YES	DELETE NO				
---	---------------	--------------	--	--	--	--

- It is possible to delete all programs at once by formatting the RAM disk as described in chapter 2.

5.6
Renaming Programs
{RENAME PROGRAM}

To change the program names assigned to the part programs stored in memory:

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {RENAME PROGRAM} softkey.

(softkey level 2)

↑	ACTI VE PROGRAM	EDI T PROGRAM	RESTR T PROGRAM	DI SPLY PROGRAM	COPY PROGRAM	→
↑	VERI FY PROGRAM	PRGRAM COMENT	DELE TE PROGRAM	RENAME PROGRAM	I NPUT DEVI CE	→
↑	REFORM MEMORY					→

3. Key-in the current program name or cursor down until the desired program is in reverse video. Then type a comma followed by the new program name, and press the {RENAME YES} softkey.

: current - program - name, new - program - name

(softkey level 3)

↑	RENAME YES	RENAME NO				
---	---------------	--------------	--	--	--	--

The old program name will be replaced with the new program name.

Abort the rename operation by pressing the {RENAME NO} softkey.

5.7
Displaying a Program
{DISPLY PRGRAM}

The control has a part program display feature that allows viewing (but not editing) of any part program.

Follow these steps to display a part program stored in the control's memory.

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Select the input device if one must be chosen using the {INPUT DEVICE} softkey (as discussed in chapter 9). This is only necessary if the currently active input device is not the device that the part program to display is currently resident on. Note the default input device is control memory.
3. Move the block cursor to the program to be displayed (if the program is resident in control memory) or key-in the program name (if reading from an input device attached to port A or port B).
4. Press the {DISPLY PRGRAM} softkey.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

5. To scroll the part program blocks, hold down the [SHIFT] key, then press the up or down cursor keys. Note that if the program is being displayed from a peripheral device only the forward direction may be used to display the program.
6. To end the displaying operation press the exit {•} softkey. The display will return to the program directory screen.

5.8
Comment Display {PRGRAM
COMENT}

It is possible to assign a short comment on the program directory screens to each individual program. These comments are used to help identify a program when it is selected for automatic operation or to be edited.

Important: These are not normally the same as a comment block made within a part program. Comment blocks are discussed in section 10.2.3. Note that if a comment block is assigned as the first block of the part program, it will also be displayed on the program directory screen as a comment. Any other comment blocks have no affect on the comment display.

To assign a comment to a program without using a comment block as the first block of the program follow the steps below:

1. Press the {PRGRAM MANAGE} softkey. This displays the program directory screen. Any existing comments previously assigned to a program are displayed to the right of the program name.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Using the up or down cursor keys select the program to add the comment to. The selected program name will be shown in reverse video.
3. Press the {PRGRAM COMENT} softkey. The comment softkey will be displayed in reverse video and the control will display the prompt "COMMENT:" on line 2 of the screen.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELE TE PRGRAM	RENAME PRGRAM	IN PUT DEVI CE	→
↑	REFORM MEMORY					→

If a comment has previously been entered it will be displayed to the right of the "COMMENT" prompt. This comment may be edited using the input cursor as discussed in chapter 2, or the old comment may be deleted by pressing the [DEL] key while holding down the [SHIFT] key.

4. Type in the new comment or edit the old comment by keying it in using the keyboard. Use this table to find the maximum number of characters you can enter:

Control	Max. Number of Comment Characters
9/230, 9/240, 9/260, 9/290 or 9/440	20
9/260, 9/290 or 9/440 with dual processing	14
9/260, 9/290, and 9/440 with EPPS	18

5. When the new comment is correctly displayed on line 2 of the screen press the [TRANSMIT] key. The new comment will be displayed next to the selected program.

5.9 Copying Programs {COPY PRGRAM}

This section discusses making a duplicate of a part program in control memory. To input or output a part program from/to a peripheral device refer to the section on inputting and outputting programs discussed in chapter 9.

To copy part programs stored in memory using different program names:

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the {COPY PROGRAM} softkey.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

3. Cursor down to or enter the program name of the program to be copied, followed by a comma and a name for the duplicate program.

COPY: FROM_NAME, TO_NAME

4. Press the {MEM TO MEM} softkey.

(softkey level 3)

↑	MEM TO PORT A	FROM A TO MEM	MEM TO PORT B	FROM B TO MEM	MEM TO MEM	
---	------------------	------------------	------------------	------------------	---------------	--

The following message will appear:

“FROM: (source program name)

“TO: (new programs name)

Important: The control will display the active communication parameters if one of the communication ports has been chosen. If the communication port parameters do not match that of the peripheral device they must be altered for a successful copy to take place. For details on setting communication port parameters see chapter 9.

5. Select softkey {COPY YES} or {COPY NO}. {COPY YES} copies the part program, while {COPY NO} aborts the copy operation.

(softkey level 4)

↑	COPY YES	COPY NO				
---	-------------	------------	--	--	--	--

6. If you want to verify that the copied program identically matches the original, use the {VERIFY PROGRAM} feature discussed in chapter 9.

5.10
Selecting the Protectable Part
Program Directory

This section contains information on how to select the protectable part program directory. Use this directory to store part programs that you wish to control access to. When part programs that have previously been protected through encryption are downloaded to the control from ODS or the Mini DNC package, they are automatically stored in the protectable part program directory.

Important: The {CHANGE DIR} softkey controls access to the protectable part program directory. This softkey is password protected. You must have the proper password to access this softkey.

If you have access to the {CHANGE DIR} softkey, you can:

- perform any of the program edit functions on the protected programs
- directly select and activate any of the protected programs
- view programs executing from this directory

You can only call a protected program from a main program using a subprogram, G-code macro, or M-code macro call without access to the {CHANGE DIR} softkey.

If you do not have access to the {CHANGE DIR} softkey, you cannot view the executing blocks of the program called from the protected directory.

To access the protectable part program directory:

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

The control displays the main program directory screen:

SELECTED PROGRAM:		
MAIN	DIRECTORY	PAGE 1 OF 1
NAME	SIZE	COMMENT
MAIN	2.3	
012345	14.3	
RRR	9.3	THIS IS A TEST PROG
TEST	3.9	
4 FILES 120.2 METERS FREE		
↑	ACTIVE PROGRAM	EDIT PROGRAM
	RESTRT PROGRAM	DISPLY PROGRAM
	COPY PROGRAM	→

2. Press the {CHANGE DIR} softkey.

(softkey level 2)

↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTRT PROGRAM	DISPLY PROGRAM	COPY PROGRAM	→
↑	VERI FY PROGRAM	PROGRAM COMENT	DELETE PROGRAM	RENAME PROGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY	CHANGE DIR				→

Important: The control does not display the {CHANGE DIR} softkey if your password does not allow you access to it.

The control displays the protectable directory screen:

SELECTED PROGRAM:			
PROTECTABLE	DIRECTORY	PAGE	1 OF 1
NAME	SIZE	COMMENT	
PROTECT1	2.3		
PROTECT2	14.3		
PROTECT3	9.3	THIS IS A PROTECTED	
PROG			
PROTECT4	3.9		
4 FILES 120.2 METERS FREE			
↑	REFORM MEMORY	CHANGE DIR	NCRYPT MODE
	SET-UP NCRYPT		→

The programs in this directory are protected. This means:

- they are processed the same as unprotected programs
- the blocks of protected programs are not displayed during program execution unless you have access to the {CHANGE DIR} softkey (in place of the protected program blocks, the last user non-protected programming block is displayed)
- you can cycle stop during program execution (but you cannot single block through a program)

5.10.1
Protected Program Encryption
and Decryption

Protected program encryption and decryption allow you to encrypt a protected program so that it is unreadable when it is uploaded. Protected programs in encrypted form can only be uploaded or downloaded by using the Upload and Download utilities of ODS or the Mini DNC package.

Use the {NCRYPT MODE} softkey to enable the protected program encryption option. If the control displays the {NCRYPT MODE} softkey in reverse video, the encryption option is enabled. When the encryption option is enabled, protected programs are encrypted when they are uploaded to ODS or the Mini DNC package. When downloading encrypted protected programs to the control, they are decrypted and loaded into the protected program directory.

Protected programs are encrypted or decrypted through a translation process that reads in a program and translates the characters one at a time to another character. Character translation is based on a character encryption/decryption table that you must set-up before programs can be encrypted or decrypted.

Important: If the encryption/decryption table is modified after a protected program is uploaded to ODS or the Mini DNC package, the program can no longer be decrypted when it is downloaded to the control.

To set-up the character encryption/decryption table:

1. Select the protected part program directory.
2. Press the {SET-UP NCRYPT} softkey.

(softkey level 2)

↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTR PROGRAM	DISPLY PROGRAM	COPY PROGRAM	→
↑	VERI FY PROGRAM	PRGRAM COMENT	DELETE PROGRAM	RENAME PROGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY	CHANGE DIR	NCRYPT MODE	SET-UP NCRYPT		→

The control displays the set-up encryption screen:

ENTER A CHARACTER:

=	.	9	D	O	Z
"	/	:	E	P	[
#	0	;	F	Q]
%	1	<	G	R	
&	2	=	H	S	
(3	>	I	T	
)	4	?	J	U	
*	5	@	K	V	
+	6	A	L	W	
'	7	B	M	X	
-	8	C	N	Y	

↑	UPDATE & EXIT	STORE BACKUP	REVRSE FILL			
---	------------------	-----------------	----------------	--	--	--

You must fill in the encryption/decryption table. During the encryption/decryption process, the control uses this table to guide the character substitution that provides an encrypted or decrypted part program. You can fill in this table by using the operator panel keys or the {REVRSE FILL} softkey.

To fill in the encryption/decryption table by using the operator panel keys:

- use the arrow keys to move the cursor to the place where you want to assign an encryption/decryption character
- then enter a character and press the [TRANSMIT] key

You must enter a unique character for each character on the set-up encryption screen.

To fill in the encryption/decryption table by using the {REVRSE FILL} softkey, press the {REVRSE FILL} softkey. Pressing this softkey automatically fills the spaces of the encryption/decryption table in a reverse order as shown below:

ENTER A CHARACTER:

=]	. = Q	9 = F	D = ;	O = 0	Z = #
" = [/ = P	: = E	E = :	P = /	[= "
# = Z	0 = 0	; = D	F = 9	Q = .] = #
% = Y	1 = N	< = C	G = 8	R = -	
& = X	2 = M	= = B	H = 7	S = '	
(= W	3 = L	> = A	I = 6	T = +	
) = V	4 = K	? = @	J = 5	U = *	
* = U	5 = J	@ = ?	K = 4	V =)	
+ = T	6 = I	A = >	L = 3	W = (
' = S	7 = H	B = =	M = 2	X = &	
- = R	8 = G	C = <	N = 1	Y = %	

↑	UPDATE & EXIT	STORE BACKUP	REVRSE FILL			
---	------------------	-----------------	----------------	--	--	--

3. Enter the encryption/decryption characters one at a time, press the {REVRSE FILL} softkey, or modify the reverse filled screen.
4. Press the {UPDATE & EXIT} softkey to update and exit the encryption/decryption table.

(softkey level 3)

↑	UPDATE & EXIT	STORE BACKUP	REVRSE FILL			
---	------------------	-----------------	----------------	--	--	--

When you press the {UPDATE & EXIT} softkey, the control does a compile/check of the encryption/decryption table to determine that no duplicate characters exist and that no characters were left blank.

If a character is:	the control displays:	and moves the cursor:
entered twice	"CHARACTERS CAN ONLY BE ENTERED ONCE"	to the first occurrence of that character in the encryption/decryption table
left blank	"NO LOCATIONS CAN BE LEFT EMPTY"	to the first blank location in the encryption/decryption table

Once the encryption/decryption table is created and you press the {NCRYPT MODE} softkey, protected programs are encrypted when they are uploaded to ODS or the Mini DNC package. When downloading encrypted protected programs to the control, they are decrypted and loaded into the protected program directory.

5.10.2
Storing Encryption/Decryption
Table to Backup Memory

To prevent the encryption/decryption table from being lost during system power failures and battery-backup failures, store it in the control's backup memory by using these steps:

1. Select the protected part program directory.
2. Press the {SET-UP NCRYPT} softkey.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DI SPLY PRGRAM	CO PY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELE TE PRGRAM	RENAME PRGRAM	I NPUT DEVI CE	→
↑	REFORM MEMORY	CHANG E DI R	NCRYPT MODE	SET-UP NCRYPT		→

3. Press the {STORE BACKUP} softkey. The control displays the message "STORING TO BACKUP - PLEASE WAIT" on the CRT until the control has finished storing the encryption/decryption table to its backup memory.

(softkey level 3)

↑	UPDATE & EXIT	STORE BACKUP	REVRSE FI LL			
---	------------------	-----------------	-----------------	--	--	--

END OF CHAPTER

Editing Part Programs Offline (ODS)

6.0 Chapter Overview

You can use the offline development system (ODS) to write or edit part programs. Once completed these part programs may be downloaded from the workstation to the control. Programs that already exist on the control may be uploaded to the workstation for editing or backup. Programs on ODS may be edited using the screen or text editor that is configured in ODS. Enhancements to this feature may be purchased in a Mini-DNC package from Allen-Bradley. If the Mini-DNC package has been purchased as an option refer to its accompanying documentation.

The following sections in this chapter are written with the assumption that:

- the Allen-Bradley offline development system (ODS) has been installed on an IBM PC XT/AT or compatible computer that will be referred to as the workstation
- a compatible screen or text editor has been configured using the Text Editor Setup option of the F5-Configuration menu
- the basics of the ODS system and how it operates is understood by the programmer

For additional information, refer to the ODS manual, publication MCD-5.1.

Important: Be aware that some features discussed here may not be available with your ODS. Some may require the purchase of the Mini-DNC package to be functional.

6.1 Selecting the Part Program Application

Selecting the Part Program application provides access to the part program utilities of ODS. To select the Part Program application:

1. Return to the main menu line of ODS.
2. Press [F3] to pull down the Application menu:

The workstation displays this screen:

Proj: PALTEST		Appl: Upload		Util: Get PAL I/O
[F1 - File]	[F2 - Project]	[F3 - Application]	[F4 - Utility]	[F5 - Configuration]
		AMP (A)		
		PAL (P)		
		I/O Assignments (I)		
		Part Program (R)		
		Upload (U)		
		Download (D)		

3. Press [R] to select the Part Program option.

The status line of the screen displayed by the workstation shows that the Part Program application has been selected.

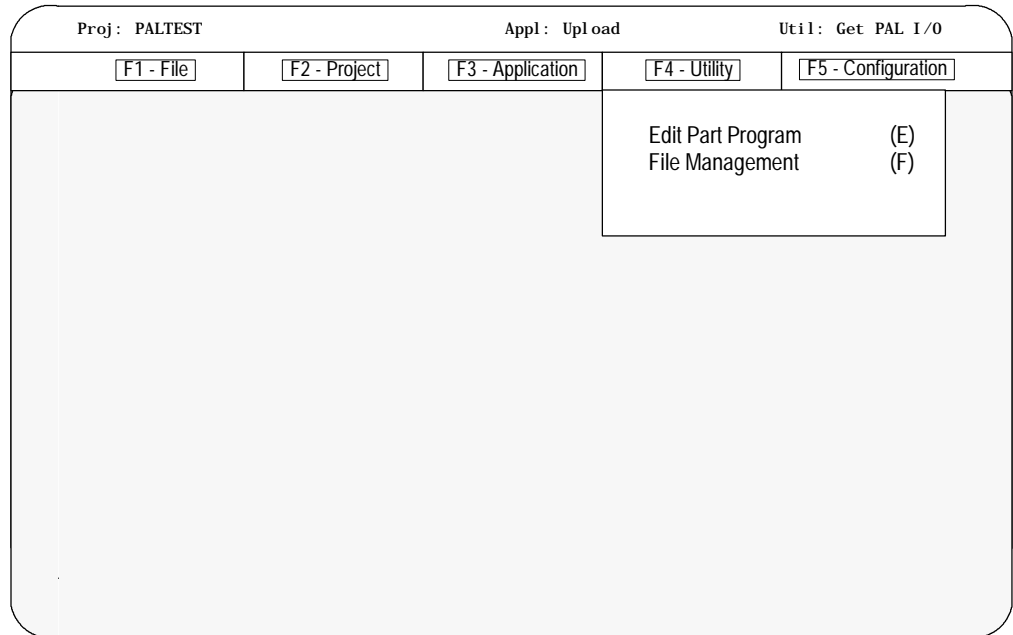
6.2 Editing Part Programs Off Line

The Edit Part Program utility of ODS is used to edit part programs on a workstation. Programs that already exist on the control may be uploaded to the workstation for editing. These programs or programs created using ODS may be edited using the screen or text editor that is configured in ODS. To edit part programs thorough ODS:

1. Select the Part Program Application (see section 6.1)

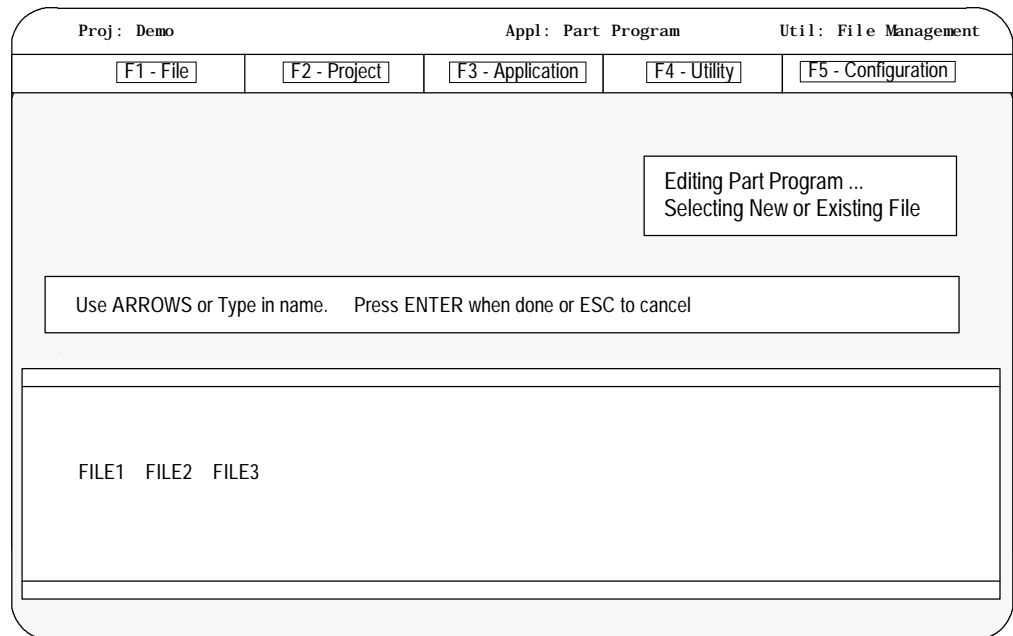
2. Press [F4] to pull down the Utility menu:

The workstation displays this screen:



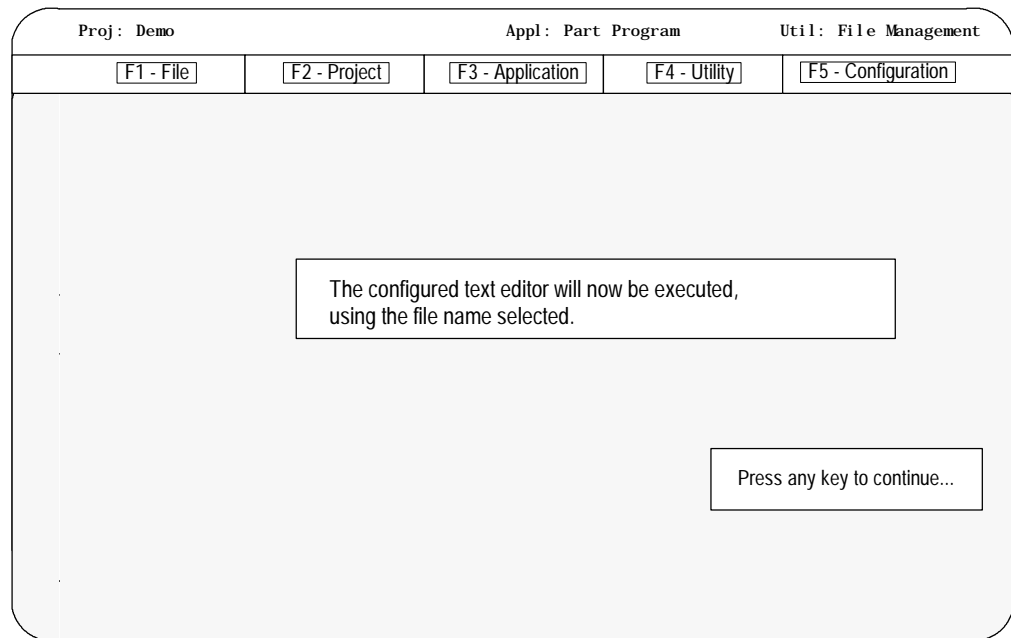
3. Press [E] to select the Part Program option.

The workstation displays this screen:



4. Select a new or existing file. To create a new file, type in the new file name. To open an existing file use the arrow keys to select a file or type in a file name. Press [ENTER] when done, or [ESC] to cancel.

After selecting a file the workstation displays a screen about the text editor:



The configured screen or text editor is used to edit part programs. The editor must be compatible with the ODS operating system. The editor must be configured using the Text Editor Setup option of the F5-Configuration menu at the main menu line. For details on how to use a specific screen or text editor see the documentation provided with the screen or text editor.

Important: The end of block statements, ";" used to separate blocks on the control should not be entered with the screen or text editor. The control will automatically insert the end of block statements ";" at the end of each line (line feed) when the program is downloaded to the control.

The maximum number of programs that you can have is 328. To store a program, it needs to occupy at least 1.3 meters of memory. Use this table to find out how much part program space there is on your system:

If your system has	this is your part program storage
64K	150 meters
128K	300 meters
192K	450 meters
Extended Part Program Storage (EPPS)	5235 meters

Extended Part Program Storage is not available on 9/230 CNCs.

6.3 Interfacing the Workstation with the Control

The following sections require the workstation to be interfaced with the control or storage device. Interface the workstation with the control or storage device using the RS-232 serial interface cable.

This cable is used to connect the RS-232 interface port on the rear of the workstation to Port B on the control or the RS-232 port on the storage device. Refer to your integration manual for more information.

When downloading to a control with a standard MTB panel, which has a serial communication port located on the front left side of the panel, connect the serial interface cable to the serial communication port. This port is connected to port B on the control.

Port B must be configured for serial communication. The serial communication parameters of this port must correspond to the serial communication parameters of the workstation or storage device. To display the serial communication parameters of port B, press the {SYSTEM SUPPORT} softkey on the operator panel, then the {DEVICE SETUP} softkey. If the port A parameters are displayed, use the left or right arrows on the operator panel to change the display from port A to port B parameters.

Make sure the DEVICE configured on this screen for port B is ODS. If it is not configured for ODS, refer to chapter 9 for details on configuring communication parameters. Once the DEVICE type of ODS is configured the default parameters should work for most PC applications.

6.4 Downloading Part Programs from ODS

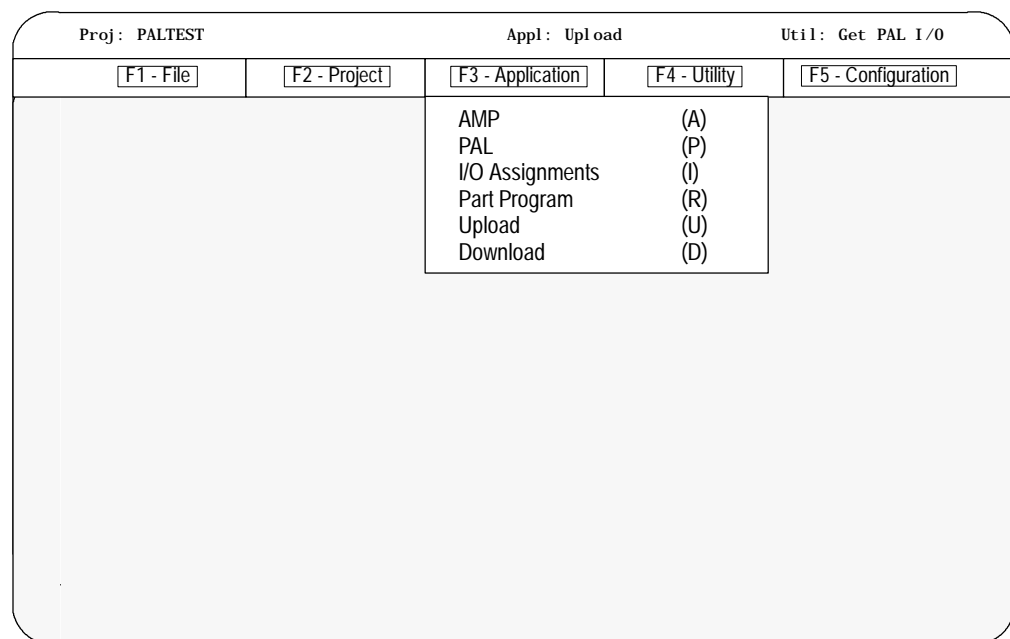
After using the part program edit utility to create or edit a part program file offline, the programmer can download this part program to the control or to a storage device using the Download application of ODS.

Important: When a program is downloaded from ODS to the control, it is automatically inserted into the normal program directory on the control. The control will automatically insert the end of block statements “;” at the end of each line when the program is downloaded to the control.

To download a part program from ODS to the control's memory, follow these steps:

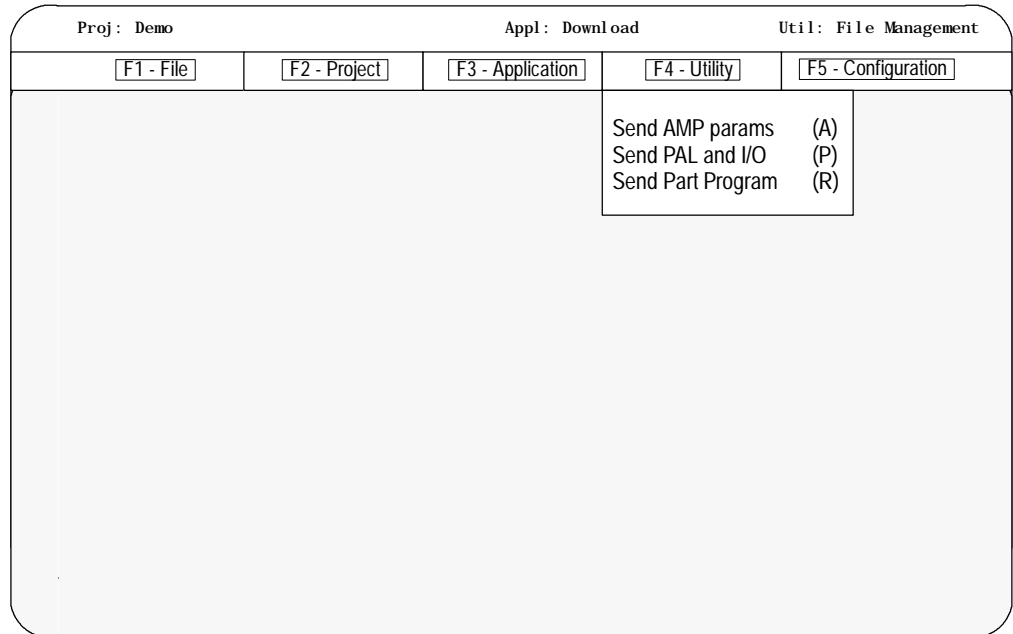
1. Interface the workstation with the control (see section 6.3)
2. Return to the main menu line of ODS
3. Press [F3] to pull down the Application menu.

The workstation displays this screen:



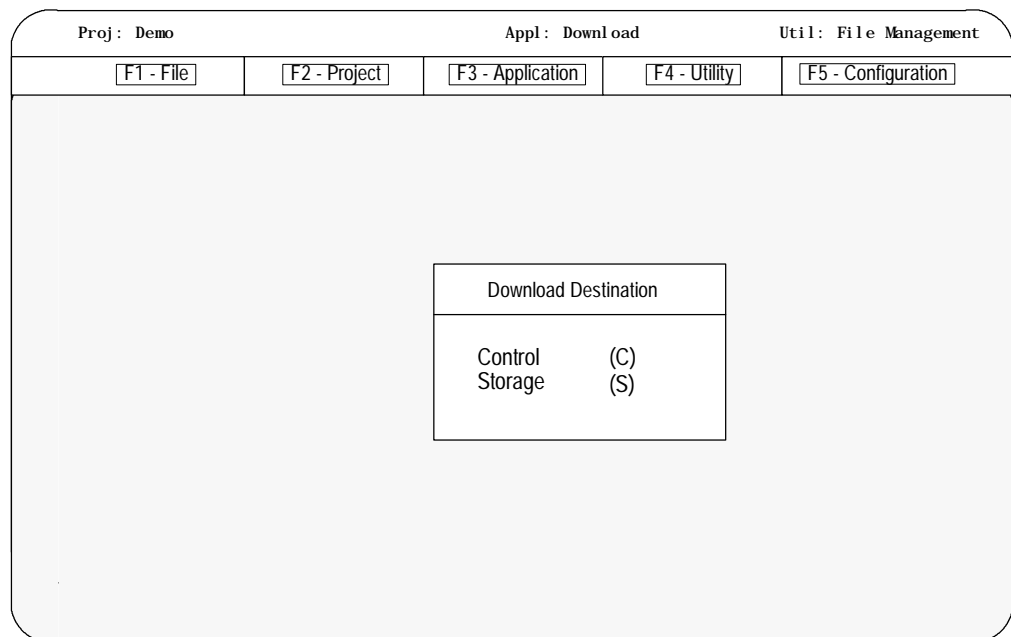
4. Use the arrow keys to highlight the Download application then press [ENTER] or press [D].

5. Press [F4] to pull down the Utility menu.



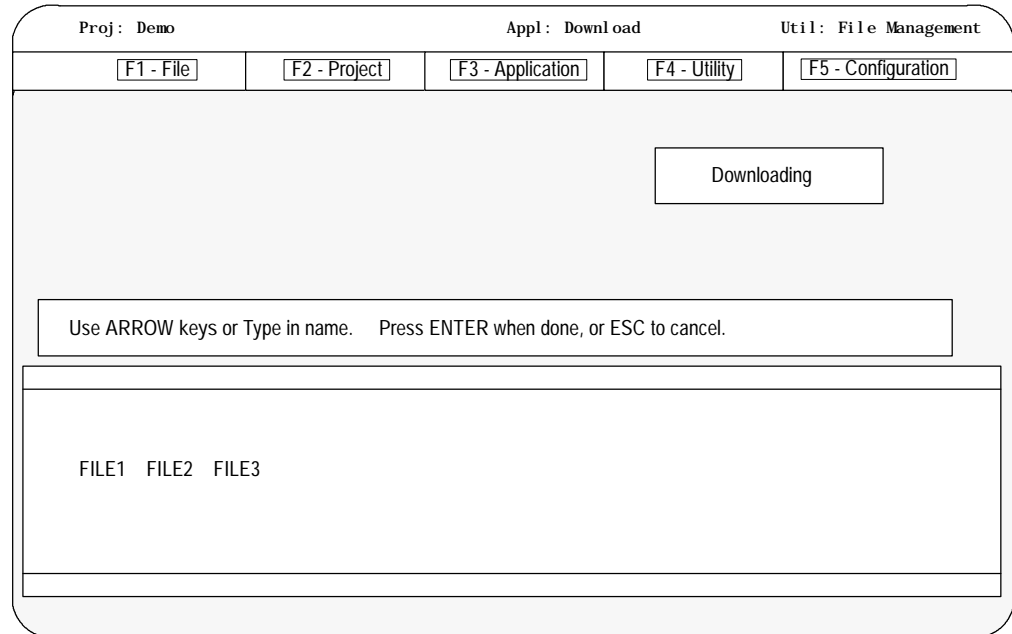
6. Use the arrow keys to highlight the Send Part Program option then press [ENTER], or press [R].

The workstation displays this screen:



7. Use the arrow keys to highlight the download destination or press the letter that corresponds to the download destination. When selected press [ENTER].

The workstation displays the part program files that are stored in the active project directory of the workstation:

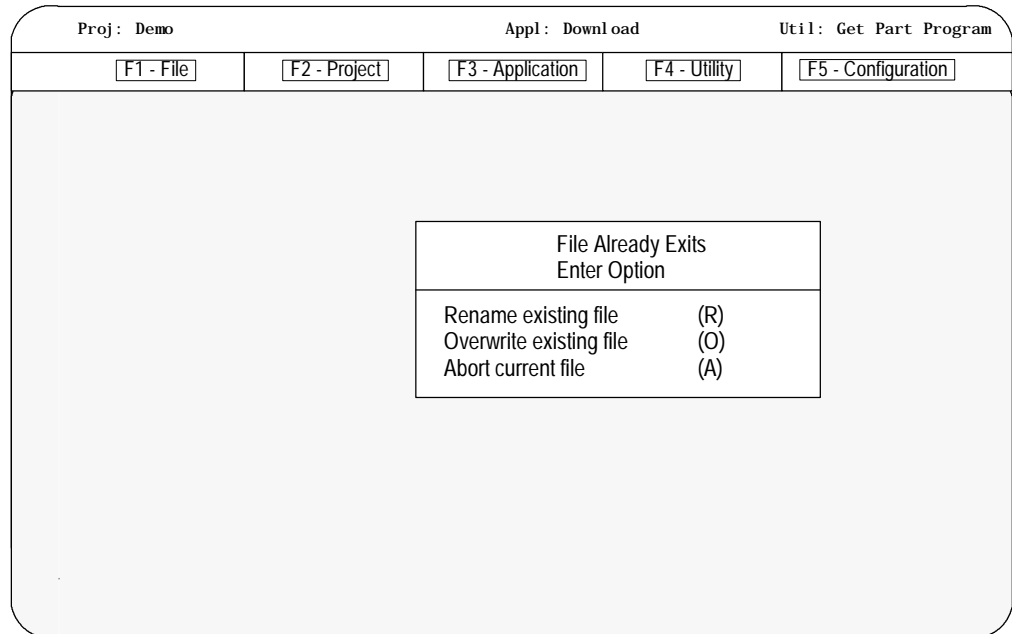


8. Use the arrow keys to highlight the name or type in the part program name to download, then press [ENTER].

When you download a program to the control, the control does not display a message to indicate that a download is taking place. If you download a large program it may take several minutes for the control to complete the download. You may want to go to the directory screen and monitor the download. As the program downloads, the control updates the size of the program.

Important: It is possible to upload more than one part program using wildcards ("*" or "?") in place of all or part of a file name. Refer to the workstation's DOS manual for additional information on using wildcards.

If the selected part program file name already exists on the control, the workstation displays this screen:



Important: The currently active or open part program on the control can not be renamed or overwritten during a download procedure.

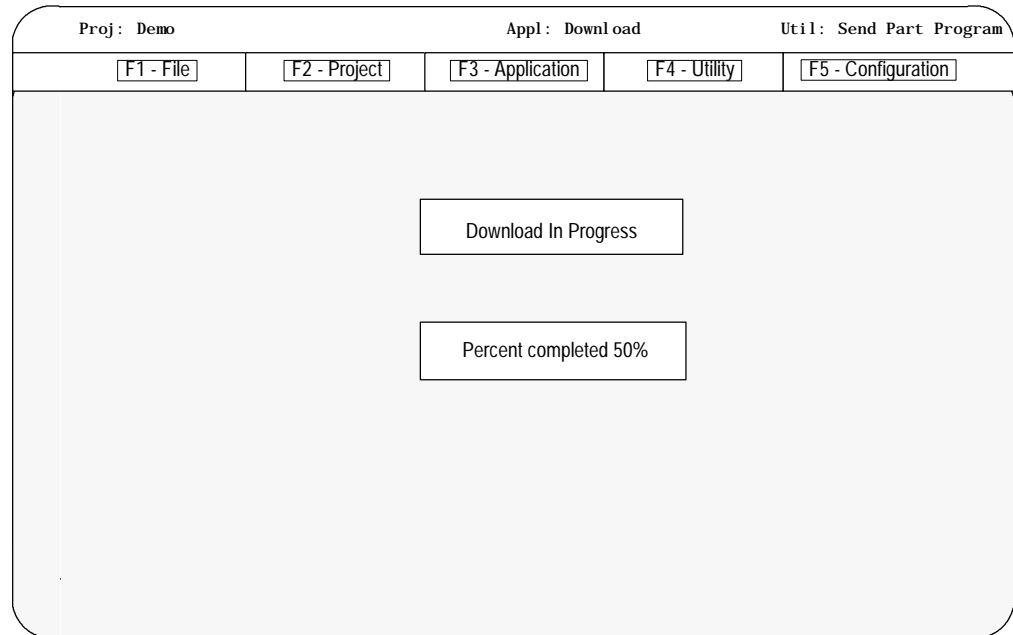
If the rename option is selected, the workstation renames the existing file, which has the same name as the file being uploaded, on the workstation. The workstation displays the part program files stored on the workstation. Type in the new name for the existing part program on the control.

If the Overwrite existing file option is selected, the part program file being downloaded overwrites the file having the same name on the control.

If the Abort current file option is selected, the download process is discontinued and the workstation prompts the programmer for additional files to download.

Important: If a wildcard was entered in place of a file name, the Abort option is repeated for each file that matches the wildcard. Pressing the [ESC] key quits the abort wildcard process.

After selecting the Rename or Overwrite option, or if the file being downloaded did not already exist on the control, the workstation displays this screen:



The percentage of the download process that has currently been completed is displayed on the screen. This value is updated continually throughout the download process.

When you download a program to the control, the control does not display a message to indicate that a download is taking place. If you download a large program it may take several minutes for the control to complete the download. You may want to go to the directory screen and monitor the download. As the program downloads, the control updates the size of the program.

When the download process is complete, you see this screen:

Proj: Demo		Appl: Download		Util: Send Part Program	
F1 - File	F2 - Project	F3 - Application	F4 - Utility	F5 - Configuration	
<p>Download Complete</p>					
<p>Download Another File?</p>					
Yes		(Y)			
No		(N)			

9. Select "Yes" or "No." If "Yes" is selected, the programmer will be prompted through the download procedure again. If "No" is selected, the workstation returns to the main menu line of ODS.

If the workstation is unable to complete the download procedure in the allotted time frame, it displays this screen:

Proj: Demo		Appl: Download		Util: Send Part Program	
F1 - File	F2 - Project	F3 - Application	F4 - Utility	F5 - Configuration	
<p>A time-out occurred ...</p>					
<p>Press any key to continue ...</p>					

Press any key to return to the ODS main menu.

6.5 Upload Part Programs to ODS

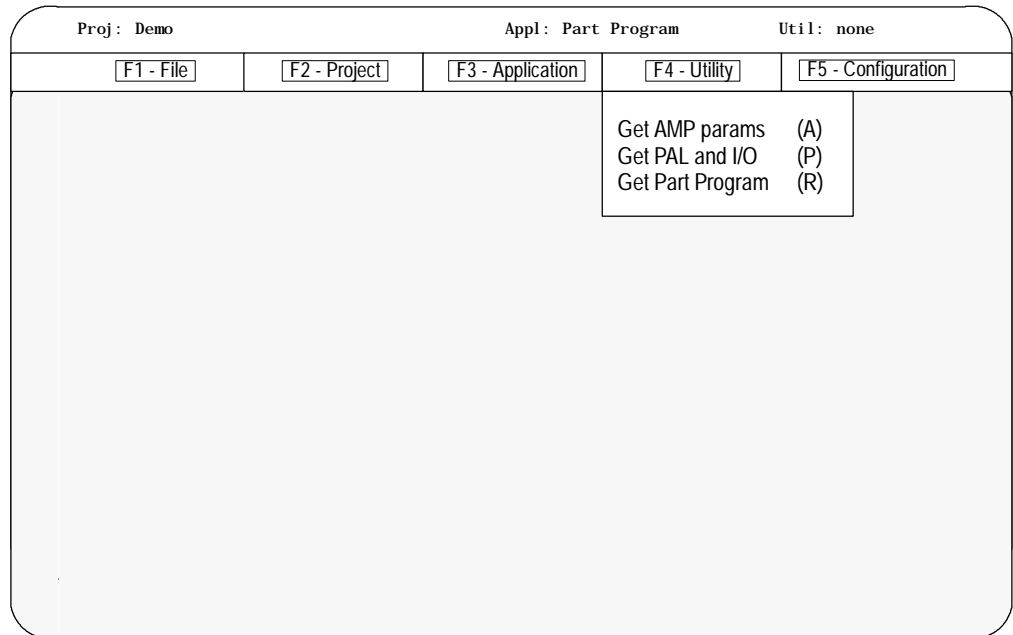
The programmer can upload a part program from the control's memory to the workstation using the Upload application of ODS. This allows the part program to be edited or stored on the workstation.

1. Interface the workstation with the control (see section 6.3)
2. Return to the main menu line of ODS
3. Press [F3] to pull down the Application menu.

The workstation displays this screen:

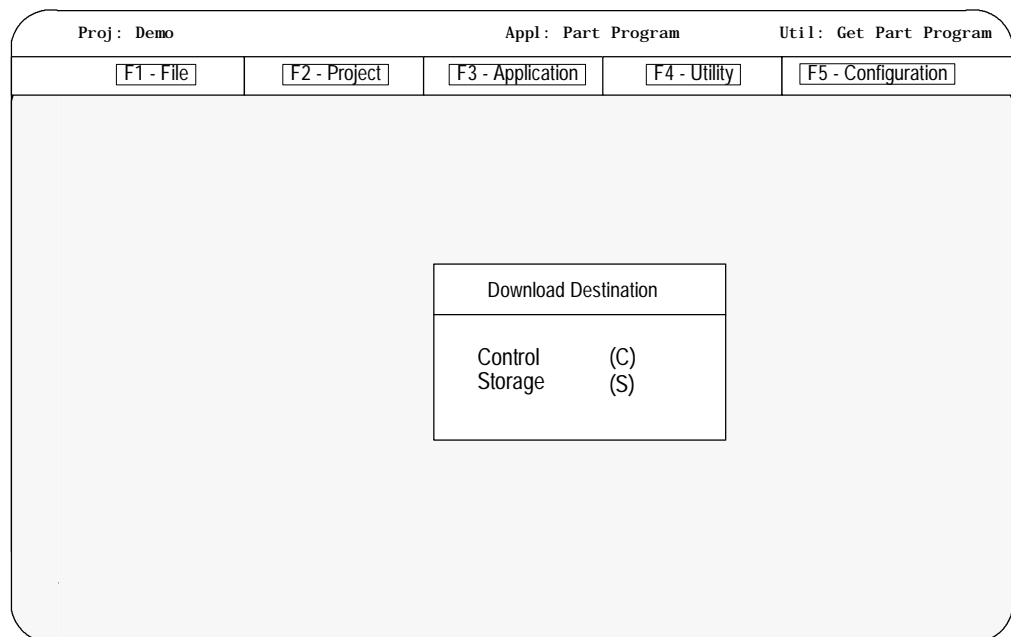
Proj: PALTEST		Appl: Upload		Util: Get PAL I/O
F1 - File	F2 - Project	F3 - Application	F4 - Utility	F5 - Configuration
		AMP (A)		
		PAL (P)		
		I/O Assignments (I)		
		Part Program (R)		
		Upload (U)		
		Download (D)		

4. Use the arrow keys to highlight the Upload application then press [ENTER] or press [U].
5. Press [F4] to pull down the Utility menu.



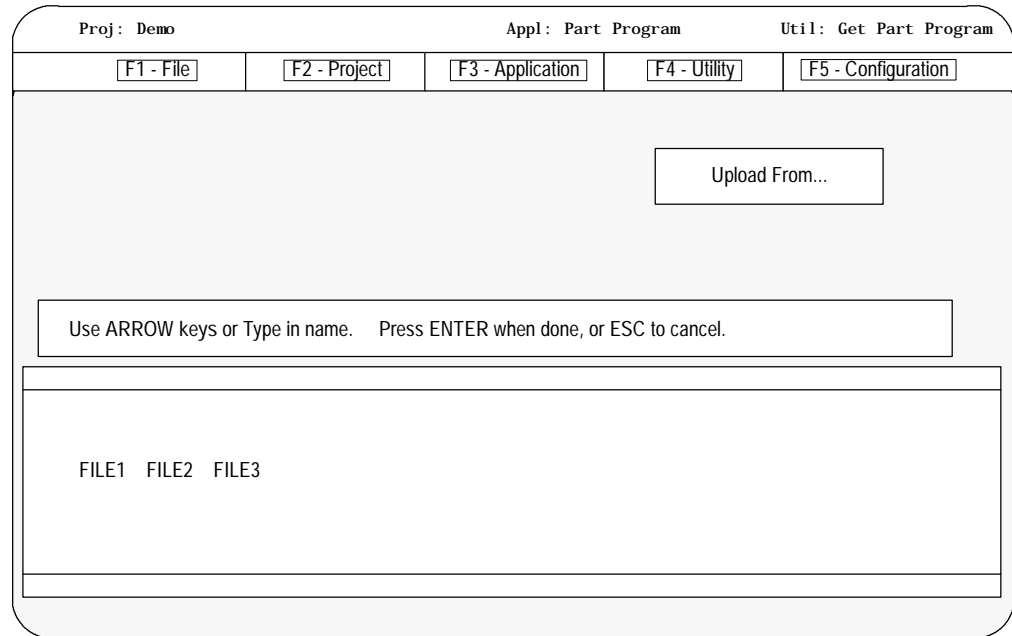
6. Use the arrow keys to highlight the Get Part Program option then press [ENTER], or press [R].

The workstation displays this screen:



7. Use the arrow keys to highlight the upload origin then press [ENTER] or press the letter that corresponds to the upload origin.

The workstation displays the part program files that are stored on the control or storage device:



8. Use the arrow keys to highlight the name of the part program to be uploaded to the workstation or type in the part program name, then press [ENTER].

When you upload a program from the control, the control does not display a message to indicate that an upload is taking place. If you upload a large program it may take several minutes for the upload to complete. If you try to edit the program while it is uploading you see an error message that says the program is already open. You have to wait until the upload is complete to edit the program.

Important: It is possible to upload more than one part program using wildcards ("*" or "?") in place of all or part of a file name. Refer to the workstation's DOS manual for additional information on using wildcards.

If the selected part program already exists on the workstation, the workstation displays this screen:

The screenshot shows a workstation interface with a title bar containing 'Proj: Demo', 'Appl: Upload', and 'Util: Get Part Program'. Below the title bar are five menu buttons: 'F1 - File', 'F2 - Project', 'F3 - Application', 'F4 - Utility', and 'F5 - Configuration'. The main area of the dialog box contains a central box with the following text:

File Already Exists	
Enter Option	
Rename existing file	(R)
Overwrite existing file	(O)
Abort current file	(A)

If the Rename option is selected, the workstation renames the existing file, which has the same name as the file being uploaded, on the workstation.

The workstation displays the part program files stored on the workstation:

The screenshot shows a workstation interface with a title bar containing 'Proj: Demo', 'Appl: Upload', and 'Util: Get Part Program'. Below the title bar are five menu buttons: 'F1 - File', 'F2 - Project', 'F3 - Application', 'F4 - Utility', and 'F5 - Configuration'. The main area of the dialog box contains two input fields: 'Enter new name:' and 'Rename To...'. Below these fields is a list box containing the text 'FILE1 FILE2 FILE3'.

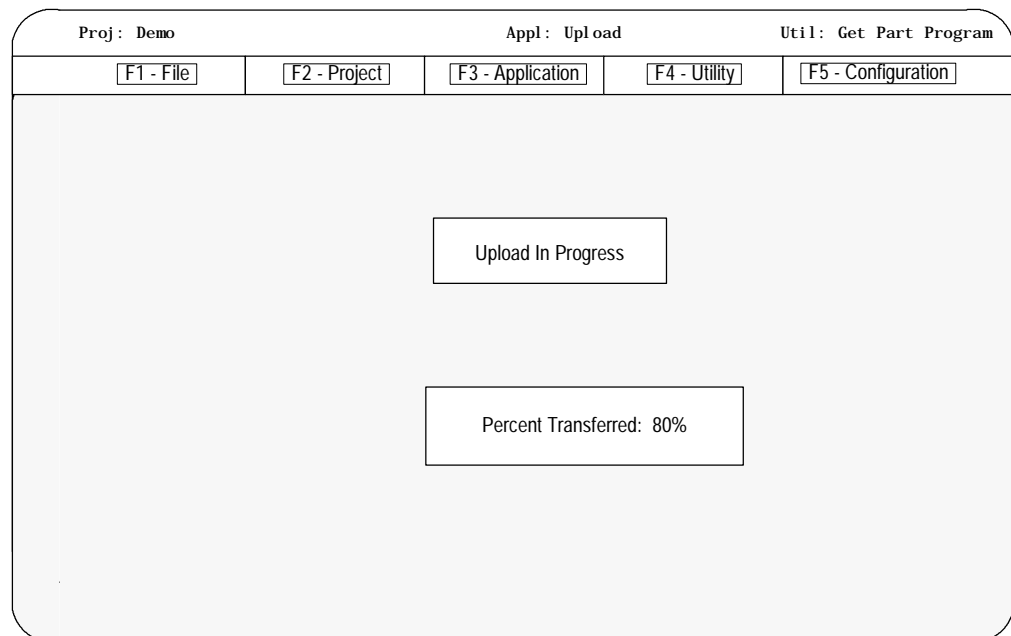
9. Type in the new name for the existing part program file on the workstation.

If the Overwrite option is selected, the part program file being uploaded overwrites the file having the same name on the workstation.

If the Abort option is selected, the upload process is discontinued and the workstation prompts the programmer for additional files to upload.

Important: If a wildcard was entered in place of a file name, the Abort option is repeated for each file that matches the wildcard. Pressing the [ESC] key quits the abort wildcard process.

If the name of the part program that was entered does not exist on the workstation or the Overwrite option was selected, the workstation displays this screen:



The percentage of the upload process that has currently been completed is displayed on the screen. This value is updated continually throughout the upload process.

When you upload a program from the control, the control does not display a message to indicate that an upload is taking place. If you upload a large program it may take several minutes for the upload to complete. If you try to edit the program while it is uploading you see an error message that says the program is already open. You have to wait until the upload is complete to edit the program.

After the part program has been uploaded to the workstation, the workstation displays this screen:

Proj: Demo		Appl: Upload		Util: Get Part Program	
[F1 - File]	[F2 - Project]	[F3 - Application]	[F4 - Utility]	[F5 - Configuration]	
Upload Complete					
Upload Another File?					
Yes		(Y)			
No		(N)			

Select “Yes” or “No.” If “Yes” is selected, the programmer will be prompted through the upload procedure again. If “No” is selected, the workstation returns to the main menu line.

END OF CHAPTER

Running a Program

7.0 Chapter Overview

This chapter describes how to test a part program and execute it in automatic mode. Major topics include:

- selecting special running conditions
- program selection options
- starting and stopping test and automatic operation
- program checking modes
- automatic operation mode
- interrupted program recover {RESTRT PRGRAM}
- jog retract
- block retrace

7.1 Selecting Special Running Conditions

The following subsections describe some of the available functions on the control that affect how the control executes a program. The use of these “special running conditions” is optional. They are activated either through the MTB panel, through programming, through the {FRONT PANEL} softkey, or some combination of the three.

7.1.1 Block Delete

When programming a slash “/” followed by a numeric value (1-9) anywhere in a block, the control will skip (not execute) all remaining programmed commands in that block if a corresponding softkey, or optionally installed switch on the MTB panel, is activated. If the “block delete type” parameter in AMP is set to “delete whole”, then the entire block will be deleted regardless of the position of the block delete character. For details on the block delete feature see chapter 10.

To activate the block delete feature the system installer may have installed a switch corresponding to a block delete number (refer to documentation prepared by the system installer), or use the {FRONT PANEL} softkey. If using the {FRONT PANEL} softkey, only block delete 1 (/ or/1) is available.

7.1.2 Miscellaneous Function Lock

When the MISCELLANEOUS FUNCTION LOCK is made active, the control displays M-, second auxiliary functions (B-codes), S-, and T-codes in the part program, except for M00, M01, M02, M30, M98, and M99.

To activate the MISCELLANEOUS FUNCTION LOCK feature the system installer may have installed a switch corresponding to the MISCELLANEOUS FUNCTION LOCK feature (refer to documentation prepared by the system installer), or use the {FRONT PANEL} softkey as discussed in chapter 2.

7.1.3 Sequence Stop {SEQ STOP}

Use this feature to cause automatic program execution to stop after a specified block. This block is determined by assigning its sequence number (N word) as the sequence stop block. This sequence number may be entered before or after part program execution begins. If this sequence number is entered after program execution begins, it must be entered before the control has executed that block. If it is not entered before the block is executed it will be ignored and execution will continue as normal.

Automatic execution will stop after the sequence stop block is completed. The control is placed in cycle stop. To resume execution from the current position in the program press the <CYCLE START> button.

Important: Once a sequence stop number is entered for a program it remains active for all programs that are executed until replaced with a different sequence stop number, or power is lost. Not entering a value for the sequence stop number or entering a value of zero will result in the sequence stop function being canceled.

If a subprogram or macro is called that also contains a sequence number that corresponds to the sequence stop number, program execution will stop in the subprogram or macro at the corresponding sequence number.

To enter a sequence number to stop execution:

1. Press the {PROGRAM MANAGE} softkey. Note that a program must have already been selected for automatic execution as discussed in section 7.3.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWIT CH LANG		→

2. Press the {ACTIVE PROGRAM}.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

3. Press the {SEQ STOP} softkey.

(softkey level 3)

↑	DE- ACT PRGRAM	SEARCH	MI D ST PRGRAM	T PATH GRAPH	T PATH DI SABL	→
↑	SEQ STOP				TI ME PARTS	→

4. Key in the sequence number where automatic operation in the part program is to stop, then press the [TRANSMIT] key.

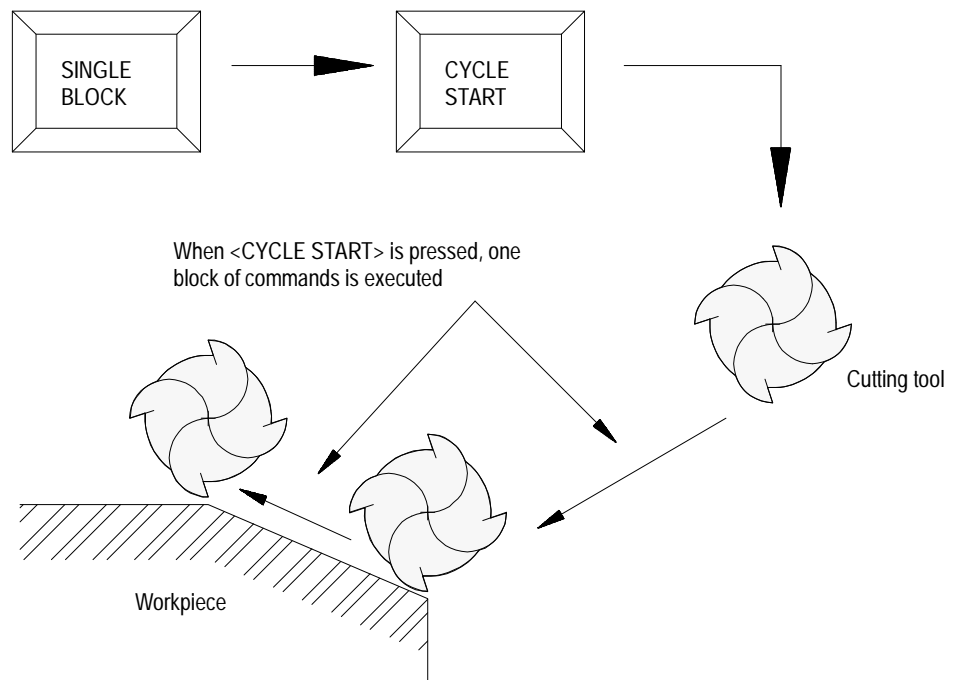
Important: The control will stop automatic operation after it completes the commands in the block.

5. Press the <CYCLE START> button to continue execution of the program from the point at which program execution was stopped.

7.1.4 Single Block

In single block mode, the control executes the part program block by block. Each time the <CYCLE START> button is pressed, the control executes one block of commands in the part program when in single block mode.

Figure 7.1
Single Block



To activate the single block function, press the <SINGLE BLOCK> button. The light inside the button lights up when active.

If the <SINGLE BLOCK> button is pressed while the control is running a part program in the automatic or MDI mode, the control activates the single block function after it completes the commands in the block that is currently being executed.

The <SINGLE BLOCK> button is a toggle switch. If pressed again while the single block function is active, the function is canceled and the light inside the button turns off. The remaining program blocks can be executed as normal by pressing the <CYCLE START> button.

7.2
Selecting a Part Program
Input Device

Before selecting a part program it is necessary to tell the control where this part program is currently residing. The program can reside:

- in the control's RAM memory
- on a peripheral device attached to port A such as a tape reader (refer to system installers documentation)
- on a peripheral device attached to port B such as a tape reader (refer to system installers documentation)

Important: If selecting a peripheral device attached to one of the two communication ports (A or B) that port must have previously been configured to communicate to that specific peripheral device. Refer to chapter 9 for details on communications setup for ports A and B. Selecting a program from the control's memory is the default configuration of the control. If port A or B is never selected, the input device should remain at its default of control memory and the following procedure should be unnecessary.

To select the part program input device:

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {INPUT DEVICE} softkey.

(softkey level 2)

↑	ACTI VE PROGRAM	EDI T PROGRAM	RESTR T PROGRAM	DI SPLY PROGRAM	COPY PROGRAM	→
↑	VERFY PROGRAM	PRGRAM COMENT	DELETE PROGRAM	RENAME PROGRAM	I NPUT DEVI CE	→
↑	REFORM MEMORY					→

3. Press the softkey corresponding to the location the part program is to be read from, {FROM PORT A} , {FROM PORT B}, or {FROM MEMORY}.

(softkey level 3)

↑	FROM PORT A	FROM PORT B	FROM MEMORY			
---	----------------	----------------	----------------	--	--	--

To activate a part program, it must be selected as discussed in section 7.3.

7.3 Selecting a Program

To select a program for automatic execution, follow the steps below.

Important: Consider the following when selecting a program:

- The control will not allow the selection of a program for execution if that program file is still open for editing. See chapter 5 on how to exit the edit mode.
- Note that the system installer may have written PAL to allow some other method of part program selection. Refer to the documentation prepared by the system installer for additional information.
- Before selecting a part program to activate the input device must have been previously selected as discussed in section 7.2. Note that the default condition is to select the part program out of control memory.
- If a program was previously activated and not deactivated, the control will not allow a different part program to be selected. If a different part program is desired, you must first deactivate the active program as discussed in section 7.4.

To select a program for automatic execution:

1. Press the {PROGRAM MANAGE} softkey. The control displays the program directory screen as shown in Figure 7.2.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

Figure 7.2
Part Program Directory

SELECTED PROGRAM			
DI RECTORY			PAGE 1 OF 1
NAME	SIZE	COMMENT	
TEST	AE 3.9		
012345	1.3	SUB TEST 1	
MAIN	1.3		
SHAFT2	1.3	THIS IS A TEST PROGRAM	
XXX	1.3		
5 FILES 137.8 METERS FREE			
↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTRT PROGRAM
	DISPLY PROGRAM	COPY PROGRAM	→

Important: If a program is active, an “A” appears to the right of the program name. If a program is being edited, an “E” appears to the right of the program name. Figure 7.2 shows program TEST as active and being edited. Make sure no part program is currently already active. If a part program is currently active the control will not allow the selection of a different part program until the currently active one is deactivated (refer to section 7.4 on deactivating a part program).

2. Key in the name of the part program to activate. Not that if the program is being selected from control memory the • or • cursor keys may be used to select the program to activate from the directory screen.

If the part program is being selected from a peripheral device (attached to port A or port B) the part program name must be manually keyed in. Also make sure the peripheral device is on and ready to output the part program. It may be necessary to perform a search operation when selecting a program from a peripheral device to find the tape location containing the desired part program.

3. Press the {ACTIVE PRGRAM} softkey to activate the selected program. The control displays the part program name, followed by the first few blocks of the selected program.

Important: Note that the following softkey level 2 indicates that the control is using control memory as an input device. If the input device is some device other than control memory, some of these softkeys will not be available.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

Important: Before the program can be executed the control must be placed in automatic mode.

7.4 Deselecting a Part Program

It is sometimes necessary to deactivate a part program that has been selected for automatic execution. This is necessary when selecting a different part program for automatic execution. To do this follow these steps:

1. Press the {PROGRAM MANAGE} softkey. The control displays the program directory screen as shown in Figure 7.2.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWITCH LANG		→

Important: If a program is active, an “A” appears to the right of the program name. If a program is being edited, an “E” appears to the right of the program name. Figure 7.2 shows program TEST as active and being edited.

2. Press the {ACTIVE PROGRAM} softkey. The control displays the first few blocks of the currently active program.

(softkey level 2)

↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTRT PROGRAM	DI SPLY PROGRAM	COPY PROGRAM	→
↑	VERFY PROGRAM	PROGRAM COMENT	DELETE PROGRAM	RENAME PROGRAM	INPUT DEVICE	→
↑	REFORM MEMORY					→

3. If the program selected is not the desired active program, press the {DEC-ACT PROGRAM} softkey. The control deactivates the part program and returns to the directory screen.

(softkey level 3)

↑	DE- ACT PROGRAM	SEARCH	MID ST PROGRAM	T PATH GRAPH	SEQ STOP	→
↑					TIME PARTS	→

7.5 Program Search {SEARCH}

Use the Program Search feature to begin program execution from some block other than at the beginning of the program. This feature requires the operator to establish the necessary G, M, S, F, and T words, work coordinate offsets, etc., that should be active for that block's execution.

The control is capable of starting a program at a chosen block and establishing any previous G, M, S, F, and T words work coordinate offsets, etc., that were established in previous blocks using the search with memory feature. For details, see section 7.4.

The program search feature is not effective for subprograms and paramacos; only blocks that are in the main program can be searched.

To perform a program search operation:

1. Press the {PROGRAM MANAGE} softkey. Note the program to search must have previously been selected for automatic execution as discussed in section 7.2.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITC H LANG		→

2. Press the {ACTIVE PROGRAM} softkey.

(softkey level 2)

↑	ACTI VE PROGRAM	EDI T PROGRAM	RESTRT PROGRAM	DI SPLY PROGRAM	COPY PROGRAM	→
↑	VERFY PROGRAM	PRGRAM COMENT	DELETE PROGRAM	RENAME PROGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

3. Press the {SEARCH} softkey.

(softkey level 3)

↑	DE- ACT PRGRAM	SEARCH	MID ST PRGRAM	T PATH GRAPH	SEQ STOP	→
↑					TIME PARTS	→

4. You can search 6 ways:

To search:	Press this softkey:
for a sequence number	{N SEARCH}
for an O word	{O SEARCH}
for the end of each block	{EOB SEARCH}
the program one line at a time	{SLEW}
for a specific character string	{STRING SEARCH}
for the beginning of your next program	{NEXT PRGRAM}

The {NEXT PRGRAM} softkey is available only if your input device is configured as a tape reader. Refer to section 7.2 on input device selection.

(softkey level 4)

↑	N SEARCH	O SEARCH	EOB SEARCH	SLEW	STRING SEARCH	→
↑	NEXT PRGRAM					

5. When using the SLEW or the EOB search options, press the {FORWRD} or {REVRSE} softkeys to move to the next or previous blocks in the program. Press the {TOP OF PRGRAM} softkey to return to the top of the program (the beginning of the first block). Press the {EXIT} softkey when the desired block is found. The selected block will be the next block for automatic execution.

Important: When performing an EOB search, the search is executed from the beginning of the part program, NOT from the point of display.

When using the N search, O search, or STRING search features, first key in the desired N number, O number, or character string to search for. After it has been keyed in, press the [TRANSMIT] key to start the search. Press the {FORWRD} or {REVRSE} softkeys to search for the entered value in the forward or reverse direction. Press the {TOP OF PRGRAM} softkey to return to the top of the program (the beginning of the first block). Press the {EXIT} softkey when the desired block is found. The selected block is the next block for automatic execution.

If no number is keyed in for an N or O search, the control simply searches for the next N or O word in the program.

When you press the {NEXT PRGRAM} softkey, the control first searches for a valid program end code (see setting communications, chapter 9). After it finds the program end code, it advances to the program start code of the next program. If the current program is the last program on the tape, the message "SERIAL COMMUNICATION ERROR #5" appears on the screen indicating a time-out error.

Important: If performing a STRING search, program execution begins at the beginning of the block that contains the desired character string. This is not necessarily the location of the string in the program block.



CAUTION: It may be necessary to position the cutting tool at a location that allows this block to execute without damaging the workpiece or cutting tool. This may be done through a manual operation or through MDI.

7.6 Search With Recall {MID ST PROGRAM}

Use the Mid-Start Program feature to begin program execution from some block other than the first block of the program. This feature will scan the program as it searches and from within the search area:

- send to PAL the last programmed modal G-codes from each modal group.
- send to PAL the last programmed modal M-codes from each modal group and set its associated PAL strobe (non-modal codes including user-defined M-codes are not sent to PAL).
- send to PAL the last programmed T code and set its associated PAL strobe
- send to PAL the last programmed auxiliary function code (B-word) and set its associated PAL strobe
- send to PAL the last programmed spindle commanded speed and set its associated PAL strobe
- resolve paramacro equations and assign paramacro variable values (note on dual process systems shared paramacro variables can be different than expected depending on the state of the part program in the other process, also equations that use PAL paramacro variables may also evaluate differently since no paramacro interaction with PAL occurs during a search operation).
- establishes any work coordinate system, including all offsets and rotations to the work coordinate system.

Important: Incremental moves that occur during a program search with recall operation, are always referenced from the last known absolute position in the part program. If no absolute position is specified in the searched part program blocks, the control will use the current axis position as the start point for incremental moves.

When a search with recall is performed, the control finds a character string or sequence number in a specific block for execution to begin from. Note that execution always begins from the beginning of the block, regardless of the location in the block of the searched string or sequence number. This searched block must be a block that would normally be executed during the full programs execution (a block that would be skipped by some means such as a jump, etc., cannot be searched for).

The program search with recall feature maybe used to search into any subprograms or paramacros that may be contained in the main program. This is provided of course, that the searched block is in the path of normal program execution.

Important: The search with recall feature will not:

- send PAL nonmodal M-codes including user-defined groups 0 - 3, group 4, group 5, and group 6 M-codes.
- on dual process systems, halt execution for synchronization codes.
- read from or write paramacro variables to PAL
- on dual process systems, shared paramacro variables between processes may not be evaluated as desired depending on the status of the other process.

Important: This feature will not search into any cycle that calls a set of profile blocks (typically specified with the P- and Q-word in the cycle). Refer to the description of your cycle for details on profile blocks.

- send to PAL gear change requests based on spindle speed

To perform a program search with recall, follow these steps:

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {ACTIVE PROGRAM} softkey.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTRT PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

Make sure that the program to search is the currently active program. If it is not, select it for automatic execution as discussed in section 7.3.

3. Press the {MID ST PRGRAM} softkey.

(softkey level 3)

↑	DE- ACT PRGRAM	SEARCH	MID ST PRGRAM	T PATH GRAPH	SEQ STOP	→
↑					TIME PARTS	→

4. To search for a sequence number press the {SEQ # SEARCH} softkey. To search for a character string press the {STRING SEARCH} softkey.

(softkey level 4)

↑	SEQ # SEARCH	STRING SEARCH				→
---	-----------------	------------------	--	--	--	---

5. Key in the desired character string or sequence number to search for and press the [TRANSMIT] key. The control locates an @ symbol to the left of the block immediately before the block that automatic execution begins from.

If this is not the block to begin execution from press either the:

{CONT} softkey to continue to search for the entered character string or sequence number.

{TOP OF PRGRAM} to return to the first block in the program.

{QUIT} softkey to end either the sequence number search or the character string search operations.

(softkey level 5)

↑	CONT	TOP OF PRGRAM	QUIT	EXIT		→
---	------	------------------	------	------	--	---



CAUTION: When you exit a mid-program start (search with memory), M- and S-codes are sent to PAL. If, during normal execution, that program activated a spindle, mid-program start may also start it. To prevent this, your system installer can use the \$BLKSTAT flag, which indicates the control is in mid-program start.

6. Press the {EXIT} or the {EXIT & MOVE} softkey once the program is at the desired location.

{EXIT} - Use this softkey if the tool is at the exact location for execution of the searched program block. While the control searches for your starting block it performs calculations to determine what the absolute position of the axes should be before your selected block is executed. If the cutting tool is not at this position when you press the {EXIT} softkey, the control aborts the mid-start operation. When this occurs the control displays the message "AXIS POSITION INCORRECT".

{EXIT & MOVE} - Use this softkey if the tool is not at the exact location for execution of the searched block. Be aware that the absolute position of the axes necessary at the start of the searched block is dependant on the previous blocks. There can be offsets activated or incremental moves that can make it difficult for you to determine the exact absolute starting point for the axes. The control generates a motion block to place the tool at the position necessary to generate the intended contour when the searched block is executed. The block generated is always a linear move with a feedrate based on the last motion block prior to the searched block. If the last motion block was a cutting move with a feedrate, then the generated block will be a linear move at that cutting feedrate. If the last motion block was a rapid move, that the generated block will be a linear move at the rapid feedrate.



WARNING: It is the responsibility of the system installers PAL application to make sure proper activation of all necessary machine functions has occurred before allowing a search with recall operation to continue. You should verify that all machine functions are in the correct state before pressing <CYCLE START>.

A message is generated telling the operator to check that all generated modal codes are correct. This message reads "WARNING - VERIFY MODAL CODES". These modal codes should be checked on the G- or M-code status screen.

The control begins program execution from the selected block when you press the <CYCLE START> button. If you have pressed the {EXIT & MOVE} button the control first executes the generated block to place the tool at the proper location. If you do not want the control to execute this generated block you can perform a block reset to abort the generated block.

Program interrupts that are enabled in blocks prior to the searched block (M96L__P__), are active and available for execution once the active program begins execution. Interrupts can not be executed while the mid-program search operation is taking place.

7.7 Basic Program Execution

After a program is written or loaded into the control, it should be thoroughly tested before a part is mounted and cut. The control offers three distinct testing modes in addition to fully automatic operation.

These modes are briefly described below in the order that they would normally be implemented.

- **Quick Check** (see section 7.7.1) -- This mode is a basic syntax checker for a part program. It checks that proper format and syntax has been followed. If Quick Check has been purchased with the graphic option, it can be used to determine if the proper tool paths are being generated (see chapter 8). No actual axis motion is produced during Quick Check however offsets and coordinate system shifts are performed.
- **Axis Inhibit** (see section 7.7.2) -- The axis inhibit mode allows the execution of a program to take place without moving a selected axis or axes. Programmed feedrates are active and the program executes in approximately the same time as normal program execution. Axis motion is simulated for any of the non-moving axes by all of the position displays changing at the programmed feedrate. Graphics are available on the active program graphics screen (chapter 8).
- **Dry Run** (see section 7.7.3) -- Dry run simply replaces all F word feedrates in a program with a special feedrate determined by the system installer in AMP. Graphics are available on the active program graphics screen (chapter 8).
- **Part Production/Automatic** (see section 7.7.4) -- In automatic mode all of the axes are active and all of the programmed feedrates are in effect. Graphics are available on the active program graphics screen (chapter 8).

All of the above modes of execution begin program execution when the <CYCLE START> button is pressed. When the control is executing a part program block that block is shown with an * to the left of that block. When the control has completed the execution of a block that block is shown with an @ symbol to the left of the block. The @ symbol is usually only seen in single block mode or in cases where it is necessary to indicate what block automatic execution begins after.

Axis Inhibit, Dry Run, and Automatic operation can be interrupted using any of the operations listed below. Execution may be resumed at the interrupted location by pressing the <CYCLE START> button:

(1) Pressing <CYCLE STOP>

When the <CYCLE STOP> button is pressed, motion of the cutting tool decelerates and stops, and the control stops automatic operation. If the <CYCLE STOP> button is pressed during a dwell, the dwell is interrupted and any remaining time/revolutions for the dwell are stored for later execution.

(2) Execution of an M00 or M01 in a Part Program

M00 -- the control stops automatic operation after it executes the remaining commands in the M00 block.

M01 -- if the **OPTIONAL STOP** condition is set to **ON**, the control stops automatic operation after it executes the remaining commands in the M01 block. If the **OPTIONAL STOP** condition is set to **OFF**, the M01 is ignored and the control continues executing the part program as normal. The optional stop condition may be turned off or on using the Front Panel feature, or through a switch installed by the system installer.

(3) Entering a Sequence Stop Number

To interrupt execution at a specific block in the part program, use the sequence stop feature described in section 7.1.3. The control stops automatic operation after it completes the commands in the designated block.

(4) Feedhold Status

The system installer may have written PAL to allow the activation of a feedhold state through the use of a button or switch. When activated the control decelerates all moving axes to a feedrate of zero until the feedhold state is deactivated. For details on using feedhold refer to documentation provided by the system installer.

7.7.1
{QUICK CHECK}

Quick Check is a basic syntax checker for a part program. It checks that proper format and syntax has been followed during programming. No actual axis motion is produced in Quick Check mode however offsets and coordinate system shifts are performed. The Quick Check feature is also available with an optional graphics feature.

If the graphics feature is to be used refer to chapter 8 for Quick Check with graphics. To use the Quick Check feature as described below without graphics the graphics option must be disabled as discussed in chapter 8.

To use the Quick Check feature follow these steps.

1. Select a program to check as described in section 7.3 and return to softkey level 1.
2. Press the {QUICK CHECK} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

When the {QUICK CHECK} softkey is pressed, quick check is displayed in reverse video, and the graphics option is prompted.

3. Press the <CYCLE START> button.

When the <CYCLE START> button is pressed the program blocks are executed as fast as they can be processed by the control.

If a program block is found that contains an error, the program check stops and the control displays the message "ERROR FOUND".

To continue checking the remaining program blocks press the <CYCLE START> button again. If no more errors are found the control displays the message "COMPLETED WITH ERROR (S)" and the part program is automatically deactivated to allow editing.

Single Block mode is not available for QuickCheck unless you purchased the QuickCheck with graphics option. If your system has QuickCheck graphics you can check your part program one block at a time by activating Single Block mode and pressing the cycle start button. Single Block is ignored during QuickCheck if the graphics option is not installed on your system.

If the control finds no errors during Quick Check the program screen displays the message “COMPLETED WITH NO ERRORS”. The control then automatically resets the program to the first block. To disable Quick Check without the graphics options, simply press the {QUICK CHECK} button again. To disable Quick Check with the graphics options, press the {Quick Check} softkey followed by the {STOP CHECK} softkey.



CAUTION: Note that when a program is run during quick check mode, the control performs all coordinate system offset operations. This means that changes to the coordinate systems or coordinate offset tables are made (G10 blocks, changes to G92 and G52 offsets, and changes to the active work coordinate systems G54-G59.9). All of these changes are discarded at any termination of QuickCheck. The pre-QuickCheck values are restored when the {Stop Check} softkey is pressed. Note that program changes to the active offset or tool offset tables are not made in QuickCheck mode.

7.7.2 Axis Inhibit Mode

When AXIS INHIBIT is activated, the control can execute a part program without moving specified axes. The control simulates axis motion by updating the axis location and feedrate displays, using the commanded feedrates, acceleration, and deceleration. If graphics are being used the tool paths of any inhibited axis are still drawn on the graphics screen.

The program is executed in approximately the same amount of time as it would be in automatic mode, even though some or all axes may not move. The axis inhibit feature may be used in conjunction with Dry Run if desired.



WARNING: When testing a program using Axis Inhibit the control still recognizes and executes M, B, S, and T-codes. To ignore M, B, S, and T-codes, execute Axis inhibit in conjunction with miscellaneous function lock (see section 7.1.2).

AXIS INHIBIT can be activated to inhibit motion of any or all of the axes depending on the configuration determined by the system installer. This includes jogging moves. When axis motion has been inhibited for a single axis the remaining axes still execute as normal and the axis location display is updated as if axis motion was occurring on all axes.



WARNING: Axes not selected for axis inhibit move as they would if the program were executed in automatic mode.

The Axis Inhibit feature may be activated using a switch installed by the system installer (see documentation provided by the system installer) or by using the {FRONT PANEL} softkey (see chapter 2). The control must be in cycle stop or E-STOP to activate or deactivate the Axis Inhibit feature. Any attempt to activate or deactivate the feature during program execution or when in cycle suspend or feedhold states is ignored. Attempts to activate the Axis Inhibit feature during jogging are also ignored.

The spindle motion may also be inhibited by using a switch installed by the system installer. See the documentation provided by the system installer. The spindle cannot use the {FRONT PANEL} softkey to activate the Axis Inhibit feature.

Program execution with the Axis Inhibit feature is started by pressing the <CYCLE START> button provided a program has been selected for execution (see section 7.3).

Program execution with Axis Inhibit can be stopped at any time by using any of the methods described in section 7.7 or by pressing the <EMERGENCY STOP> button.

7.7.3 Dry Run Mode

The dry run function permits the checking of a part program to make sure that machine motions are correct. It is intended to be executed without the material or part mounted. The dry run function replaces all programmed feedrates with the maximum cutting feedrate. Jogging moves and moves that are programmed using rapid traverse (G00) are not effected by dry run.

The Axis Inhibit feature may be used in conjunction with Dry Run if desired.

If the external decel feature is used simultaneously with the Dry Run feature the feedrates that are assigned to External decel feature are used and the Dry run request is ignored.

The <FEEDRATE OVERRIDE> switch may be used to modify the cutting feedrate. The system installer determines in AMP if rapid feedrates are overrides by <RAPID FEEDRATE OVERRIDE> or the <FEEDRATE OVERRIDE> switch during Dry Run.

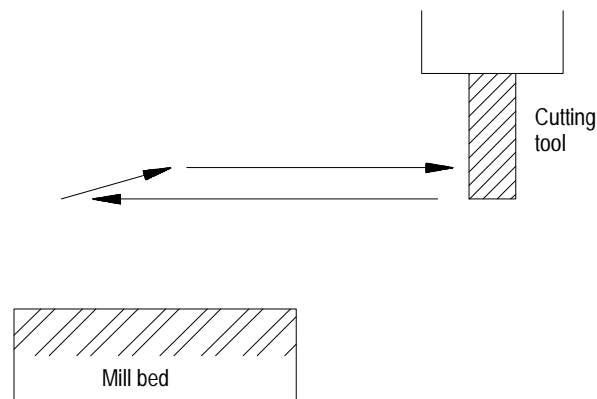


CAUTION: When testing a program using Dry Run the control still recognizes and executes M, B, S, and T-codes. To ignore M, B, S, and T-codes execute Dry Run in conjunction with miscellaneous function lock (section 7.1.2).



WARNING: It is possible for the system installer to write PAL to allow the operator to select DRY RUN at any time. This means that during normal automatic operation the operator can select maximum cutting feedrate and replace all feedrates programmed with an F word with the AMP assigned DRY RUN feedrate. This may result in damage to the machine, part, or injury to the operator.

Figure 7.3
Dry Run



The Dry Run feature may be activated using a switch installed by the system installer (see documentation provided by the system installer) or by using the {FRONT PANEL} softkey (chapter 2).

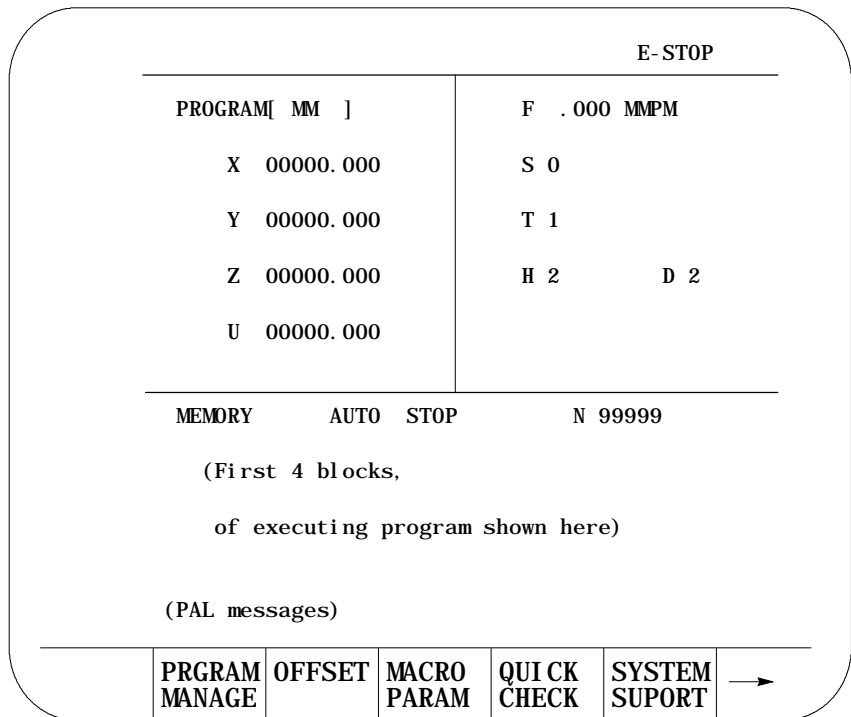
7.7.4
Part Production/Automatic
Mode

Automatic mode is the normal operating mode of the control. A program that is run in the automatic mode is executed with all of the axes active and all of the programmed feedrates active. Graphics is also available as discussed in chapter 8.

To select the automatic mode, place <MODE SELECT> (on the MTB panel) in the AUTO position. If not equipped with a mode select switch, use the {FRONT PANEL} softkey.

Automatic mode is the default mode whenever AUTO appears on the Main Menu screen and is always active unless one of the program checking modes has been selected.

Figure 7.4
Main Menu Screen in AUTO Mode



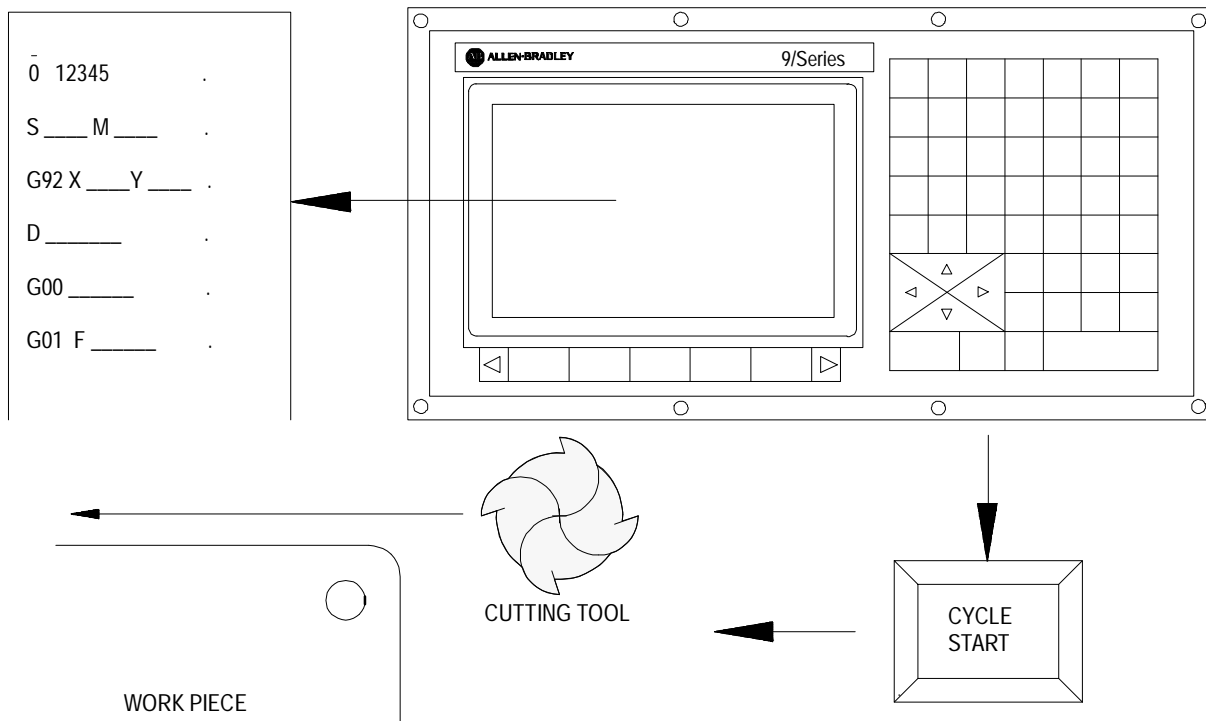
In automatic mode, the control manages machine operations according to the commands in a part program.

- CYCLE START -- begins part program execution
- CYCLE STOP -- stops part program execution



WARNING: Always test a program prior to automatic operation. Always verify that the workspace is clear and all safety features are intact before pressing CYCLE START.

Figure 7.5
Automatic Mode



Execution of a part program continues until the control encounters an M02 or M30. If the control does not encounter an M02 or M30 at the end of a program, the error message “MISSING M02/M30” appears.

Execution can be stopped at any time by using any of the methods described in section 7.7 or by pressing the <EMERGENCY STOP> button.

7.8 Interrupted Program Recover {RESTRT PROGRAM}

Use the program recover feature to resume a program that was executing and was interrupted by some means such as a control reset, E-STOP, or even power failure in some cases. This feature will scan the program as it searches for the interrupted block and from within the search area:

- send to PAL the last programmed modal G-codes from each modal group.
- send to PAL the last programmed modal M-codes from each modal group and set its associated PAL strobe (non-modal codes including user-defined M-codes are not sent to PAL).
- send to PAL the last programmed T code and set its associated PAL strobe
- send to PAL the last programmed auxiliary function code (B-word) and set its associated PAL strobe
- send to PAL the last programmed spindle commanded speed and set its associated PAL strobe
- resolve paramacro equations and assign paramacro variable values (note on dual process systems shared paramacro variables can be different than expected depending on the state of the part program in the other process, also equations that use PAL paramacro variables may also evaluate differently since no paramacro interaction with PAL occurs during a search operation).
- establishes any work coordinate system, including all offsets and rotations to the work coordinate system.

Important: Incremental moves that occur during a interrupted program recover operation, are always referenced from the last known absolute position in the part program. If no absolute position is specified in the searched part program blocks, the control will use the current axis position as the start point for incremental moves.

Unless Cutter Compensation is active, when a program recover is performed, the control automatically returns the program to the beginning of the block that was interrupted. In the case of power failure, the control will even reselect the program that was active prior to the interruption.



CAUTION: When a program recover is performed, the control automatically returns the program to the beginning of the block that was originally interrupted. The beginning of the block is probably not the point that axis motion was interrupted. For absolute linear moves this causes no problem if the tool is still somewhere along the path of the block that program execution was interrupted while cutting. In incremental or circular mode however, if the cutting tool is still located at the point that program execution was interrupted a restart may damage the part. If a program recover operation is performed in incremental mode it is important that the cutting tool be at the location that the interrupted program block began, not the location that the program was interrupted at.

This feature may be used to search into any subprogram or paramacro that may be contained in the main program also.



WARNING: It is the responsibility of the system installer's PAL application to make sure proper activation of all necessary machine functions has occurred before allowing a interrupted program to continue. You should verify that all machine functions are in the correct state before pressing <CYCLE START>.

Important: The interrupted program recover feature will not:

- send PAL nonmodal M-codes including user-defined groups 0 - 3, group 4, group 5, and group 6 M-codes.
- on dual process systems, halt execution for synchronization codes.
- read from or write paramacro variables to PAL
- on dual process systems, shared paramacro variables between processes may not be evaluated as desired depending on the status of the other process.
- send to PAL gear change requests based on spindle speed

To perform a program restore operation after automatic program execution has been interrupted follow these steps:

1. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

Important: DO NOT SELECT A PROGRAM AS AN ACTIVE PROGRAM. Do not disable the currently active program (if any). If a program is re-selected as active or disabled by the operator the program restore feature is canceled.

2. Press the {RESTRT PRGRAM} softkey. The control automatically re-selects the interrupted program if it was disabled by the control when power was lost.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTRT PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

3. To automatically search for the block in the current program that was interrupted press the {EXEC} softkey.

The control will locate an @ symbol to the left of the block immediately before the block that automatic execution was interrupted at.

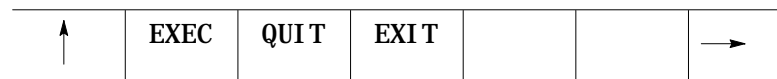
If this is not the block to begin execution from, press the {QUIT} softkey. The program restore feature will be aborted.



CAUTION: When you exit a program restart operation (search with memory), M- and S-codes are sent to PAL. If, during normal execution, that program activated a spindle, mid-program start may also start it.

4. Press the {EXIT} softkey if the block selected is the block to begin program execution from. If it not the desired block, it will be necessary to disable the program or perform a search with memory operation to locate the desired block manually.

(softkey level 3)



When the {CYCLE START} button is pressed the control resumes program execution from the block selected with the program re-start feature.

7.9 Jog Retract

The jog retract feature is intended to allow for inspection or change of the cutting tool during automatic program execution. It allows the cutting tool to be jogged from the workpiece in multiple steps, and then returned to the workpiece automatically by having the control retrace the jogging steps that were used.

The control remembers up to 15 jog retract moves. The actual number of moves retained may vary from 0 to 15 as determined by an AMP parameter set by the system installer. The control returns the tool along the jog retract path at a feedrate specified in AMP.

Important: If the same axis is used in succession during a jog retract operation, the control assumes that only one jog retract move has been executed on that axis.

Only simple single axes jog moves can be performed during the jog retract function. Multiple axis jogs, arbitrary angle jogs and jogging offset are not allowed.

Tool offsets can be changed at any time during jog retract (see chapter 3). The control will not make these offsets active until the execution of the first block after the tool has been returned from jog retract.



CAUTION: If the Jog Retract function is deactivated during its execution (performing a control reset, E-STOP, etc.), attempting to return the tool by pressing cycle start may cause the Jog Retract function to abort. The tool will return to the start point of jog retract along a linear path. This is most likely not the retracted path. To avoid possible part or tool damage it is recommended that the cutting tool be jogged to the point from which jog retract was started prior to pressing <CYCLE START>.

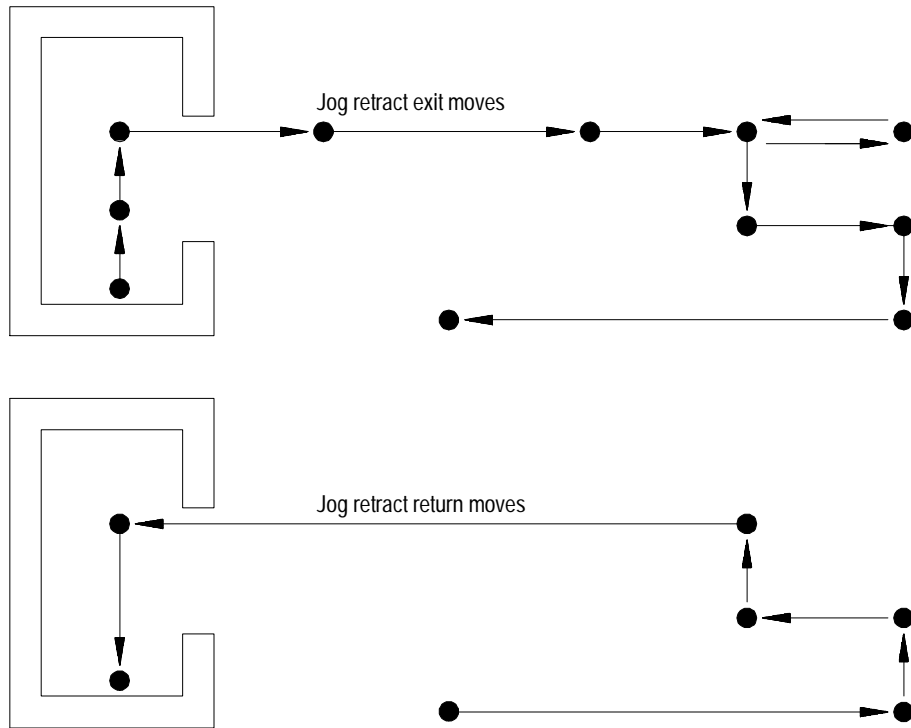
To perform a jog retract operation:

1. Press the <CYCLE STOP> or activate <SINGLE BLOCK> feature button to stop program execution.
2. Press the <JOG RETRACT> button. The light inside the button turns on to indicate that the function is active.
3. Move the cutting tool from the workpiece using either continuous jog, incremental jog, or HPG jog operations. (see chapter 4 for jogging information.)
4. Inspect and change the tool or tool offset as desired. Details on how to do this are in chapter 3.
5. After completing the desired inspection or tool change, press the <CYCLE START> button. Any tool offset changes you have made become active when the cycle start is requested. The tool returns to the location where jog retract began, following the same path used when you jogged the tool away from the work piece (+ or - any new tool offset values).

Cycle stop may be pressed during the tool's automatic return to the jog retract start position. When this is done the tool may be retracted from this point using jog moves and the control will add these moves to any remaining jog retract steps that have not yet been returned.

6. Once the cutting tool is fully returned from a jog retract operation the control will continue on in the part program unless in single block mode. If in single block mode the control will go to the cycle stop state when the return from jog retract is completed. Press <CYCLE START> again to resume program execution.

Figure 7.6
Jog Retract Operation



In Figure 7.6 notice that the control only recognized 6 jog moves upon returning instead of the actual 11 moves that were made to retract the tool. This is because the jog retract feature records consecutive jog moves on the same axis as one move.



CAUTION: If the number of jog retract moves performed exceeds the maximum allowed number set in AMP, the control will move the cutting tool directly from the final point of jog retract to the last remembered jog retract point along a straight line when <CYCLE START> is pressed. Then the tool will be returned in the normal jog retract fashion.

Figure 7.7
Jog Retract Moves that Exceed the Maximum Allowed in AMP

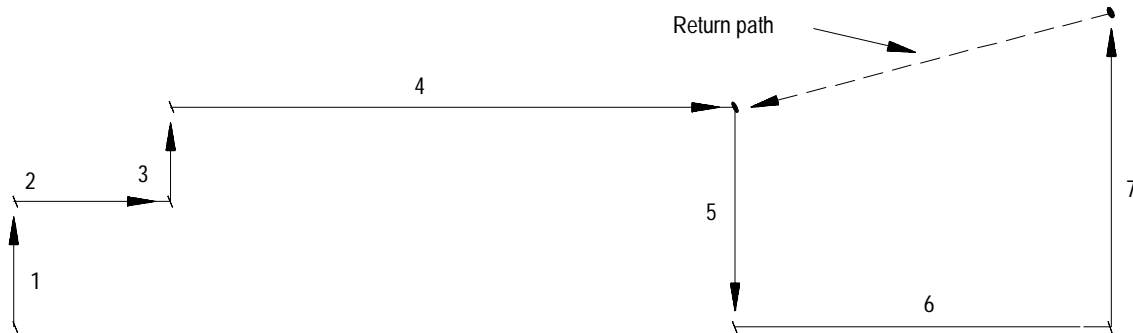


Figure 7.7 emphasizes the possible problems that may result from exceeding the maximum allowed jog retract moves. In this example the number of allowed moves set in AMP is four.

When the cycle start button is pressed at the end of the 7th jog move the control ignores moves 5, 6, and 7 and takes the shortest path to the endpoint of exit move 4. This is because the maximum number of jog retract moves set in AMP has been exceeded. After reaching the endpoint of move 4 the control continues the jog retract return operation as normal.

If the jogging moves of 5, 6, and 7 were intended to avoid a protrusion during the jog retract, a collision could result when returning the tool.

7.10 Block Retrace

The block retrace function allows the operator to retrace the motion created by up to 15 consecutive part program blocks. The actual number of retrace blocks allowed is set by the system installer in AMP, and may vary from 1 to 15.

Important: For maximum control efficiency when executing programs it is recommended that the maximum number of allowable block retraces is set as small as possible for the current machine application. This is because the number of allowable Block Retraces directly effects the controls block look ahead operation.

This function can only be enabled when the control is in cycle stop or cycle suspend state and is ignored if the control has already executed an M02 or M30 end of program.

To perform a block retrace operation:

1. Press the <CYCLE STOP> or activate the <SINGLE BLOCK> feature button to stop program execution.
2. Press the <BLOCK RETRACE> button.

After the <BLOCK RETRACE> button is pressed the control will retrace the block that was being executed when the cycle stop occurred or retrace the block just completed if the single block button was pressed, provided that the block is a legal block for retrace.

While the block retrace function is active, the light in the <BLOCK RETRACE> button will be on. The block that was shown as active when the block retrace was activated will still appear as the currently active block in the program display area during the entire use of the block retrace function.

Important: If the <CYCLE STOP> button was used to halt execution to begin a block retrace the control re-executes the portion of the block that has been executed. For example, if the block requests an axis move of 20 millimeters and the axis has moved 12 millimeters when the <CYCLE STOP> button is pressed, a block retrace will reverse the axis direction 12 millimeters.

All retraced blocks are executed at the feedrate programmed for that block though this may be modified by the use of the <FEEDRATE OVERRIDE> switch.

Press the <CYCLE START> button at any time during a block retrace to return the cutting tool to normal forward execution. Program execution will return to the normal forward direction from the currently retraced block. The control executes the retraced blocks in normal order until the tool is positioned at the start point of block retrace. From this point it continues program execution in a normal fashion unless <SINGLE BLOCK> is active. If <SINGLE BLOCK> is active the control will halt execution when the return from block retract is complete.

While block retrace is active the control disables all jog features with the exception of <JOG RETRACT> . MDI is not available to insert blocks during a block retrace operation.

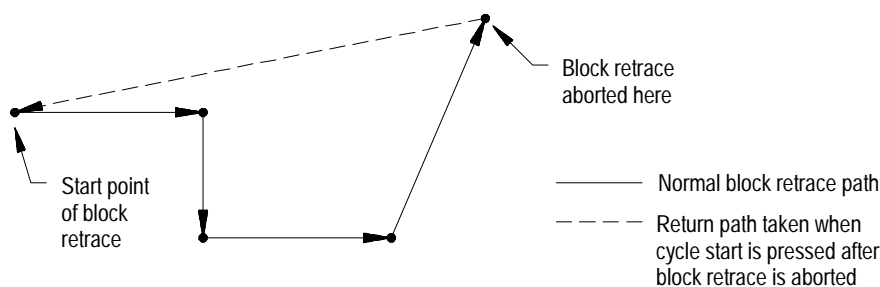
The block retrace function is unable to retrace any of the following blocks and an attempt to do so will result in an error message.

- Threading
- Tapping
- Boring
- Inch/Metric changes (unit conversion)
- A block that commands a tool change operation.
- A block that commands a change in the coordinate system.
- Any block that is followed by a Manual Jog Move except a Jog Retract.
- The number of blocks retraced is already equal to the maximum number of re-traceable blocks as determined in AMP.
- Certain Paramacro Parameter Assignments
- Interrupt Macros



CAUTION: If the block retrace function is deactivated during its execution (performing a control reset, E-STOP, etc...), attempting to return the tool by pressing cycle start may cause an undesired return path. The tool will return to the start point of block retrace along a linear path. This is most likely not the retracted path. To avoid possible part or tool damage it is recommended that the cutting tool be jogged to the point from which block retrace was started prior to pressing cycle start.

Figure 7.8
Pressing Cycle Start When Retract Path is Lost



END OF CHAPTER

Display and Graphics

8.0 Chapter Overview

The first part of this chapter gives a description of the different data displays available on the control. The second part gives a description of the control's graphics capabilities.

8.1 Selection of Axis Position Data Display

Pressing the [DISP SELECT] key displays the softkeys for selecting the axis position data screens.

The control provides 8 different axes position data screens as described in Table 8.A. Four of these screens may be displayed in normal (9 axis maximum), large (4 axis triple size or 6 axis double size) or small (all axis in process) characters if desired. Normal size is the default.

Table 8.A
Display Select Softkeys

Display	Description
{PROGRAM}	Axis position in the current work coordinate system is displayed. Each time this softkey is pressed the display toggles between normal, large, and small ¹ .
{ABS}	Axis position in the machine coordinate system is displayed. Each time this softkey is pressed the display toggles between normal, large, and small ¹ .
{TARGET}	Coordinate values, in the current work coordinate system, of the end point of commanded axis motion is displayed. Each time this softkey is pressed the display toggles between normal, large, and small ¹ .
{DTG} Distance to go	Distance from the current position to the end point of the commanded axis move displayed. Each time this softkey is pressed the display toggles between normal, large, and small ¹ .
{AXIS SELECT}	This softkey is used to select which axes are going to be displayed on normal (when more than 9 axes are available) and large displays. Small displays always show all system axes.
{M CODE STATUS}	M codes that are currently active are displayed.
{PROGRAM DTG}	This screen provides a multiple display of information from the program display screen and the distance to go screen.
{ALL}	This screen provides a multiple display of position information program, target, absolute, and distance to go screens. The all display is only available on systems with 6 or less axes. On systems with more than 6 axes, other combination screens are available which display a subset of the data available on the ALL display.
{G CODE STATUS}	G codes that are currently active are displayed.
{SPLIT SCREEN ON/OFF}	If your system is AMPed for more than one process, you will have this additional softkey. It allows you to view both processes at the same time. The active process appears in reverse video. This selection remains for all future power up cycles or until you change it.

¹ Small displays are only available when the system is configured to have more than 9 real axes or, for dual process systems, more than 8 axes in a process. Virtual axes count as axes in your total. Adaptive depth probes configured as axes are not counted in this total.

The screens described above may also show in addition to axis position:

- The current unit system being used (millimeters or inches)
- E-STOP
- The current feedrate
- The current spindle speed of the controlling spindle
- The current tool and tool offset numbers
- The active program name (if any)
- The active subprogram name (if any)
- The current amount of usable memory remaining
- The current operating mode (MDI, manual or automatic)
- The current operating status (cycle stop, suspend, start, feedhold)
- The current block executing (sequence number)
- Up to four blocks of the current program selected for program execution
- Subprogram paramacro 01 canned cycle repeat count executing

To select an axis position data display :

1. Press the [DISP SELECT] key, to display the softkeys for selecting axis position data screens. Press the [DISP SELECT] key at any time from any softkey level. Pressing the page {• } softkey displays additional selections.

Important: The [DISP SELECT] key is ignored if you are viewing the PAL search monitor utility. You must select the screen to display before you access PAL search monitor. Additionally you can not activate PAL search monitor while viewing the Display Select options.

2. Press the softkey corresponding to the display wanted. The softkeys will toggle between large and regular display mode each time the corresponding softkey is pressed, provided that screen is available as a large display.

The “large” and “small” display is available only for the axis position screens (Program, Absolute, Target, and Distance to Go). Small displays are only available when more than 9 axes are AMPed in a system or, for dual process, more than 8 axes in a process.

For example, immediately after power up and accessing the [DISP SELECT] feature, pressing the {DTG} softkey displays the distance to go in normal size. Pressing it again changes the display to show the distance to go in large character size.

The control can display any 4 axes in triple-height characters and any 6 axes in double-height characters. If you ask the control to display more than 6 axes on a large display or 9 axis on a normal display, the control displays the error message, “TOO MANY AXES SELECTED FOR DISPLAY.” Small displays always show all axes in the system in the active process.

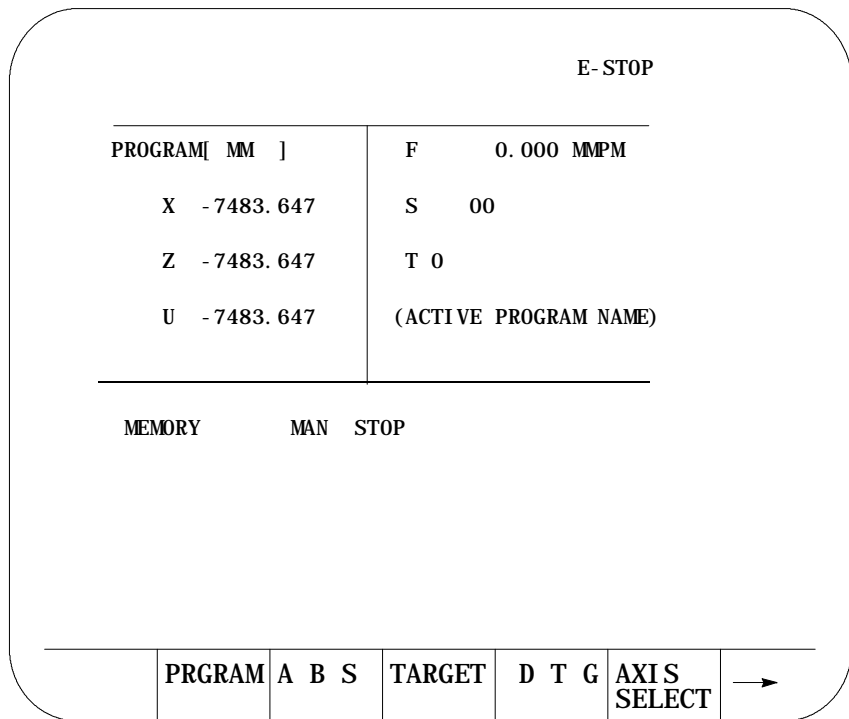
- To return to softkey level 1, press the [DISP SELECT] key again. The most recently selected data position screen will remain in effect for softkey level 1 until either power is turned off or a different position display screen is selected. The default screen selected at power up is the regular size program display.

The following figures show the axis position data display that will result when the corresponding softkey is pressed.

(1) {PRGRAM}

Axis position in the current work coordinate system displayed in normal size characters.

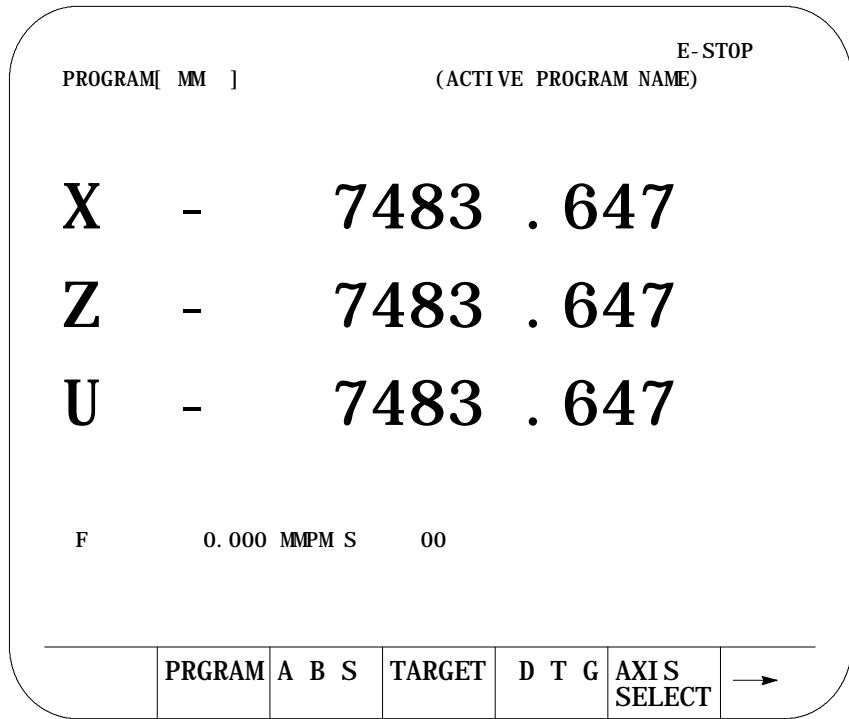
Figure 8.1
Result After Pressing {PRGRAM} Softkey



(2) {PRGRAM} (Large Display)

Axis position in the current work coordinate system displayed in large characters.

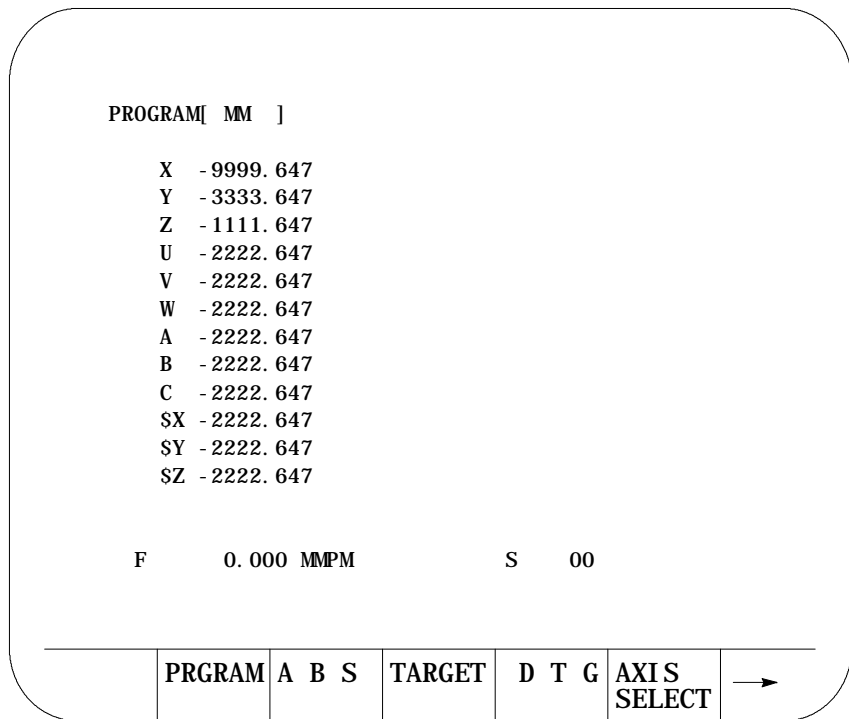
Figure 8.2
Results After Pressing {PRGRAM} (Large Display) Softkey



{PRGRAM} (Small Display)

Axis position in the current work coordinate system displayed for all system axes in the active process (only available when more than 9 axes are AMPed in the system, or more than 8 axes in the process for dual process systems).

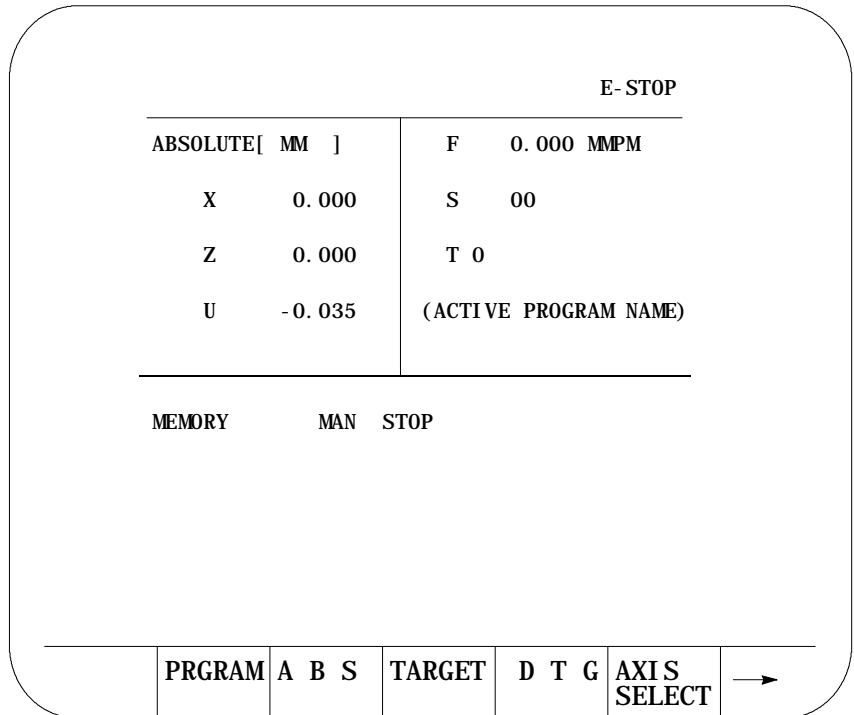
Figure 8.3
Results After Pressing {PRGRAM} (Small Display) Softkey



(3) {ABS}

The axis position data in the machine coordinate system.

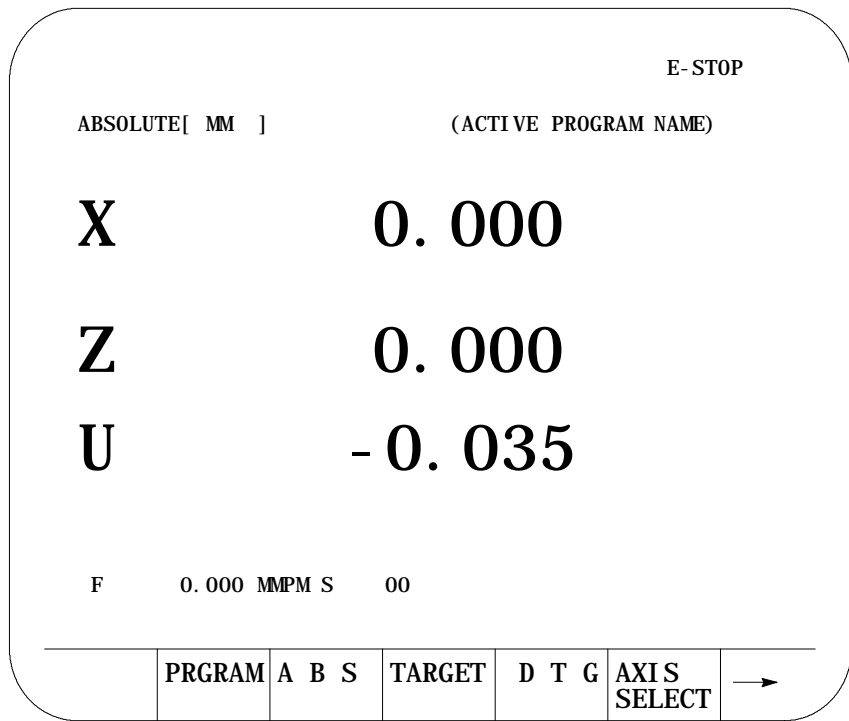
Figure 8.4
Results After Pressing {ABS} Softkey



(4) {ABS} (Large Display)

Axis position in the machine coordinate system displayed in large characters.

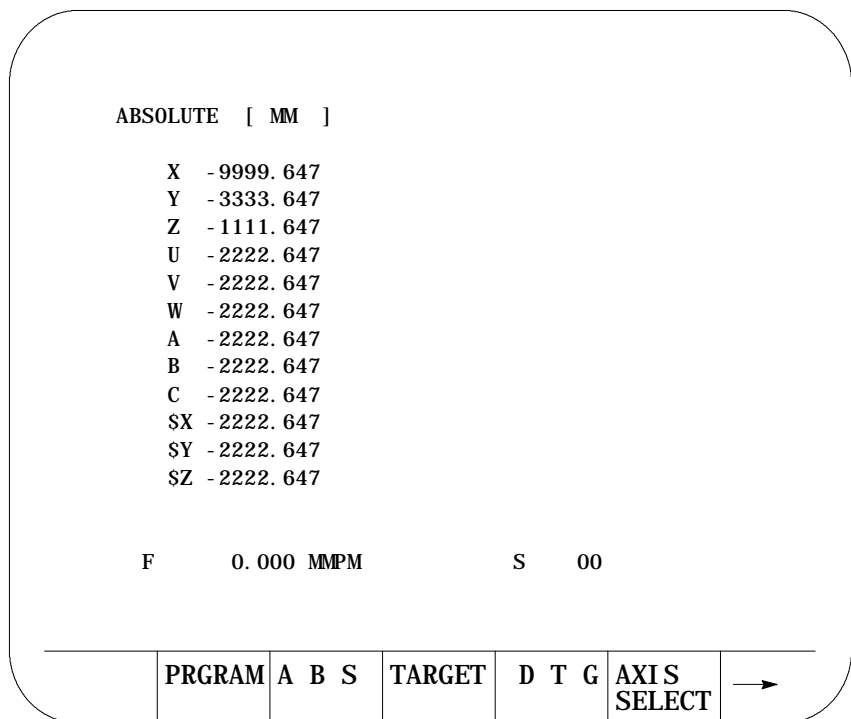
Figure 8.5
Results After Pressing {ABS} (Large Display) Softkey



{ABS} (Small Display)

The axis position data in the machine coordinate system displayed for all system axes in the active process (only available when more than 9 axes are AMPed in the system, or more than 8 axes in the process for dual process systems).

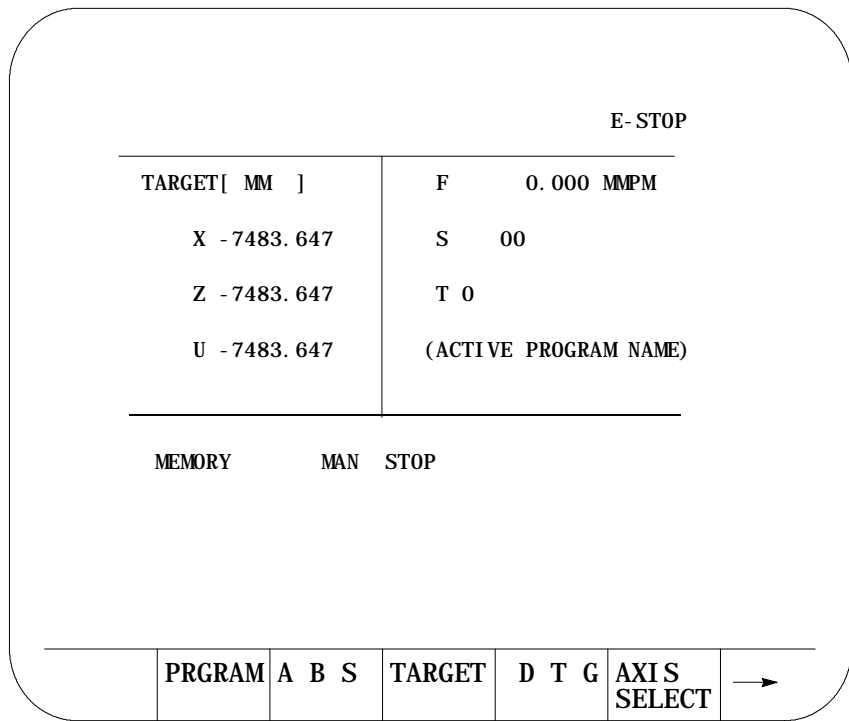
Figure 8.6
Results After Pressing {ABS} (Small Display) Softkey



(5) {TARGET}

The coordinate values of the end point of the currently executing axis move is displayed at a position in the current work coordinate system.

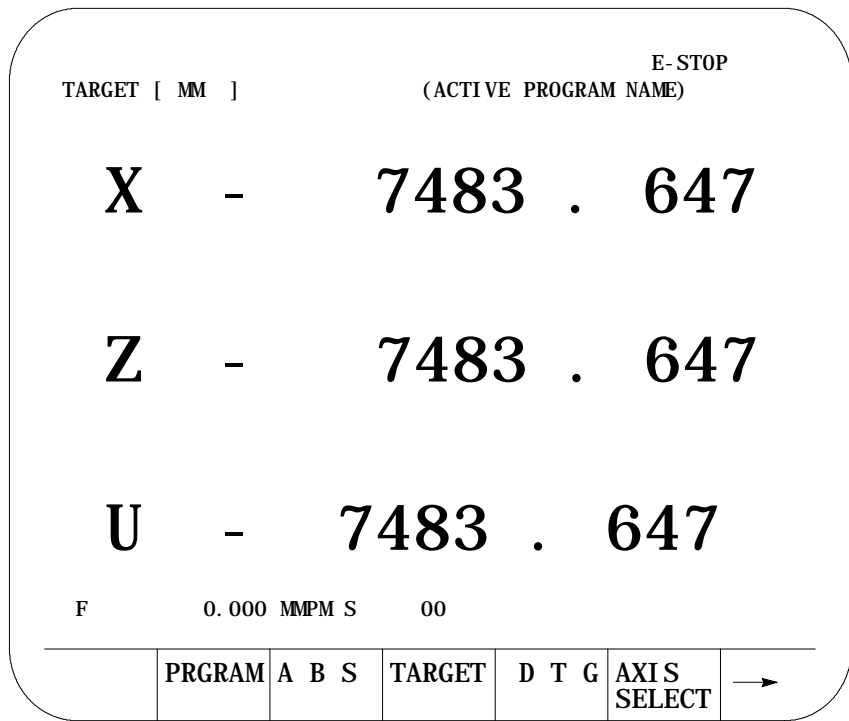
Figure 8.7
Results After Pressing {TARGET} Softkey



(6) {TARGET} (Large Display)

The coordinate values in the current work coordinate system, of the end point of commanded axis moves in normal size characters.

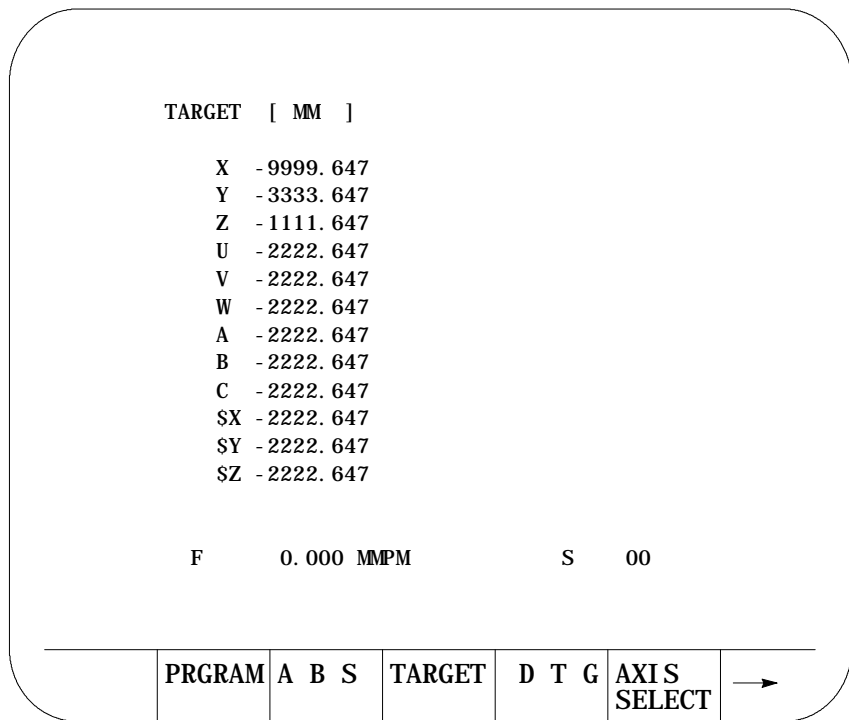
Figure 8.8
Results after Pressing {TARGET} Softkey



{TARGET} (Small Display)

The coordinate values of the end point of the currently executing axis move is displayed at a position in the current work coordinate system for all system axes in the active process (only available when more than 9 axes are AMPed in the system, or more than 8 axes in the process for dual process systems).

Figure 8.9
Results After Pressing {TARGET} (Small Display) Softkey



(7) {DTG}

The distance from the current position to the command end point, of the commanded axis in normal size characters.

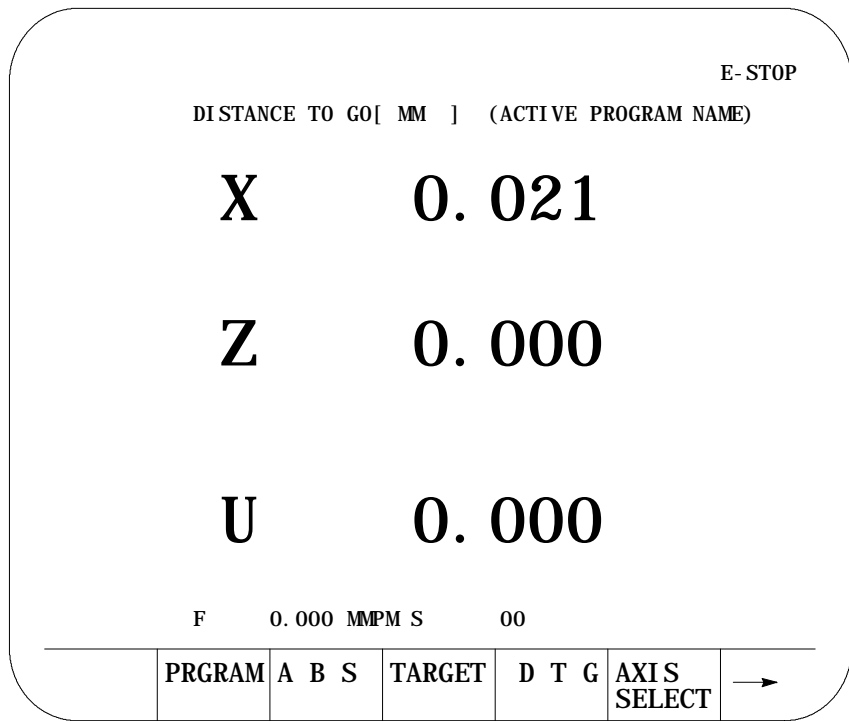
Figure 8.10
Results After Pressing {DTG} Softkey

DISTANCE TO GO[MM]		E-STOP			
X	0.021	F	0.000 MPM		
Z	0.000	S	00		
U	0.000	T	0		
		(ACTIVE PROGRAM NAME)			
MEMORY		MAN STOP			
PROGRAM	A B S	TARGET	D T G	AXIS SELECT	→

(8) {DTG} (Large Display)

The distance from current position to the command end point of the commanded axis move in large characters.

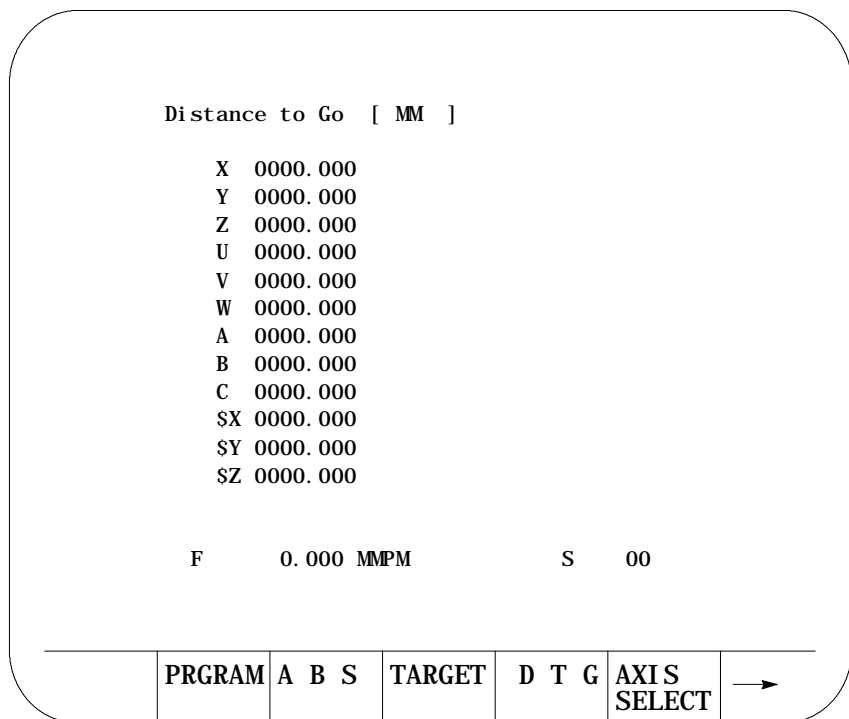
Figure 8.11
Results After Pressing {DTG} (Large Display) Softkey



{DTG} (Small Display)

The distance from the current position to the command end point, of the commanded axis in normal size characters is displayed for all system axes in the active process (only available when more than 9 axes are AMPed in the system, or more than 8 axes in the process for dual process systems).

Figure 8.12
Results After Pressing {DTG} (Small Display) Softkey



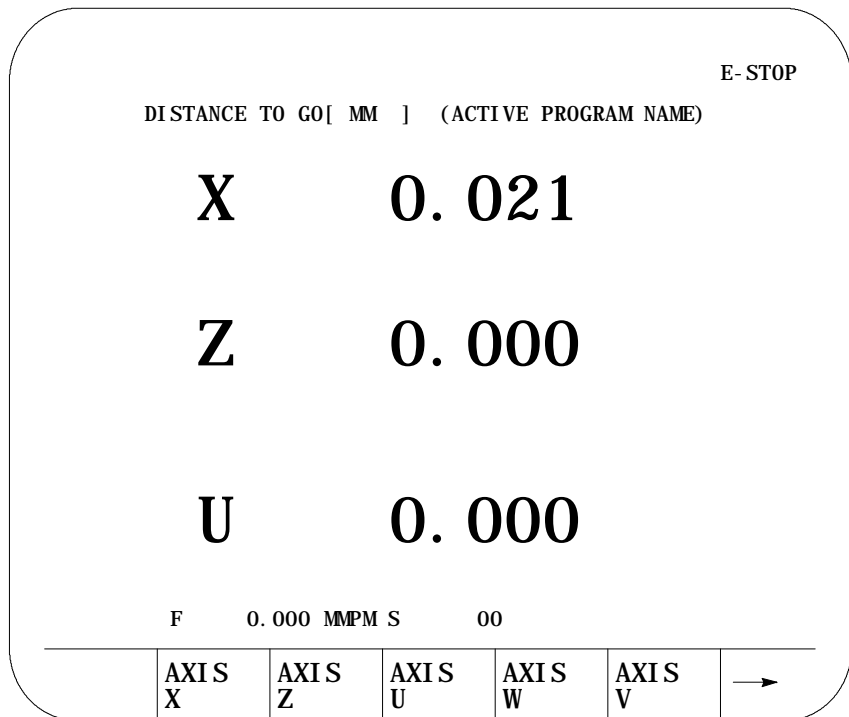
(9) {AXIS SELECT}

Important: {AXIS SELECT} is available only during a large character display or when more than 9 axes are displayed on a normal size display.

When you press {AXIS SELECT}, the control displays the axis names in the softkey area. Press a specific axis letter softkey to toggle the position display of that axis on and off.

If a normal size display is being viewed and the system has more than 9 axes, the axis select features can also select the axes for these normal size displays. This feature has no affect on small displays. Small displays always show all axes (only available on systems with more than 9 axes or on dual process systems with more than 8 axes in a process).

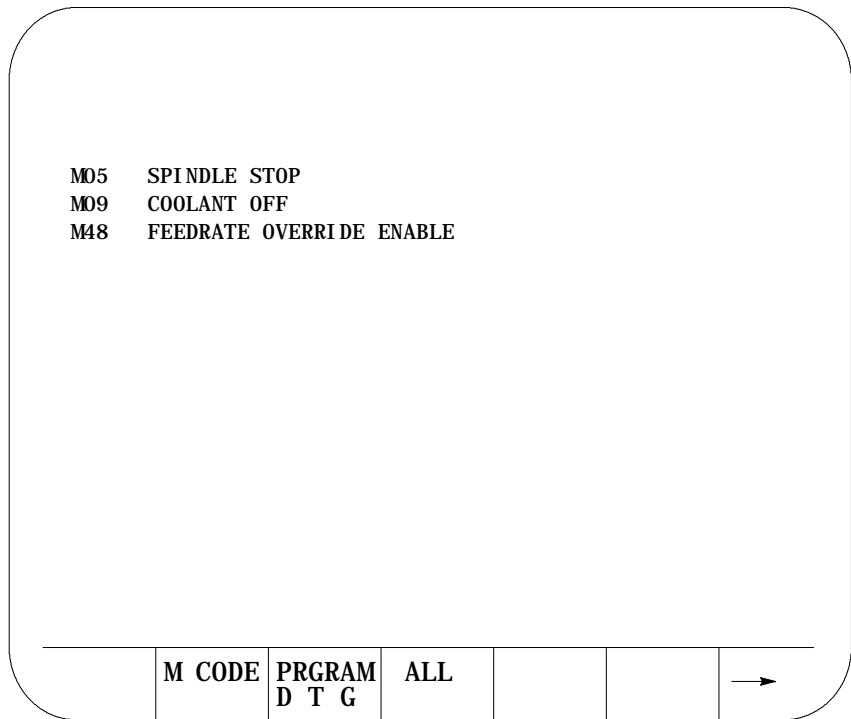
Figure 8.13
Results After Pressing {AXIS SELECT} Softkey



(10) {M CODE STATUS}

The currently active M codes are displayed. This screen indicates only the last **programmed** M code in the modal group. It is the PAL programmer's responsibility to make sure proper machine action takes place when the M code is programmed.

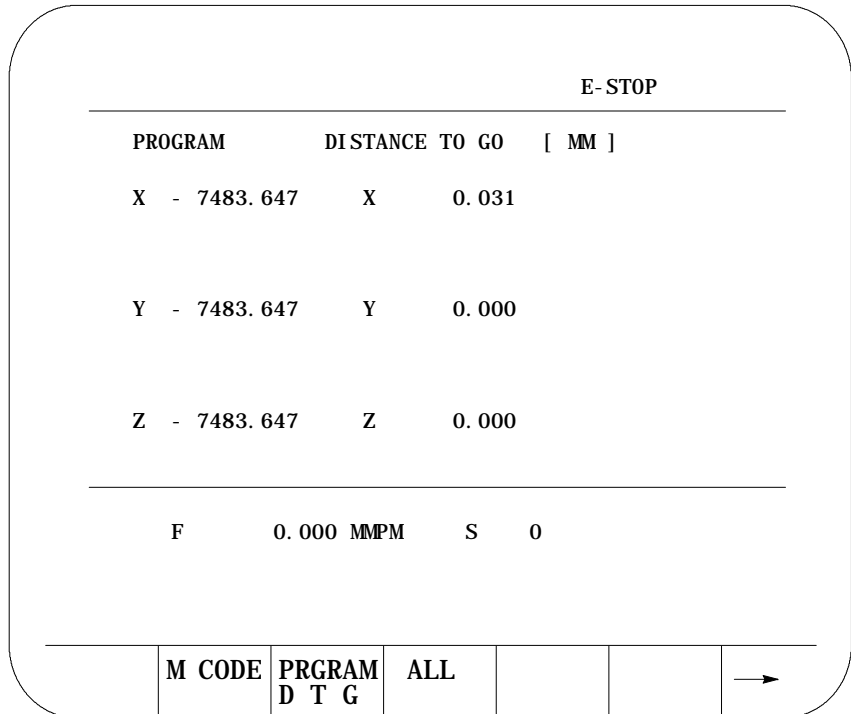
Figure 8.14
Result After Pressing {M CODE} Softkey



(11) {PRGRAM DTG}

This screen provides a multiple display of position information from the program screen and the distance to go screen.

Figure 8.15
Program, Distance to Go Screen



{PRGRAM DTG} (Small Display)

This screen provides a multiple display of position information from the program screen and the distance to go screen. It displays all system axes in the active process (only available when more than 9 axis are AMPed in the system, or more than 8 axis in the process for dual process systems).

Figure 8.16
Program, Distance to Go Screen (Small Display)

PROGRAM		Distance to Go [MM]	
X	-9999.647	X	0000.000
Y	-3333.647	Y	0000.000
Z	-1111.647	Z	0000.000
U	-2222.647	U	0000.000
V	-2222.647	V	0000.000
W	-2222.647	W	0000.000
A	-2222.647	A	0000.000
B	-2222.647	B	0000.000
C	-2222.647	C	0000.000
SX	-2222.647	SX	0000.000
SY	-2222.647	SY	0000.000
SZ	-2222.647	SZ	0000.000
F	0.000 MPM	S	00

M CODE	PRGRAM D T G	ALL			→
--------	-----------------	-----	--	--	---

(12) {ALL}

This screen provides a multiple display of position information from the program, distance to go, absolute, and target screen. The all display is only available on systems with 6 or less axes. On systems with more than 6 axes, other combination screens are available which display a subset of the data available on the ALL display.

Figure 8.17
Result After Pressing {ALL} Softkey

E- STOP					
PROGRAM		DISTANCE TO GO [MM]			
X	- 7483.647	X	0.000		
Y	- 7483.647	Y	0.000		
Z	- 7483.647	Z	0.000		
ABSOLUTE		TARGET			
X	0.000	X	- 7483.647		
Y	0.000	Y	- 7483.647		
Z	0.000	Z	- 7483.647		
U	0.000	U	- 7483.647		
F	0.000 MPM	S	0		
M CODE		PRGRAM	ALL	G CODE	
		D T G		STATUS →	

The {ALL} selection is not available for more than 6 axes.

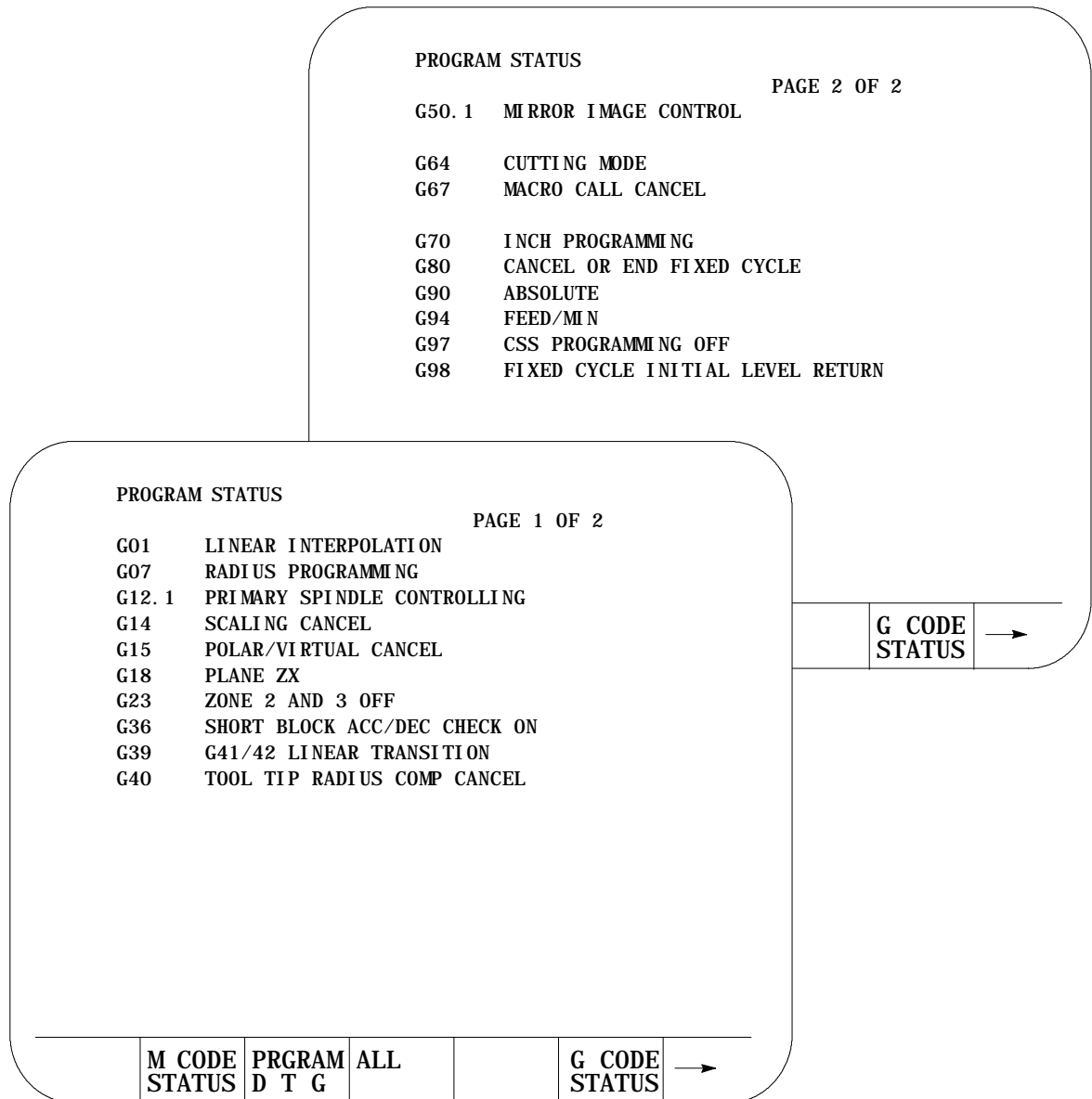
Important: If you have more than 6 axes, then you will not have an {ALL} softkey. A {PRG TAR} and a {PRG ABS} softkey appear.

↑	M CODE STATUS	PRGRAM D T G	PRG TAR	PRG ABS	G CODE STATUS
---	------------------	-----------------	------------	------------	------------------

(13) {G CODE STATUS}

The currently active G-codes are displayed.

Figure 8.18
Results After Pressing {G CODE} Softkey



(14) {SPLIT ON/OFF}

The split screen softkey is only available if your system installer has purchased the dual-process option.

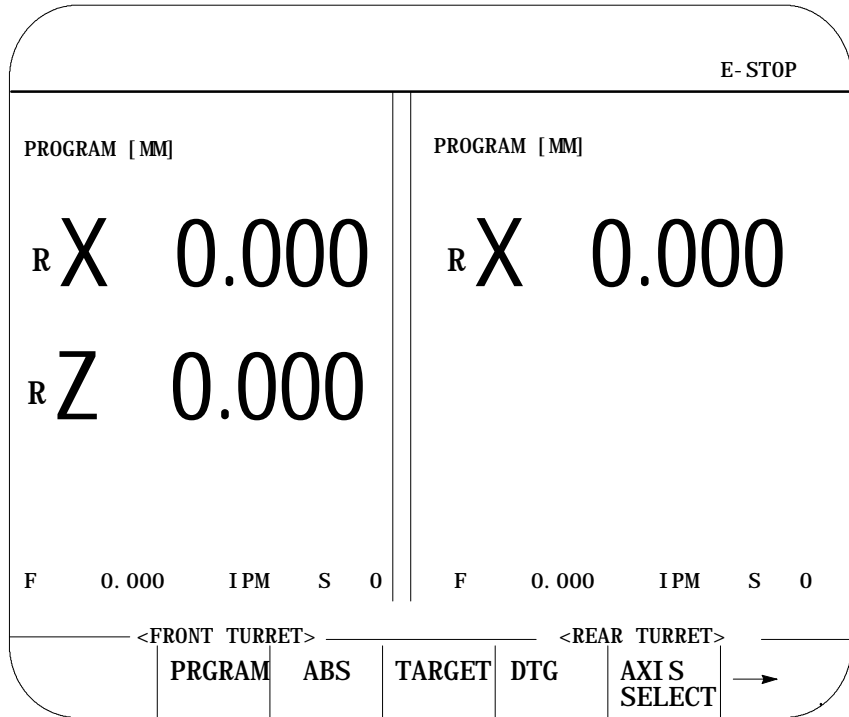
When you press the {SPLIT ON/OFF} softkey, you can view information for both processes. The screen displays two 40-column screens on one 80-column screen. Process 1 is displayed on the left, and process 2 is displayed on the right. The active process appears in reverse video.

You can display axis display, M-code, G-code, and large axis screens as a split screens.

Important: You can only select one process on the split screen at a time. The selected process appears in reverse video. If only one process is AMPed, the split screen display is disabled.

E- STOP			
PROGRAM [MM]	F	PROGRAM [MM]	F
X 0.000	S	X 0.000	S
Z 0.000	T	Z 0.000	T
U 0.000		U 0.000	
MEMORY AUTO STOP N1999		MEMORY AUTO STOP N9999	
N001G90: N002G70: N003M101:		N001G90: N002G70: N003M101:	
<FRONT TURRET>		<REAR TURRET>	
PRGAM MANAGE	OFFSET	MACRO PARAM	PROGRAM CHECK
			SYSTEM SUPORT →

A large screen display makes it easier for you to see the axes.



8.2 PAL Display Page

If desired the system installer has the option of configuring custom screens that will show up on the CRT. These screens may be activated through an input to PAL (such as a switch or push button) or by the use of softkeys also defined by the system installer. These screens may have their own softkey tree configuration or parameters to assign. Refer to the system installers documentation for details on the operation of any PAL display pages that may be defined on your system.

PAL display pages may be accessed in either automatic, manual, or MDI modes. If the control is in MDI mode when a PAL display is active the MDI prompt will be available in the normal position on the screen. This will however prevent the operator from typing in any input for that screen using the keyboard. Any data typed in on the keyboard will be used for an MDI operation.

When changing the value of some parameter on the PAL display page, part program execution is not typically interrupted. If some data that is used in a currently executing part program is changed the control will handle that data in the following manner:

- If the parameter altered is used in the currently executing program block, that value will not be activated until the following block (unless a cutter compensation value is being altered).
- If the parameter is altered in a block that is within the controls look ahead range (refer to chapter 22 for details on block look ahead) then the look ahead blocks are re-setup and the new parameter value is incorporated in them (unless a cutter compensation value is being altered).
- If a cutter compensation value is being altered the control will not re-setup any blocks in the block look ahead. If the modified value is currently active the block look ahead buffer is re-setup with the new offset value. If the modified value is not currently the active value the control will display an error message.

8.3 Changing Languages

The 9/Series control is equipped to display all screens, softkeys, and messages in multiple languages. Functionality of the 9/240 language function is somewhat different than other 9/Series controls.

Press the {SWITCH LANG} softkey to access these languages.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

Each time you press the {SWITCH LANG} softkey, the language displayed on the screen changes. The system installer can password protect this softkey.

9/230, 9/260, 9/290 and 9/440 CNCs

These control types are capable of displaying two languages. Your system installer selects the languages and the order in which they are displayed. Refer to your system installers documentation for details on which languages your system supports. The default languages are English and German.

9/240 CNCs

The 9/240 control is equipped to display four languages. The languages available and the order they are displayed are fixed in this order:

- English
- Italian
- Japanese
- German

8.4 Graphics

QuickCheck and active program graphics function similarly. They both plot tool paths. The following section describes how to use both types of graphics and distinguishes how they differ.

Use the QuickCheck with graphics feature to test a program for correct syntax and to visually check if the tool path plotted by the part program is correct. No actual axis motion takes place during QuickCheck. The QuickCheck with graphics feature is identical to the QuickCheck feature described in chapter 8 with the exception that there is an added graphics capability. Using the graphics feature is optional with QuickCheck; programs can be checked as described in the QuickCheck section without using graphics. QuickCheck also lets you single-block through the part program.

Use the active program graphics feature to plot the actual tool path of a program on the screen that is executing in either the axis inhibit, dry run, or part production mode. The graphics feature is optional and you do not need to use it when you run a program.

8.4.1 Selecting the Program for Graphics

If you have not already selected a program to execute, select one now, following these steps:

1. Press the {PROGRAM CHECK} softkey.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	PROGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWITC H LANG		→

2. Select a program. Press {SELECT PRGRAM}.

(softkey level 2)

↑	SELECT PRGRAM	QUICK CHECK	STOP CHECK	T PATH GRAPH	T PATH DISABL
---	------------------	----------------	---------------	-----------------	------------------

3. Use the up and down cursors to select a program.
4. Press {ACTIVE PRGRAM} to return to level 2 and activate the program.

8.4.2 Running Graphics

Follow these steps to run graphics:

1. Press the {PRGRAM CHECK} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
--	------------------	--------	----------------	-----------------	------------------	---

	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWITCH LANG		→
--	----------------	------------------	---------------	----------------	--	---

2. Select a program. Press {SELECT PRGRAM}.

(softkey level 2)

↑	SELECT PRGRAM	QUICK CHECK	STOP CHECK	T PATH GRAPH	T PATH DISABL
---	------------------	----------------	---------------	-----------------	------------------

3. Use the up and down cursors to select a program.
4. Select graphics. Follow the appropriate steps:

Active Graphics

(softkey level 2)

↑	SELECT PRGRAM	QUICK CHECK	STOP CHECK	T PATH GRAPH	T PATH DISABL
---	------------------	----------------	---------------	-----------------	------------------

QuickCheck Graphics

(softkey level 2)

↑	SELECT PRGRAM	QUICK CHECK	STOP CHECK	T PATH GRAPH	T PATH DISABL
---	------------------	----------------	---------------	-----------------	------------------

(softkey level 3)

↑	GRAPH	SYNTAX ONLY			
---	-------	----------------	--	--	--

The graphics you select remains active until you disable graphics.

5. Press the <CYCLE START> button to plot the part program.

The control for both QuickCheck and active graphics continues to plot tool paths, even if the graphics screen is not displayed. Actual display of the tool paths is only possible on the graphics screen. When the graphics screen is displayed again, any new tool motions appear on the screen.

While on the graphics screen only the currently executing block is displayed. The currently executing block is displayed on line 22 of the CRT, and it is limited to 80 characters. Any characters following the eightieth character in a block are truncated for display purposes only.

Important: If the graphic screen does not display the tool path you want, you may need to alter the graphic parameters. To change your graphic parameters see page 8-27.

Scale

When you first activate the graphics screen, the scale for the axes is based on the software overtravel limits (see chapter 13). The selected axis with the largest software overtravel limit is used as the scale axis for the graphics screen. The range for the other axis is then scaled up to create a proportional graph. This helps keep the shapes on the graph proportional (circles look like circles not ellipses etc.).

The zero of the scale for each axis is determined by the currently active work coordinate system (G54-G59.3). Any time that a coordinate system is changed, the scale automatically changes to match the current position with the new position in the new coordinate system. Any offsets to the work coordinate system also change the scale for the axes. When the scale changes, the plot of the tool path does not change; only the scale changes to match the plot to the new coordinate system.

If you want to adjust the scale, use zoom window. Refer to page 8-33.

Important: You can adjust the scale factor automatically to match the size of the part program currently being executed. To do this, use the auto size feature described in chapter 1.

8.4.3 Disabling Graphics

In some cases, you may want to operate without graphics. For example, you cannot edit a part program using QuickView while in graphics, or you may want to speed up processing by disabling graphics.

To disable graphics, press the appropriate softkey:

Active Graphics						QuickCheck Graphics					
(softkey level 2)						(softkey level 2)					
↑	SELECT PROGRAM	QUICK CHECK	STOP CHECK	T PATH GRAPH	T PATH DISABL	↑	SELECT PROGRAM	QUICK CHECK	STOP CHECK	T PATH GRAPH	T PATH DISABL

When you attempt to use a feature that cannot be run with active graphics. You will be prompted to disable graphics.

A {T PATH DISABLE} softkey also appears on level three. If you press the {PROGRAM MANAGE} softkey on level one, followed by {ACTIVE PROGRAM} softkey on level 2, you will have the option to disable graphics. Both {T PATH DISABL} keys deactivate graphics.

8.4.4 Changing Parameters

QuickCheck graphics and active graphics share some of the same graphic parameters. This section describes the graphic parameters they share and identifies their separate parameters.

When the parameter screen is displayed, it always shows the last set of parameter values that were saved. If you disable graphics, the control still saves the last set of parameters. However, parameters are not saved after you turn the power off.

QuickCheck and active graphics share the same parameter setup. If you change a parameter in one graphic feature, it is saved to the other.

Important: Any time you change one of the parameters on the graphics parameter screen (with the exception of auto erase), it causes any tool paths that may have already been drawn on the actual active graphics screen to be cleared. Drawing resumes from the point that the program is currently executing.

Important: If you use a color system and you want a different tool color representation, see chapter 21 for details on changing tool color.

You may want to change the parameters to alter your graphics. If you want to view a different graphics screen, you must change the default values for the parameters. These are the default parameter values for QuickCheck:

PROCESS SPEED: [FAST]
 RAPID TRAVERSE: [ON]
 AUTO SIZE [OFF]
 GRID LINES [OFF]
 OVERTRAVEL/ZONE LINES [OFF]
 MAIN PROGRAM SEQUENCE STARTING #[0]
 MAIN PROGRAM SEQUENCE STOPPING #[0]

These are the default parameter values for active graphics:

RAPID TRAVERSE [ON]
 AUTO ERASE [ON]
 AUTO SIZE [OFF]
 GRID LINES [OFF]
 OVERTRAVEL/ZONE LINES [OFF]

Follow these steps to change your parameters. We assume you have already completed the procedure to run graphics. See page 8-25 for details on running a graphics program.

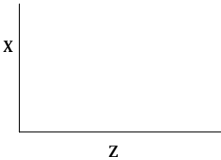
- To load parameters, press {GRAPH SETUP}.

(softkey level 4)

↑	CLEAR GRAPHS	MACHIN INFO	ZOOM WINDOW	ZOOM BACK	GRAPH SETUP
---	-----------------	----------------	----------------	--------------	----------------

The control displays these graphics parameter screens:

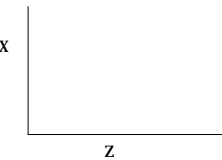
CHECK RUN GRAPHICS PARAMETERS
 USE <ARROW KEYS> TO SELECT PARAMETERS
 ACTIVE PROGRAM: G52



PROCESS SPEED FAST
 RAPID TRAVERSE ON
 AUTO SIZE OFF
 GRID LINES OFF
 OVERTRAVEL/ZONE LINES OFF
 MAIN PROGRAM SEQUENCE STARTING #: 0
 MAIN PROGRAM SEQUENCE STOPPING #: 0

↑	DEFALT PARAM	SAVE PARAM				
---	-----------------	---------------	--	--	--	--

ACTIVE PROGRAM GRAPHICS PARAMETER
 USE <ARROW KEYS> TO SELECT PARAMETERS
 ACTIVE PROGRAM: G21



RAPID TRAVERSE ON
 AUTO ERASE OFF
 AUTO SIZE OFF
 GRID LINES OFF
 OVERTRAVEL/ZONE LINES OFF

↑	DEFALT PARAM	SAVE PARAM				
---	-----------------	---------------	--	--	--	--

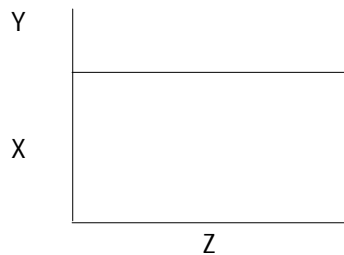
2. **Set Select Graph.** Use the up and down cursor keys to select the axes. Then set them by pressing the left or right cursor keys. The data for the selected axes change each time you press the left or right cursor key.

A pictorial representation of the selected graph, which is determined by the selected axes, is displayed on the screen.

You have three fields that you can adjust. The axes are shown as horizontal and vertical axes. If you choose a third axis for display, it appears as a horizontal axis above the first one. Only linear axes can be selected for display on the graphics screen.

For example:

If the X, Y, and Z axes are selected, this representation appears:



Choose a set of axes that best displays the current program's cutting path.

3. **Set Rapid Traverse.** Use the up and down cursor keys to select the parameter. Set it by pressing the left or right cursor keys. The value for the selected parameter changes each time you press the left or right cursor key.

There are two options available for the rapid traverse parameter:

Choosing:	Causes monochrome:	Causes color:
On	rapid moves to be drawn with a dashed line and other moves to be drawn with a solid line.	rapid moves are always in red, unless the rapid moves overwrite green feed lines. If this is the case, the control uses yellow for the rapid moves that overlap the green feed lines.
Off	graphics do not draw any move that is made in rapid traverse. All other moves are drawn with a solid line.	graphics do not draw any move that is made in rapid traverse.

4. **Set Auto Size.** Use the up and down cursor keys to select the parameter. Set auto size by pressing the left or right cursor keys. The value for the selected parameter changes each time you press the left or right cursor key.

If you turn this parameter “ON”, the control re-sizes the graphics screen to the size of the programmed part. To use this feature, turn this parameter “ON”, then run the part program. When the part program is finished, the control re-sizes the graphics screen then turns this parameter “OFF.”

5. **Set Grid Lines.** Use the up and down cursor keys to select the parameter. Set it by pressing the left or right cursor keys. The data for the selected parameter changes each time you press the left or right cursor key.

When this parameter is:	The control:
On	overlays the graphics screen with grid lines. These grid lines are solid lines drawn at the axis coordinate tick marks.
Off	does not display lines.

6. **Set Overtravel Zone Lines.** Use the up and down cursors to select the parameter. Set it by pressing the left or right cursor keys. The data for the selected parameter changes each time you press the left or right cursor key.

When this parameter is:	The control:
On	draws overtravel/zone lines on the graphics screen. These overtravel/zone lines correspond to the active software overtravel limits and the programmable zone limits. The software overtravel limits are specified in AMP by your system installer. The programmable zone limits must be programmed by the operator. The software overtravel and programmable zone lines are dotted lines.
Off	does not display lines.

7. Set the Main Program Sequence Starting #: parameter. It is only available with QuickCheck. Use the up and down cursors to select this parameter. Set it by typing in the new value for that parameter using the keys on the operator panel. Press the [TRANSMIT] key when the new value has been typed in. The old value for the sequence number is replaced with the new value.

Use this parameter to select a sequence number (within a main program) to mark where the graphics will begin drawing. Enter a sequence number that is in the main program or zero as a value for this parameter. The control does not search subprograms for sequence numbers, so the sequence number entered here must be in the main program, unless you enter a value of zero.

Graphics start drawing when the control begins executing the block that corresponds to the entered sequence number. Entering a zero as a sequence number causes graphics to begin executing the first block. Entering any sequence number not found in the main program (other than zero) stops the graphics from starting.

8. Set the Main Program Sequence Stopping #: parameter. It is only available with QuickCheck. Use the up and down cursors to select this parameter. Set it by typing in the new value for that parameter using the keys on the operator panel. Press the [TRANSMIT] key when the new value has been typed in. The old value for the sequence number is replaced with the new value.

Use this parameter to select a sequence number within a main program for the graphics to stop drawing. Enter a sequence number that is in the main program as a value for this parameter or enter a value of zero. The control does not search subprograms for sequence numbers, so the sequence number entered here must be in the main program unless entering a value of zero.

Graphics stop drawing when the control finishes execution of the block that corresponds to the entered sequence number. Entering a value of zero (or any other sequence number not found in the main program) causes the graphics to continue until the last block in the main program is finished executing.

9. **Set the Process Speed parameter.** It is only available with QuickCheck. Use the up and down cursors to select this parameter. Set it by pressing the left or right cursor keys. The data for the selected parameter changes each time you press the left or right cursor key.

Use this parameter to select the speed for the control to draw graphics. There are two options available for process speed:

Choosing:	Causes the QuickCheck with graphics to be drawn:
FAST	at the full processor speed.
SLOW	to be drawn at a slower speed. This makes it easier to determine the actual sequence of motion along the tool paths.

10. **Set Auto Erase.** It is only available with active graphics. Use the up and down cursors to select the parameter. Set it by pressing the left or right cursor keys. The data for the selected parameter changes each time you press the left or right cursor key.

Use this parameter when a part program is going to be executed many times in repetition (as in production), and when you want a clear graphics screen for displaying the part at the beginning of each part program.

When you select :	The control:
YES	clears the graphics screen at the beginning of each program execution.
NO	does not clear the graphics screen and any subsequent program executions result in the graphics screen plotting the new tool paths over the old tool paths on the screen.

Important: If you choose yes for this parameter and the first moves in a part program are rapid, it is possible that the screen may not clear before the beginning of the move. If this is the case, then a small portion of the actual tool rapid motion is not displayed after the screen clears.

11. If you want to save these parameters, press the {SAVE PARAM} softkey.

8.4.5 Graphics in Single-Block

The active and QuickCheck graphics features can run in single-block or continuous mode as described in chapter 8.

In:	This happens:
Single block	one block of a part program executes each time you press the <CYCLE START>.
Continuous mode	the control continues to execute blocks sequentially as they are read. To control the speed of the graphics plot in QuickCheck, see QuickCheck with graphic parameter "PROCESS SPEED" in chapter 1.

8.4.6 Clearing Graphics Screen

You can clear the screen if it becomes too cluttered because of many different tool paths. Clear the currently drawn tool paths by pressing the {CLEAR GRAPHS} softkey. Any tool paths on the screen are erased and the plot continues from the current tool location without stopping.

You can clear active graphics with Auto Erase. Refer to page 8-32.

8.4.7 Displaying Machine Information in Graphics

The machine information window displays the currently programmed axes positions, feedrate, active tool number, and active work coordinate system G-code along with G52, if there is a G52 offset active.

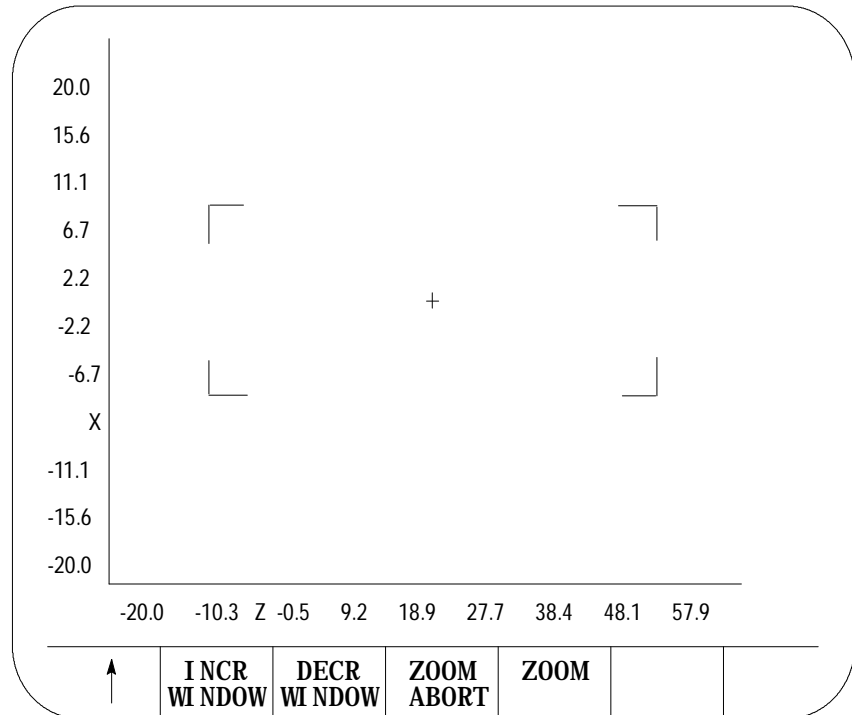
Press the {MACHINE INFO} softkey to activate the machine information window. The window toggles on and off each time you press the softkey.

Move the machine information window around the screen to prevent it from obscuring any tool paths. Press the up, down, right, and left cursor keys on the operator panel.

8.4.8 Zooming Graphics

You can take a closer look at a specified area of tool motion on the graphic display. This is helpful when you want to get a better view of the actual tool paths in that area or if the part being machined is too small to be easily seen on the current graphic display. Press the {ZOOM WINDOW} softkey to display the zoom window graphic display screen.

Figure 8.19
Zoom Window Graphic Display Screen



This screen resembles the regular QuickCheck graphics screen with the exception that it includes a window and different softkeys. Use the window to define a new size and location for the tool path graphic display.

The area within the window will become your next screen. The crossed lines (+) become the center of the screen. You must move the window to the location that you want to zoom in on to make sure that it appears on your next screen. It is helpful to run the program first so that you can see the tool path. This helps you position the tool path within the window.

If you decide not to change the window size or location, press the {ZOOM ABORT} softkey to abort the operation and return to the graphics screen.

Important: Any time you change the window size and/or location, the control erases all tool paths from the screen. These tool paths cannot be recovered. When returning to the graphics screen using a new window, the graphics plot starts from the current tool position.

To use the zoom window feature:

1. Press the {ZOOM WINDOW} softkey. This changes the display to the zoom window display.

(softkey level 3)

↑	CLEAR GRAPHS	MACHNE INFO	ZOOM WI NDO	ZOOM BACK	GRAPH SETUP	
---	-----------------	----------------	----------------	--------------	----------------	--

2. Use the cursor keys on the operator panel to move the center of the window around the screen. To move the window center at a faster rate, press and hold the [SHIFT] key while pressing the cursor keys.

Important: On this screen, the machine information window cannot be moved. The cursor keys are dedicated to the zoom window only.

If more than one window is displayed (as in the case of 3 axis graphs), choose the window to move by pressing either the {SELECT UPPER} or {SELECT LOWER} softkey. This softkey toggles between upper and lower each time that you press it. If both the upper and lower graph have a common horizontal axis, the left and right cursor keys move both windows. The current window that is the active moveable window for the 3-axes graph is displayed with a + symbol in the center of the window.

(softkey level 4)

↑	INCR WI NDO	DECR WI NDO	ZOOM ABORT	ZOOM	SELECT UPPER	
					or	
					SELECT LOWER	

Important: The {SELECT UPPER} or {SELECT LOWER} softkey is only displayed if the control is currently using a 3-axes display for the graphics screen.

If the center of the window is moved up against the boarder of the displayed axis, the window size shrinks automatically. This keeps the window within the active area of the tool paths. Otherwise, the zoom window is displayed at a default size and the cursor keys only move the window center.

- To change the size of the window, use the {INCR WINDOW} or {DECR WINDOW} softkeys. To change the window size at a faster rate, press and hold the [SHIFT] key while pressing the {INCR WINDOW} or {DECR WINDOW} softkeys.

Each time you press:	The Zoom Window :
{INCR WINDOW}	increases in size.
{DECR WINDOW}	decreases in size.

- Once the size and the location of the window are correct, press the {ZOOM} softkey to return to the regular QuickCheck graphics screen.

(softkey level 4)

↑	I NCR W I N D O W	D E C R W I N D O W	Z O O M A B O R T	Z O O M		
---	----------------------	------------------------	----------------------	---------	--	--

The graphics screen is automatically cleared of the previously plotted tool paths. The control begins plotting new tool paths from its current location in the program using the new window size.

If you press {ZOOM BACK}:	It:
one, two or three times	reverses the zoom in increments, going back to your most recent zoom (not the original).
a fourth time	returns the graphics screen to the default size.

When you press the {ZOOM BACK} softkey, the control clears the graphics screen of any previously plotted tool paths. The control resumes plotting new tool paths from its current location in the program using the current window size.

8.6
Power Turn-on Screen

When power is turned on, the control displays the power turn-on screen.

The following section discusses how to modify information displayed on this screen at power up.

Editing the System Integrator Message Lines

To edit the system integrator message lines of the power turn-on screen, do the following:

1. Press the [SYSTEM SUPORT] softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWITCH LANG		→

The control brings you to softkey level 2.

2. Press the [PTOM SI/OEM] softkey.

(softkey level 2)

	PRGRAM PARAM	AMP	DEVI CE SETUP	MONI - TOR	TIME PARTS	→
	PTOM SI /OEM		SYSTEM TIMING			→

The control changes the screen to display the PTO screen.

* Asterisks indicate the three lines for system integrator messages. The softkeys used to change these lines are password protected.

3. Use the up or down cursor keys to highlight the line that you want to change on the PTO screen. The line selected is shown in reverse video.

4. Press the {ENTER MESSAGE} softkey. This highlights the softkey, and the control displays the input prompt "PTO MESSAGE:" at the top of the screen. Also, the current text, if any, of the selected message line is shown on the input line next to the prompt. (The text may be edited like any other input string.)

(softkey level 3)

↑	ENTER MESSAGE	STORE BACKUP				
---	------------------	-----------------	--	--	--	--

5. Once the line has been edited, press the <TRANSMIT> key. This transfers the edited line to the PTO screen. After pressing the <TRANSMIT> key, you can either:
 - edit another line
6. To save changes to the system integrator message lines, press the {STORE BACKUP} softkey. When pressed, the softkey will be shown in reverse video until the control has finished storing the lines.

Important: If the messages are not stored to EEPROM, the text will be lost the next time the system is powered down.

(softkey level 3)

↑	ENTER MESSAGE	STORE BACKUP				
---	------------------	-----------------	--	--	--	--

8.7
Screen Saver

The 9/Series screen saver utility is designed to reduce the damage done to the CRT from “burn in”. Burn in is the result of the same lines or characters being displayed at the same location on the screen for a such a long period of time that they leave a permanent imprint on the CRT.

After a preset length of time without any keyboard strokes, the screen saver utility replaces the 9/Series display with a single horizontal line that continuously scrolls from top to bottom on the CRT. This horizontal line is a standard character high, green on color operator panels, amber on monochrome.

When any operator panel key, PAL keyboard request, or softkey is pressed, the display is returned to the last viewed 9/Series display. Exceptions are screens accessed through hot keys on the keyboard (such as Display Select or the PAL rung monitor). When the screen saver returns from one of these screens, the 9/Series screen previous to the hot key selection is displayed.

In the event that a system error or warning, PAL display page, PAL message, or E-Stop condition occurs while the screen saver is active, the horizontal scrolling line is replaced with a scrolling message “MESSAGE PENDING, PRESS A KEY TO DISPLAY.” The operator should press any keyboard key or softkey to return to the normal 9/Series screen and view the condition. The system installer can write PAL to disable the screen saver automatically when one of these conditions occur.

Important: The system installer has the ability to disable the softkey activation of the screen saver utility in the PAL program. Refer to your system installer’s documentation for details.

The screen saver does not impact machine operation. MTB panel, PAL, or part program requests for machine motions or functions occur regardless of the status of the CRT. You should exit the screen saver before performing manual or PAL machine motions. It is also recommended you disable the screen saver for the first few runs of a part program until you are confident machine motions are occurring as desired.

Enable the screen saver and set the activation timer value (time without any keyboard input before the screen saver starts) as follows:

1. Press the [SYSTEM SUPORT] softkey.

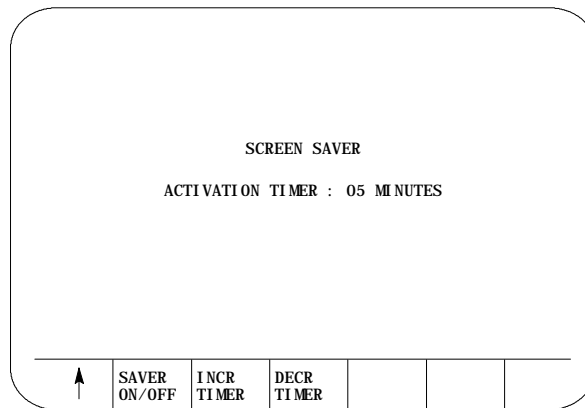
(softkey level 1)

PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
------------------	--------	----------------	-----------------	------------------	---

2. Press the [SCREEN SAVER] softkey.
(softkey level 2)

	PRGRAM PARAM	AMP	DEVI CE SETUP	MONI - TOR	TIME PARTS	→
	PTOM SI /OEM		SYSTEMS TIMING	SCREEN SAVER		→

The screen saver setup screen appears.



Press This Softkey	To:
SAVER ON/OFF	toggle between enabling and disabling the screen saver. When the softkey name is shown in reverse video, the screen saver is enabled. Note the system installers PAL program can override this softkey setting.
INCR TIMER	increase the duration of the Activation Timer by five minute increments. The activation timer can be set to a maximum of 60 minutes.
DECR TIMER	decrease the duration of the Activation Timer by five minute increments. The activation timer can be set to a maximum of 60 minutes.

The activation timer determines the length of time that takes place after the last keyboard or softkey stroke occurred before the screen saver starts.

Softkey settings that activate the screen saver and determine the activation timer duration remain active even after power is cycled.

END OF CHAPTER

Communications

9.0 Chapter Overview

This chapter covers:

- communication port parameters
- inputting part programs from a tape reader
- outputting part programs to a tape punch
- verifying saved materials
- error conditions for inputting and outputting part programs

9.1 Setting Communications

This section covers the communication port parameters that are available with the control. You use communication parameters to let the control communicate with peripheral devices.

The control has default parameters permanently stored in memory for a number of peripheral devices. These devices are listed in Table 9.A and are selected by the DEVICE parameter. You can also modify the default parameter separately for each device.

9.1.1 Setting Communication Port Parameter Values

This section explains how to change communication parameter values. A description of each of the parameters is presented in section 9.1.2.

Display the device setup screen as follows:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {DEVICE SETUP} softkey to display the device setup screen as shown in Figure 9.1.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVICE SETUP	MONI - TOR	TIME PARTS	→
↑	PTOM SI/OEM					→

The 9/230 CNC does not support port A. It uses only port B.

Figure 9.1
Device Setup Screen

E-STOP

SERIAL PORT: A
 DEVICE: DECITEK AB 8000-XPDR
 PORT TYPE: RS232C
 BAUD RATE: 1200
 PROTOCOL: LEVEL__2*
 PARITY: EVEN
 STOP BITS: 1
 DATA LENGTH: 8
 TIMEOUT: 15 SEC
 OUTPUT CODE: N/A
 AUTO FILENAME: NO
 STOP PRG END: YES
 PROGRAM END: M02, M30: YES
 M99: YES
 %: NO
 PRGRM NAME: YES

↑	SAVE CHANGE	COPY DEFLTS				
---	----------------	----------------	--	--	--	--

Important: Figure 9.1 displays a typical tape device setup screen. AUTO FILENAME, STOP PRG END, and PROGRAM END are available only if you have selected a tape reader or tape punch as your device. Refer to section 9.1.2 on selecting a device.

3. Use the up or down cursor keys to move the cursor to the parameter to be changed. The current value for each parameter will be shown in reverse video.

Important: Select both the SERIAL PORT (A or B) and the DEVICE being set first (see Figure 9.1) since all other parameters are device dependent.

4. To change a value after a parameter has been selected, press the left or right cursor keys. The control scrolls through the available parameter values as the left or right cursor key is pressed (or held).
5. To save the changes made to the communications parameters for a specific peripheral device, press the {SAVE CHANGE} softkey. The message "SAVE COMPLETED" is displayed. If the saved parameters differ from the default parameters for that device, the message "PARAMETERS DO NOT MATCH THE DEFAULTS" appears beneath the parameters.

Important: You must save changes made to the parameter values for a peripheral device before you select another device.

To restore the default parameters for a particular device, select that device using the left or right cursor key, and press the {COPY DEFLTS} softkey. The message "DEFAULTS LOADED" appears at the top of the screen.

To exit the device setup screen, press the exit {•} softkey.

Important: When you exit, the serial ports and devices displayed on the device setup screen becomes your current settings.

9.1.2 Communication Port Parameters

PORT (setting of input/output port)

The 9/240, 9/260, 9/290 and 9/440 controls have two communication ports: port A and port B. The 9/230 control uses only port B. This parameter selects which of the two ports is going to be modified. When you select port A, you see the letter "A" displays; and when you select port B, you see the letter "B" displays.

All of the following parameters can be set independently for each communication port (A or B).

DEVICE (setting type of peripheral)

Select your peripheral device immediately after selecting your serial port. The devices with default communication parameters stored in the control are listed in Table 9.A. If the device that you are using is not listed, select either USER PUNCH, USER PRINTER, or USER READER.

Important: You cannot select the same device for both peripheral ports. Once you select a device for PORT A, that device is no longer an option available for PORT B and vice-versa.

Table 9.A
Available Devices For Communication

Device	Device Type
Allen-Bradley 1770-SB	Cartridge
Ricoh PTR240R	Reader
Facit N4000	
Decitek AB 8000-XPDR	
DSI SP75	Punch
Facit 4070	
Facit N4000	
Epson LX-810 (USA)	Printer
Epson SP-500 (JAPAN)	
User Punch	User Defined
User Reader	
User Printer	
Teach Pendant	
PAL-RS232 Comm	
ODS	ODS Terminal
Generic Level_2	Computer using only level 2 protocol
Greco Minifile	Intelligent Storage
Intelligent Device	Personal Computer using Communication Software

For more information on some of these peripheral devices and their function, refer to your integration manual.

PORT TYPE

Port type options differ depending on the port you select.

Port	Type
Port A	RS232-C
Port B	RS232-C or RS422A

BAUD RATE

You can set the baud rate at these speeds (in bits per second):

300, 600, 1200, 2400, 4800, 9600, MAX

MAXIMUM BAUD RATE

If you need to operate your 9/Series control at a baud rate higher than 9600 bps, you can toggle between 19200 (default setting) and 38400 bps:

↑	SAVE CHANGE	COPY DEFLTS	SETMAX 19200	SETMAX 38400		
---	----------------	----------------	-----------------	-----------------	--	--

Important: The “SAVE CHANGES” and “COPY DEFLTS” softkeys have no affect on the maximum baud rate setting.

Although you can make your selection at any time, only when “MAX” displays is it implemented for all ports on your control. Until then, your selection is highlighted to remind you of your currently selected maximum baud rate. Once you’ve made a selection, it is effective until you make another selection.

Important: Selecting a maximum baud rate that your device does not support causes the selection to be ignored. For ODS device drives, you must cycle power on the control after changing the baud rate in order for the control to recognize your selection. If one or both ports are busy, the request to select a maximum baud rate is ignored, returning the error, “PORT IS BUSY - REQUEST DENIED”.

Important: Your “MAX” selection establishes the maximum baud rate for all serial ports on your 9/Series control. A combination of 19200 baud on one port and 38400 baud on another port is not allowed.

PROTOCOL

Select the protocol for communications from the following options.

LEVEL_1
LEVEL_2*
DF1
RAW

PARITY (parity check)

Select the parity from the following parity check schemes:

Parity	Parity Check
NONE	No parity check
EVEN	Even parity
ODD	Odd parity

STOP BIT (number of stop bits)

Select the number of stop bits with this parameter. You can select:

1, 1.5, or 2 bits

DATA LENGTH

Select the number of bits that constitute one character with this parameter. You can select:

7 or 8 bits

The parity bit is not counted as a data length bit.

TIME-OUT

Select the time-out value that specifies how long the control tries to establish communications with a peripheral. If the control cannot establish communications within the specified time-out value, an error message is displayed on the control's CRT. You can select:

3, 15, 30, or 60 seconds

or

2, 3, 5, or 10 minutes

or

UNLIMF (no time-out)

OUTPUT CODE

Select either EIA (RS-244A) or ASCII (RS-358-B) as output codes for 8 bit data lengths. Selecting 7 bit data length sets this output code to “N/A” since EIA and ASCII do not apply to this type.

AUTO FILENAME

This parameter is valid only if you are inputting part programs to the control from a tape reader (refer to DEVICE for details). This parameter is used only if your tape contains more than one part program.

AUTO FILENAME lets you either automatically assign program names to all programs or key in or read the filenames for each program individually as they are copied into memory from a multiple program tape.

Setting	Result
Yes	<p>The first program on a multiple-program tape must have its program name either keyed in manually, or there must be a program name as the first block of the first program. The control automatically creates program names for successive programs on that tape by incrementing the program name by one as each new program is read. Any program names that exist as the first block of the program are ignored for any program read after the first program on tape (though these blocks are still copied into memory they are not used as the program name). An error is generated if when reading program names into memory, the control needs to increment a program name that is greater than 99999.</p> <p>Important: If the first program name is to be read from tape (not manually keyed in), the first block of the first program must contain a program name in one of the following formats: Oxxxxx (ASCII), Nxxxxx or :xxxxx where xxxxx is any valid integer.</p>
No	<p>The program name for each program on a multi-program tape must be either entered from the keyboard or located as the first block of each program after the program start code on the tape. If no program name is found in the first block and no program name is keyed in, an error is generated.</p> <p>Important: If the program name is to be read from tape (not manually keyed in), the first block of the program must contain a program name in one of the following formats: Oxxxxx (ASCII), Nxxxxx or :xxxxx where xxxxx is any valid integer.</p>

STOP PRG END

This parameter is available only if you are reading a tape and have selected a tape reader as your device (refer to DEVICE for details). It determines if the tape reader is to stop at the end of each program or continue reading until the end-of-tape code is reached. Refer to the PROGRAM END section to determine what defines the end-of-program for your system.

Setting	Result
Yes	the tape reader stops every time it encounters a program end code.
No	the tape reader stops only if it encounters an error condition or the end of tape code.

CAUTION: If no program end code is found while reading a multi-program tape, all programs are read and merged into one program.

Important: If % is the program end code on your tape, the tape reader stops every time it encounters a “%”. consequently, even if STOP PRG END is set to “no”, the tape reader will stop at the end of each program.

The tape reader stops if % (ASCII) or ER (EIA) is found. If you selected % as the program end-code with (ER) format, you can continue reading on that tape after a % is read. If, however, you selected M30, M02, or M99, and one of these M-codes is read followed by a % (ER), the % is considered the tape end, and the directory screen is displayed.

PROGRAM END

This parameter is available only if you select a tape reader or tape punch as your device (refer to DEVICE for details). It is used to designate valid “end-of-program” codes. You can select more than one of these codes at a time.

The tape reader stops every time it encounters one of these program end-codes that has been set to “yes” on the device setup screen. This assumes that the STOP PRG END parameter has been set to “yes” (refer to the STOP PRG END section). If STOP PRG END is set to “no”, the tape reader does not stop unless % (ER) is encountered. Each program read is stored in the control’s memory.

If “%” is set to “yes”, making it a valid program end-code, no program end-code other than PRGRM NAME can be set to “yes”. If another program end-code is set to “yes”, the “%” option is automatically set to “no”. Refer to the descriptions for M-codes in chapter 10 for details.

- M02, M30 - refer to the descriptions for M-codes in chapter 10 for details
- M99 - refer to the descriptions for M-codes in chapter 10 for details
- % - also used as end-of-tape code. Refer to section on STOP PRG END

Important: If you set the “%” field to “yes”, the tape punch adds a % code after the last block of every program that is output to tape.

- PRGRM NAME - if set to “yes”, a program name is recognized as the end of the previous program. The program name must be in one of these forms where xxxxx indicates an integer from 0 to 99999:
 - Oxxxxx (ASCII)
 - :xxxxx (EIA)
 - Nxxxxx (except for N00000)

Important: If an N-code is used to designate the program name, be aware that all part program blocks that contain N-codes will be interpreted as new part programs.

9.2 Inputting Part Programs from a Tape Reader

If a program is loaded on a peripheral device, such as a tape reader, and you want to send a copy of that program to control memory, follow these steps:

1. Verify that the peripheral device is connected to the correct serial port and that the port is configured for that device (see section 9.1.1).
2. Press the {PROGRAM MANAGE} softkey. Figure 9.2 displays the program directory screen.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWIT CH LANG		→

Figure 9.2
Program Directory Screen

SELECTED PROGRAM:		
DIRECTORY		PAGE 1 OF 1
NAME	SIZE	COMMENT
012345	1.3	SUB TEST 1
TEST	3.9	NEW
MAIN	1.3	
TTTE	1.3	THIS IS A TEST PROGRAM
XXX	1.3	

5 FILES 120.7 METERS FREE

↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTRT PROGRAM	DISPLY PROGRAM	COPY PROGRAM	→
---	----------------	--------------	----------------	----------------	--------------	---

3. Press the {COPY PROGRAM} softkey.

(softkey level 2)

↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTRT PROGRAM	DISPLY PROGRAM	COPY PROGRAM	→
↑	VERI FY PROGRAM	PRGRAM COMENT	DELETE PROGRAM	RENAME PROGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

4. If the name is already entered in the program, skip step 4 and go to step 5. Otherwise enter the program name to copy by either selecting it using the up/down cursor keys or typing it in using the alphanumeric keys on the keyboard. The program name is displayed on the input line (line 2) of the screen next to the prompt "FILENAME:".

If an error is made while typing in the new program name, corrections can be made on the input line as discussed in chapter 2.

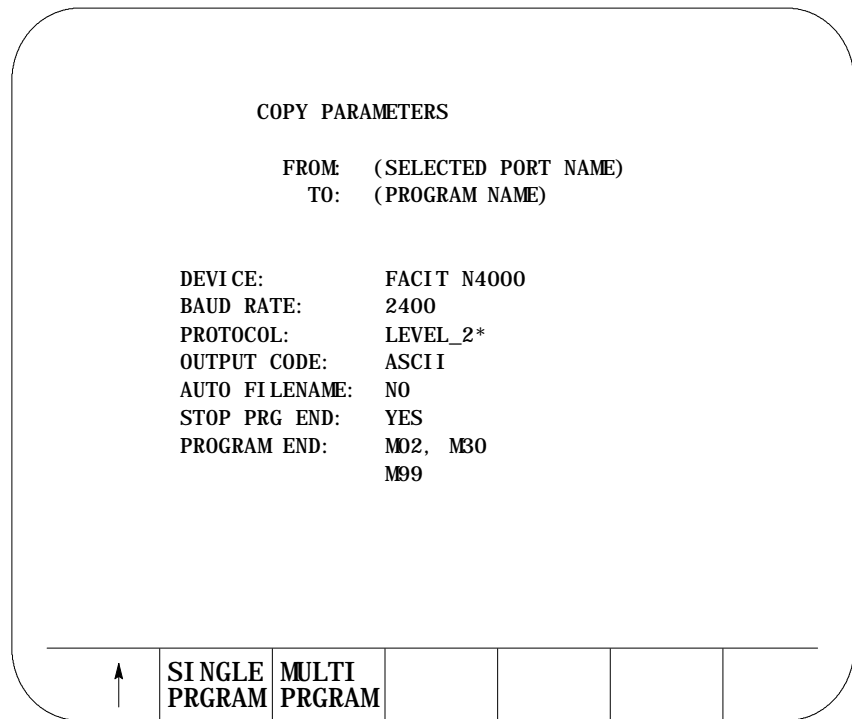
5. Select the device to copy from by using this table.

If the peripheral device is connected to:	Press this softkey:
Port A	{FROM A TO MEM}
Port B	{FROM B TO MEM}

The screen is changes to the “COPY PARAMETERS” screen (Figure 9.3) and displays the current device and setup parameters for that communication port.

If the device displayed on the screen is not correct, select the correct device using the procedure described in section 9.1.1

Figure 9.3
Copy Parameter Screen



6. Specify if you want to copy one program or multiple programs.

- **Input Single Program**
Press {SINGLE PROGRAM} to copy one program from tape. Input terminates when the first program end or tape end code is encountered.
- **Input Multiple Programs**
Press {MULTI PROGRAM} to copy multiple programs from the tape into memory.

If STOP PRG END was set to	the tape reader
"yes"	stops each time it encounters a program end or tape end code.
"no"	continuously reads programs until it encounters a tape end code.

For details on how multiple programs are input and named, refer to the AUTO FILENAME, STOP PRG END, and PROGRAM END parameters discussed in section 9.1.2.

When you input a program, the control does not erase the program from the peripheral device. If the program being read contains a comment in the first block, the control loads the comment and displays it with the program name in the program directory.

If the selected program name already exists in control memory, the following prompt appears: "OVERWRITE PROGRAM IN MEMORY (Y/N):" Type "Y" to start the tape reader and overwrite the existing program in memory with the new program. Type "N" to abort the copy operation.

To abort the copy operation at any time, press the {CANCEL} softkey.

(softkey level 4)

					CANCEL
--	--	--	--	--	--------



CAUTION: If you select overwrite, the control will erase the existing program from its memory prior to replacing it with the new program. If an error occurs during the input operation, or you abort the copy operation, the original program will be lost, and the new one will not be copied.

9.3 Outputting Part Programs to a Tape Punch

If a program is in control memory and you want to send a copy of that program to a peripheral device, follow these steps:

1. Verify that the peripheral device is connected to the correct serial port and that the port is configured for that device (see section 9.1.1).
2. Press the {PROGRAM MANAGE} softkey. The control displays the screen shown in Figure 9.4.

(softkey level 1)

	PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

Figure 9.4
Program Directory Screen

SELECTED PROGRAM						
DI RECTORY		PAGE 1 OF 1				
NAME	SIZE	COMMENT				
012345	1.3	SUB TEST 1				
TEST	3.9	NEW				
MAIN	1.3					
TTTE	1.3	THIS IS A TEST PROGRAM				
4 FILES 127.5 METERS FREE						
↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTRT PROGRAM	DISPLY PROGRAM	COPY PROGRAM	→

3. Press the {COPY PROGRAM} softkey.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	I NPUT DEVI CE	→
↑	REFORM MEMORY					→

4. Enter the program name to output from memory. There are two ways to do this:

- Type in the program name using the alphanumeric keys on the keyboard. The control displays program name on the input line (line 2 of the screen) next to the prompt "FILENAME".
- Use the up or down cursor keys to position the cursor at the program it is desired to send. The control displays the selected program in reverse video.

If outputting all programs, cursor to the first program in your program directory.

5. Select the device you want to copy as follows:

If the peripheral device is connected to:	Press this softkey:
Port A	{MEM TO PORT A}
Port B	{MEM TO PORT B}

(softkey level 3)

↑	MEM TO PORT A	FROM A TO MEM	MEM TO PORT B	FROM B TO MEM	MEM TO MEM
---	------------------	------------------	------------------	------------------	---------------

The Copy Parameters screen appears (see Figure 9.5).

6. Specify if you want to output one, multiple, or all programs onto tape.

- **Output Single Program**

Press {SINGLE PRGRAM} to output the program selected in step 4.

- **Output Multiple Programs**

Press {MULTI PRGRAM} to output more than one program. After you pressed the {MULTI PRGRAM} key, the program selected in step 4 is output. The Program Directory Screen (refer to Figure 9.4) appears with the following set of softkeys:

(softkey level 4)

OUTPUT PRGRAM	FINISH TAPE				
------------------	----------------	--	--	--	--

Select the next program to output by selecting the file name as described in step 4. Press the {OUTPUT PRGRAM} key to output the program.

For details on how multiple programs are named, refer to chapter 10 and the end of section 9.1.2.



CAUTION: Be aware that the information in the directory such as program name and comment are not output to tape. If you want to transfer this information to tape, you must include the program name and comment from the directory in the first block of the program. Refer to chapter 10 for details on “O” word program name.

- After the last program is output to the tape, press {FINISH TAPE} to punch the tape end and trailer code to the tape.

Important: If an error occurs or you abort the copy operation while punching a tape, the tape may not be useable.

- **Output All Programs**
Press {OUTPUT ALL} to copy all programs in memory to tape at one time.
 - {OUTPUT ALL} works like {MULTI PRGRAM} except that you cannot select the programs you want to output. {OUTPUT ALL} selects all programs automatically and outputs them to the peripheral device.
 - All programs are copied to the peripheral device and stored using the same program name as the original, in the order that they appear on the Program Directory Screen.

(softkey level 3)

↑	SINGLE PRGRAM	MULTI PRGRAM	OUTPUT ALL			
---	------------------	-----------------	---------------	--	--	--

Figure 9.5
Copy Parameters Screen

COPY PARAMETERS

FROM: (Program Name)
TO: (Selected Port Name)

DEVICE: FACIT N4000
BAUD RATE: 2400
PROTOCOL: LEVEL_2*
OUTPUT CODE: ASCII
AUTO FILENAME: NO
STOP PRG END: YES
PROGRAM END: M02, M30
M99

					CANCEL
--	--	--	--	--	--------

Important: Press {CANCEL} to abort the copy operation at any time.

9.4
Verifying Part Programs
Against Source Programs

To verify that a part program stored in memory matches a source program stored in memory or on a peripheral device:

1. If one of the programs to either verify or verify against is on a peripheral device, make sure that the peripheral device is connected to the correct serial port and that the port is configured for that device (see section 9.1.1).

2. Press the {PROGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

3. Press the {VERIFY PROGRAM} softkey.

(softkey level 2)

↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DI SPLY PRGRAM	COPY PRGRAM	→
↑	VERI FY PRGRAM	PRGRAM COMENT	DELE TE PRGRAM	RENAME PRGRAM	IN PUT DEVI CE	→
↑	REFORM MEMORY					→

4. Type program name(s) into the input area (lines 1- 2) in the following manner. If both programs are resident in control memory, type both program names separated by a comma. The order in which these names are entered is not important.

VERIFY: program name, program name

If one of the programs to verify is on a peripheral device and the other is resident in control memory, it is only necessary to type the name of the program resident in the control.

VERIFY: program name

5. To verify a part program in memory against a part program stored on a peripheral device, press the {VERIFY PORT A} or {VERIFY PORT B} softkey depending on where the peripheral device is connected.

To verify a part program in memory against another part program in memory, press the {VERIFY MEMORY} softkey.

(softkey level 3)

↑	VERI FY PROT A	VERI FY PORT B	VERI FY MEMO RY			
---	-------------------------	-------------------------	--------------------------	--	--	--

6. Press the {VERIFY YES} softkey. To abort the verify operation press the {VERIFY NO} softkey.

(softkey level 4)

↑	VERI FY YES	VERI FY NO				
---	-------------------	------------------	--	--	--	--

The control displays one of the following messages when the verify operation is performed:

“PROGRAMS ARE DIFFERENT” if programs do not match.

“PROGRAMS ARE IDENTICAL” if programs match.

9.5 Error Conditions (Inputting and Outputting Part Programs)

An error occurs if you attempt these operations:

- output data to an input device (tape reader, etc.)
- output or input data when no peripheral device is connected to the selected port
- output or input data when the data communication line is either broken or not connected
- input data from an output device (tape puncher, etc.)
- input a program name that already exists in control memory. The control displays “OVERWRITE PROGRAM IN MEMORY(Y/N):”
- input a program without reading in or typing in a filename when AUTO FILENAME is set to “no”

END OF CHAPTER

Introduction to Programming

10.0 Chapter Overview

The control performs machining operations by executing a series of commands that make up a part program. These commands are interpreted by the control which then directs axis motion, spindle rotation, tool selection, and other CNC functions.

Part programs can be executed from the control's memory or from a CNC tape. Programs on tape can be executed directly from the tape, or can be loaded into the control and executed from memory.

This chapter begins with an explanation of CNC tape format. The remainder of the chapter deals with the contents of a part program. It explains blocks, words, G-codes, M-codes, etc.

For information on how to enter and edit part programs through the keyboard, refer to chapter 5. For information on how to edit part programs off-line and download them to the control, refer to chapter 6. For more information on how to enter a part program from a peripheral device, refer to chapter 9.

This chapter covers:

- part program tape format
- program configuration
- block skip
- using subprograms
- word formats, functions, and descriptions

10.1 Tape Format

This section describes the format that the control requires for inputting and outputting part programs from or to a tape device.

The control accepts:

- ASCII format (RS-358B)
- EIA-244-B format

Important: In addition to the control's normal tape format (described in this section), the control is also equipped to read and execute standard Allen-Bradley 7320 and 7360 CNC tapes. Refer to Appendix D of this manual for details.

Tape with Program End = M02, M30, M99

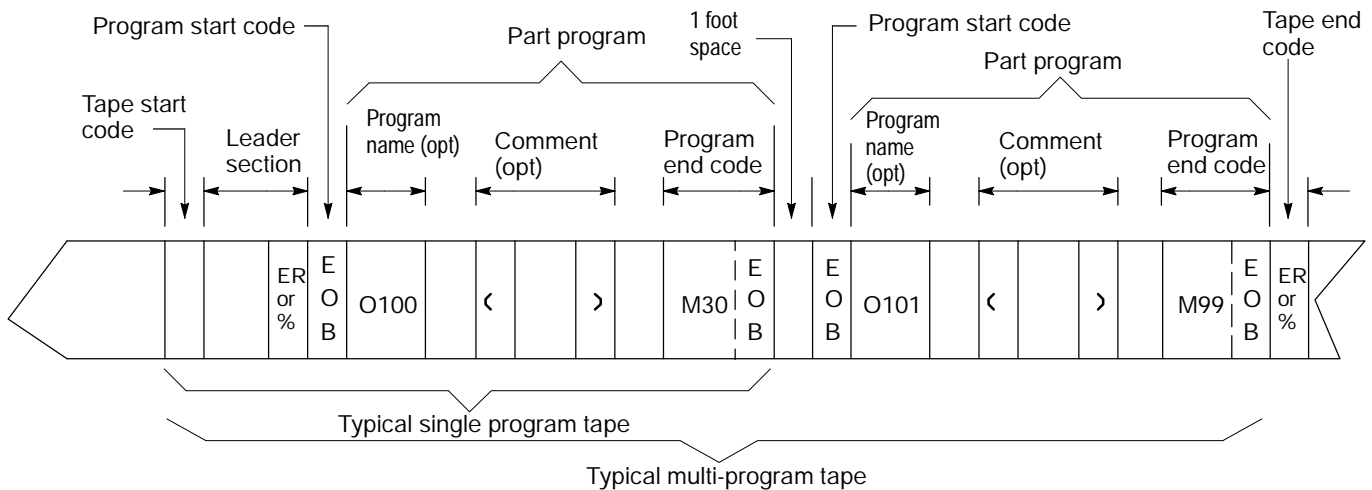
This particular tape format allows single- or multi-program format on a tape. It also allows you to enter either M02, M30 or M99 as a program end code (refer to chapter 9 for details on legal program end codes).

Figure 10.1 shows a typical configuration for a multiple program tape with M30 and M99 as program end codes.

If the first block of the file program is a filename, you can set AUTO FILENAME to “yes” to create filenames for all subsequent programs. If AUTO FILENAME is set to “no”, and no filenames are included in the programs on the tape, you have to enter all filenames from the keyboard when prompted. For more information, refer to the section on O-word Programming in this chapter.

The control automatically recognizes EIA or ASCII during input when it reads the first EOB code from the tape.

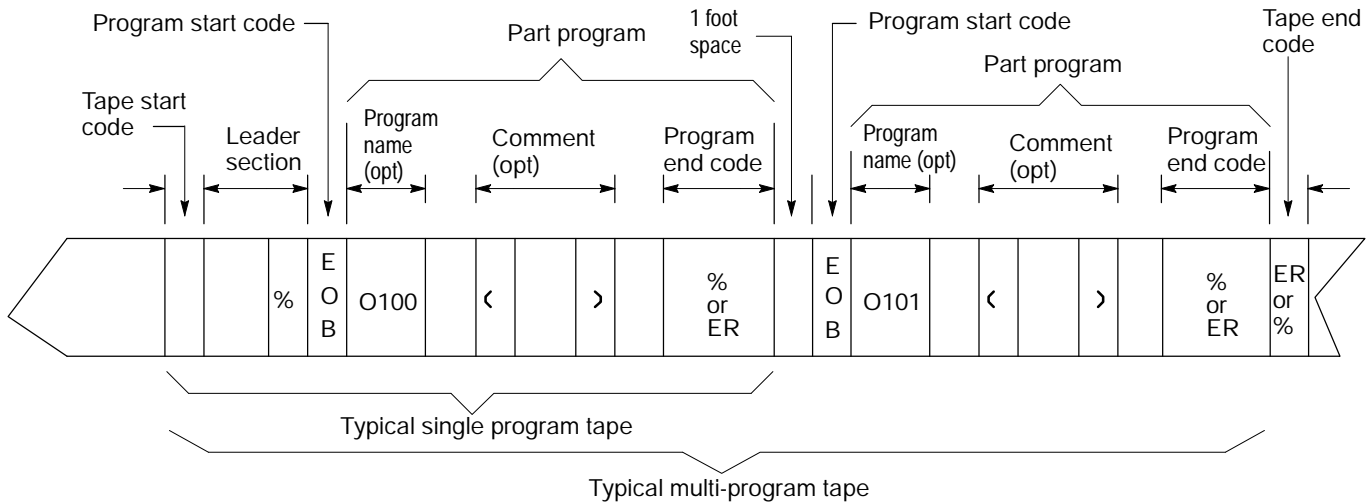
Figure 10.1
Tape Configuration (Program End = M02, M30, M99)



Tape with Program End = % (ASCII), ER (EIA)

Unlike the previous tape type mentioned, this type of tape accepts only the “%” (ER) field as the program end code (see Figure 10.2). Refer to chapter 9 for details on legal program end codes and the effect of STOP PRG END.

Figure 10.2
Tape Configuration (Program End = % (ASCII), ER (EIA))



This tape format should conform to the variable block length format specified by EIA standard RS-274D.

The control automatically recognizes EIA or ASCII during input when it reads the first EOB code from the tape.

(1) Tape Start (Rewind, Stop Code)

The tape start code, indicating the beginning of a tape, is designated with:

- % character -- ASCII format
- ER -- EIA format

This code must be on the tape if programming a tape rewind code (M30, M99) in the part program. In other cases (M02, %), it is not necessary to have this code on the tape. Any tape punched from the control has the tape start code.

(2) Leader Section

The information between the tape start and the program start is called the tape leader section. The leader section is simply a tape indexing section. On punched tape, the holes punched in the leader section can be configured to show alphanumeric characters. The control ignores information within the leader section and does not perform a parity check on this information.

Important: A program start code must not appear within the leader section. If the program start code appears, the control starts reading information and assumes that it is the part program. This causes parity errors or “nonsense” codes read in by the control.

(3) Program Start Code

The first end-of-block code (EOB code) after the leader section indicates the beginning of the part program. The EOB code is designated with:

- line feed (LF) -- ASCII format
- carriage return (CR) -- EIA format

Important: When performing an EOB search, the search is executed from the beginning of the part program, NOT from the point of display.

(4) O-word Program Name

The program name, if on the tape, must follow the program start code.

When outputting to tape the program name can be determined by:

Program Name:	If:
Manually Keying in the program name	--
Selected from the first block of the program	No name has been manually keyed in and the first block contains an O-word, : (for EIA format), or N-word after the program start code.
Selected from the program directory	No name has been manually keyed in and no name exists in the first block of the program. For the program name to be taken from the directory screen the program must be stored on that directory with the letter O followed by up to 5 numeric digits. In this case the control will generate an O-word block as the first block of the program written to tape.
No name is used	A program will be stored on tape with no program name if none of the above naming conditions are met.

(5) Part Program Section

A part program is made of blocks of information. Individual blocks contain words that the control can interpret. Individual words are made up of an address with a numeric value. For details on words and addresses, see the remaining sections of this chapter.

This section should include a program name, program blocks, comments, and end-of-program. Each block in the part program is separated by an EOB code. The control displays a semicolon “;” to indicate the presence of an EOB code.

(6) Comment

Information punched between the control out code “(” and the control in code “)” within the program section is considered a comment and is not handled as significant information (even though it is copied to and from control memory or tape). Any number of comments can be included in a part program interspersed with program blocks or words.

Example 10.1
Comments in Part Programs

```
X1. Z1. (START MOTION);
(CALL SUBPROGRAM 1);
(12345);
```

Each of the above program blocks contains a comment. The control does not regard the comment as significant information, except when it appears in the first block of the program. In this case, the comment is displayed on the program directory screen as part of the program name.

The comment can be up to 128 characters long (including the control out and control in codes), and can consist of any alphanumeric characters and special symbols. However, the comment cannot include the following codes:

```
(
)
ER, %      (rewind stop codes)
EOB       (end of block)
```

Important: If rewind stop codes are included in the comment, the tape rewind function (M30) causes the rewind to stop in the comment section of the tape.

(7) Program End

The control stops reading the part program when it reads a program end code. The following codes can be used as program end codes:

- M02 -- end of program
- M30 -- end of program and tape rewind
- M99 -- end of program, tape rewind, and restart the program
- % -- end of program and end of tape code
- PRGRM NAME -- end of program

(8) Tape End (Rewind, Stop Code)

The tape end code, indicating the end of a tape, is designated with either:

- % -- ASCII format
- ER -- EIA format

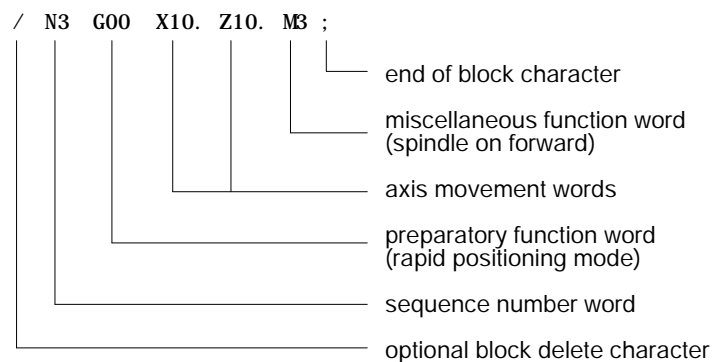
10.2 Program Configuration

Each machining operation performed by the control is determined by the control's interpretation of a group of words (commands) called a "block." Individual blocks in a part program define each machining process. Part programs consist of a number of blocks that together define a complete operation on a part.

Part program blocks are made up of:

- characters -- A character is a number, a letter, or a symbol that has a specific meaning for the control. For example, 1, G, ; are characters the control recognizes as meaningful information
- addresses -- An address is a letter that defines the instruction for the control. Examples of addresses are: G, X, Z, F
- words -- A word consists of an address followed by a numeric value. Examples of words are: G01, X10.5, F50., M2. Each word requires a specific format for its numeric part. These formats are given in section 10.5
- codes -- There are industry standards for many of the G and M-codes used here.
- parameters -- The control has a number of fixed cycles that are initiated by a specific G-code. Other words appearing in those G-code blocks are referred to as "parameters", because their values are relevant only to that G-code. For example, a Z-word generally refers to a Z axis move, but when it appears in a block with a G83 peck drilling cycle, its value refers to the depth of the hole to be drilled. In that case, it is a "parameter" of the G83 fixed cycle

A block is a set of words and characters that define the operations of the control. For example:



The control sequentially executes blocks in a part program to conduct the required machining operation.

Important: To make jumps, loops, or calculations within an executing program or subprogram use the paramacro features as discussed in chapter 28. A part program has a:

- beginning -- setting up the control and the machine to perform the operations wanted.
- middle -- performing the machining operations
- end -- returning the machine to a safe stop position, and preparing the control for the next part program.

The blocks programmed vary for each section of the program. For an example, consider the following simple program.

Example 10.2
Simple Part Program

G91G21;	- begi nni ng
G00X28. ;	
G33Z- 46. E4. ;	- mi ddl e
G00X5;	
Z2. ;	- end
M02;	

A complete part program may consist of a main program and several subprograms. For more on subprograms, see section 10.3.

10.2.1 Program Names

Enter up to 8 alphanumeric characters for program names, which the control uses to call up programs for editing or automatic operations.

Subprograms are designated with the letter O followed by 5 numbers. If a new program name is entered with 5 numeric characters, the control assumes that it is a subprogram and automatically inserts the letter O as the first character in the name. However, the control does not consider programs with more than 5 numeric characters as subprograms.

Entering Program Names

To enter a program name, do the following:

1. Press the softkey {PROGRAM MANAGE}. This calls up the program directory, which lists subprograms first, then programs by alphabetical order.
2. Type in the name of a new program or one already listed. You cannot enter spaces or special characters.
3. Press {EDIT}, which will initiate the editing mode for the program selected.

Important: When the control reads a program from tape, the O-word is the program name. If there is no O-word, the control interprets the first sequence number as the program name.

Example 10.3 Entering Subprogram Names

Name entered	Program name stored by control
000123	000123
0123	000123
123	000123
12345	012345

Example 10.4 Legal Program Name Blocks

```
012345;  
012345(TAPPING PROGRAM);  
0333  
02;
```

10.2.2 Sequence Numbers

Each block in a part program can be assigned a sequence number to distinguish one block from another. Sequence numbers begin with an N address followed by a one to five digit numeric value.

Sequence numbers can be assigned at random to specific blocks or to all blocks if desired. Blocks assigned sequence numbers can be called later by designating their sequence number. Sequence numbers are necessary to make program jumps and to specify a block for paramacro calls and returns.

Below is an example of two blocks with sequence numbers 10000 and 10010.

Example 10.5
Blocks With Sequence Numbers

```
N10000 X5. Z4. ;
N10010 X2. Z2. ;
```

Typically when assigning sequence numbers to blocks the N-word comes first in the block except when designating block delete (see section 10.2.4). It is not necessary to program the N-word first in the block, the control will still find it for jumps; however, it will not be found for renumber operations.

If more than one N-word is in a block, the control only uses the first N-word encountered for that block's number.

Different blocks may be assigned the same sequence number. If this number is called by a "GOTO" or some other command the first block found by the control with the required sequence number calling block is used. Note that the control searches for the sequence number in the forward direction (from the calling block) first. Then it searches in the reverse direction (from the calling block). How the control reacts if the sequence number is not found is determined by the specific operation being used.

The control has a programming feature that will renumber existing sequence numbers or assign all block sequence numbers. For details refer to chapter 5.

10.2.3 Comment Blocks

Information between the control out code “(” and the control in code “)” within a part program is regarded as a comment and not handled as significant information. The comment can be described in up to 128 characters (including the control out/in codes) consisting of alphanumeric characters and special symbols.

Example 10.6
Program Block With A Comment

```
N00010G91X5. (CHANGE TO INC. MOVE X 5);
```

10.2.4 Block Delete and Multi Level Delete

When programming a slash “/” followed by a numeric value (1-9) anywhere in a block, the control will skip (not execute) programmed commands in that block. The block delete feature is turned on either with the {FRONT PANEL} softkey, or with an optionally installed switch on the MTB panel. If the {FRONT PANEL} softkey is used, only block delete /1 is available. If the system installer has installed switches on the MTB panel to activate block delete, up to 9 different switches may be used, each corresponding to a different block delete character (/1 - /9).

The system installer determines in AMP if the entire block is deleted or if only the characters to the right of the block delete / are deleted. If the entire block is to be deleted, it is done regardless of the position of the / character in the block.

Example 10.7
Using Block Delete in a Part Program

Program Block	Comment
N1000 X__ Z__;	first block
N1010 Z__;	second block
/1N1020 X__;	control skips this block if switch 1 is on
/1/2N1030 X__;	control skips this block if switch 1 or switch 2 are on
/N1032 X__;	control skips this block if switch 1 is on
N1040 X__;	
/2N1050 X__;	control skips this block if switch 2 is on

The control always reads several blocks into its buffer memory so that it can prepare for moves and commands before it executes them. The switch (or softkey) controlling a block delete must be set before the control reads a block from buffer memory; otherwise, a skip will not occur.

The control considers a “/” without a number to mean “/1”. However, “/1” must be programmed if more than one block delete number is to be used in a block.

The block delete is active for sequence number search and dry run operations.

The control ignores the block delete when loading a part program from tape or other device into control memory. The control also ignores the block delete when a part program is saved on punched tape or other device from control memory.

For details on the block delete switch(s), see your PAL reference manual and the documentation prepared by the system installer.

10.2.5 End of Block Statement

All program blocks must have an end of block statement (EOB) as the last character in the block. This character tells the control how to separate data into blocks. The control uses the “;” to mark the end of a block.

Important: When performing an EOB search, the search is executed from the beginning of the part program, NOT from the point of display.

To specify an end of block character “;” use the [EOB] key on the operator panel.

For programs created offline, a linefeed designates EOB.

10.3 Using Subprograms

When the same series of blocks are repeated more than once it is usually easier to program them using a subprogram.

The key difference between a subprogram and a G65 paramacro is that a paramacro always gets a new set of local parameters, a subprogram uses the same set of local parameters that the main program used. See chapter 28 for details on paramacros and local parameters.

This section explains the following:

- 1) Main and subprograms
- 2) Subprogram calls

Important: To make jumps, loops, or calculations within an executing program or subprogram use any of the paramacro features as discussed in chapter 28.



CAUTION: Any edits that are made to a subprogram or paramacro program (as discussed in chapter 5) that have already been called for automatic execution are ignored until the calling program is disabled and reactivated. Subprograms and paramacros are called for automatic execution the instant that the calling program is selected as active (as discussed in chapter 5).

It is possible to call a subprogram from the MDI command line.

10.3.1 Subprogram Call (M98)

Generally, programs are executed sequentially. When an M98Pnnnnn (“nnnnn” representing a subprogram number) command is entered in a program, the control will merge the subprogram, designated by the address P, before the block that immediately follows the M98 command. Note that the control will issue the error message “CANNOT OPEN SUBPROGRAM” if it can not find the subprogram designated by the M98 command.

For example,

```
M98 P00001 ;
```

would cause execution to transfer from the current program to the subprogram numbered 00001.

Important: For a program to be used as a subprogram it must have a program name starting with the letter O followed by up to a 5 digit numeric value. When calling the subprogram with a P-word only the numeric value is used, and the letter O is omitted.

Sometimes it is desirable to execute a subprogram more than one time. For example,

```
M98PnnnnnLmm;
```

would cause the subprogram numbered nnnnn to be merged in the main program mm times. When an L command is entered in a M98 command, the control will merge the subprogram, designated by the address P, before the block that immediately follows the M98 command the number of times designated by the L-word. Note that both the P- and L-words must follow the M98 command in a program block.

Omission of an L-word is regarded as L1. An L-word can not be a negative value or have a value of zero.

Important: If M02 or M30 codes are found in a subprogram before the program reads an M99 they will stop program execution. The program will reset or rewind if an M30 code is executed or the program will end if the M02 code is executed.

10.3.2 Main and Subprogram Return (M99)

M99 code acts as a return command in both sub- and main programs. There are specific differences, however, when the code is used in a sub program and when it is used in a main program.

Using M99 in a Main Program

When used in a main program, M99 does the following:

- executes all commands in the block, regardless if information is programmed in the block to the right of the M99 command
- clears all modal codes similar to an M02 or M30 (simulates start-up conditions)
- resets the current main program to the first block
- automatically performs a cycle start on the program after it is reset and program execution starts over.

Using M99 in a Subprogram

When used in a subprogram, M99 does the following:

- tells the control the end of a subprogram
- does not merge any commands within a subprogram that follow the M99 code into the main program

Program the M99 code anywhere in a program block provided no axis words are programmed to the left of M99. Any information (other than axis words) programmed to the left of M99 is executed as part of the subprogram, while information (including axis words) programmed in the block to the right of the M99 command is ignored.

Important: If you want to merge a file the into the calling program, it is not necessary to program a M99 code in the subprogram unless it is in a block immediately before the end of program command (M02 or M30) in the subprogram.

```
M99X10; -- X10 is ignored in this subprogram block  
X10M99; -- X10 generates an error in this subprogram  
M03M99; -- M03 is executed as normal in this subprogram
```


Example 10.8
Subprogram Calls and Returns

MAIN PROGRAM	SUBPROGRAM 1	SUBPROGRAM 2
(MAIN PROGRAM);	(SUBPROGRAM 1);	(SUBPROGRAM 2);
N00010. . . ;	N00110;	N00210;
N00020. . . ;	N00120. . . ;	N00220. . . M99;
N00030M98P1;	N00130M99;	
N00040. . . ;	N00140. . . ;	
N00050. . . ;	N00150M30;	
N00060M98P2L2;		
N00070M30;		

The following path of execution will result when the main program above is selected as the active program.

```

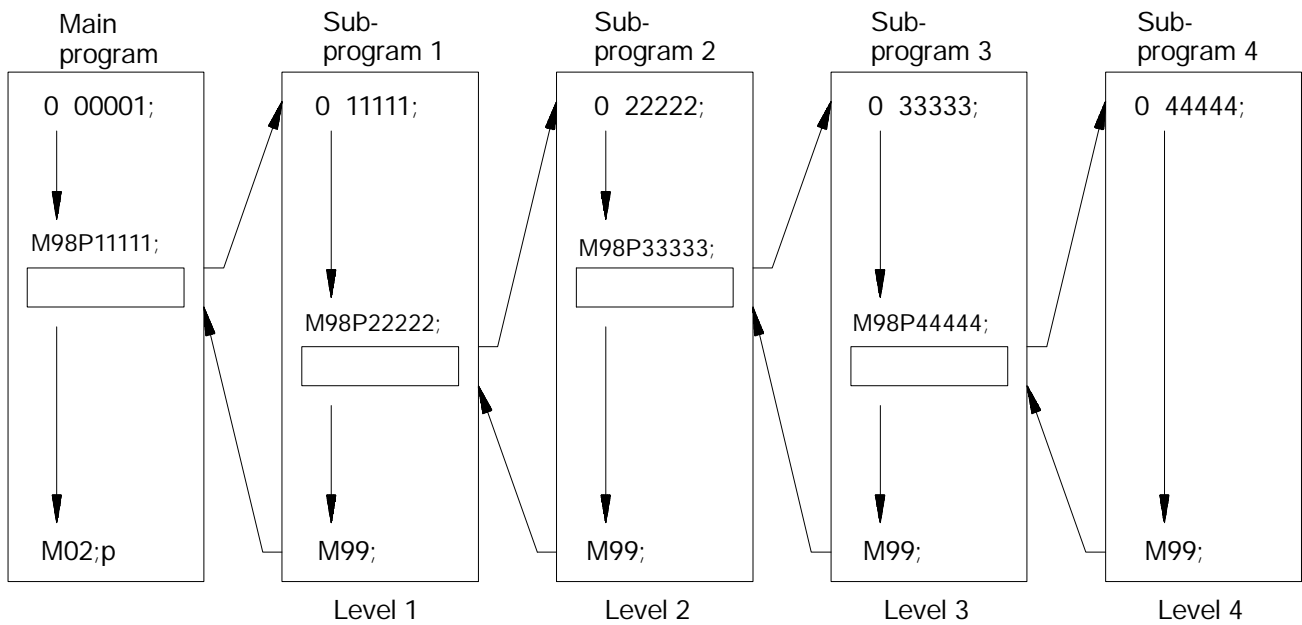
(MAIN PROGRAM) ;
N00010. . . ;
N00020. . . ;
N00030M98P1;
(SUBPROGRAM 1) ;
N00110;
N00120. . . ;
N00130M99;
N00040. . . ;
N00050. . . ;
N00060M98P2;
(SUBPROGRAM 2) ;
N00210;
N00220. . . M99;
(SUBPROGRAM 2) ;
N00210;
N00220. . . M99;
N00070M30;

```

10.3.3 Subprogram Nesting

Nesting is the term used to describe one program calling another. The program called is said to be a nested program. When a subprogram is called from the main program it is said to be on the first nesting level or nesting level 1. If that subprogram in turn calls another subprogram the called subprogram is said to be in nesting level 2. Subprograms may be nested up to a maximum of 4 levels.

Figure 10.3
Subprogram Nesting



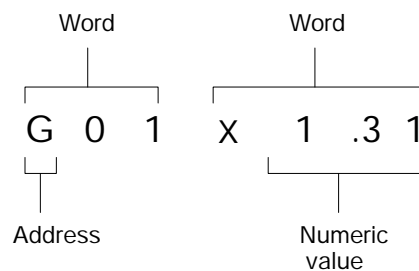
Important: Calling a macro (see chapter 28) does not add to the nesting level of any active subprograms. Up to four subprograms may still be nested though the combined total of nested macros and subprograms may not exceed 8.

10.4 Word Formats and Functions

Words in a part program consist of addresses and numeric values.

- Address -- A character to designate the assigned word function.
- Numeric value -- A numeral to express the event called out by the word.

Figure 10.4
Word Configuration



For each word used in a part program, there is a format that designates the number of digits allowable as a numeric value for that word. The format for an M-code, which is a word, for example, is normally M2 which indicates that an M-address can be followed by only two digits.

For words that allow the use of a decimal point in a numeric value, the decimal point format is used. In this case, the numeral to the left of the decimal point indicates the number of digits acceptable as integers, and the numeral to the right of the decimal point indicates the number of fractional digits acceptable.

The format "X3.4" for an X-word, indicates that three digits to the left and four digits to the right of the decimal are acceptable as numeric values. With this format selected, the maximum programmable value for an X-word would be 999.9999.

The system installer can choose from the following programming format types in AMP:

- Leading Zero Suppression -- enable or disable
- Trailing Zero Suppression -- enable or disable

Table 10.A shows the effects of leading zero suppression (**LZS**) and trailing zero suppression (**TZS**). It presumes that the system installer has set a format of X5.2 (integer 5 digits, decimal 2 digits) in AMP. Different formats would result in different decimal point placement compared to those shown on the following page, but the end result would be comparable.

Table 10.A
How The Control Interprets Numeric Values

Programmed X Value	Position Interpreted by the control		
	TZS Disabled LZS Disabled	TZS Disabled LZS Enabled	TZS Enabled LZS Disabled
X123456.	ERROR	ERROR	ERROR
X12345.6	12345.60	12345.60	12345.60
X1234.56	1234.56	1234.56	1234.56
X123.456	123.45	123.45	123.45
X12345	12345.00	123.45	12345.00
X012345	ERROR	123.45	1234.50
X123456	ERROR	1234.56	12345.60
X1234567	ERROR	12345.67	12345.67
X12345678	ERROR	ERROR	ERROR

Using LZS and TZS with G-Codes

The following table illustrates how the control interprets different G-Codes in leading zero and trailing zero suppression modes.

Leading Zero Suppression Mode (decimal assumed at end if not programmed)		Trailing Zero Suppression Mode (2-digit G-code assumed unless decimal point programmed)	
Program this:	Results in this:	Program this:	Results in this:
G02	2	G02	2
G2	2	G2	20
G2.	2	G2.	2
G92	92	G92	92
G920	920	G920	920 or 92 (if no AMP defined macro 920)
G92.1	92.1	G92.1	92.1

Important: If backing up a table using a G10 program (such as the offset tables or coordinate system tables), keep in mind the G10 program output is generated in the current format of the control (LZS or TZS). If you intend to transport this table to a different machine it must also be using the same format.

Programming without Numeric Values

The system installer can also set an AMP parameter to generate an error or use a value of zero for characters that are programmed without numeric values. If this AMP feature is disabled, programming:

```
GX;    rapid move to X zero  
M;     program stop
```

would result in the actions described in the comments following the blocks. If the feature is enabled, the error “NUMERIC MISSING” would have occurred upon execution of either of those blocks.

Word Descriptions and Ranges

Table 10.B shows, in alphabetical order, the addresses for words that are recognized by the control, their typical formats, and their general meanings. Since most of these formats are configured in AMP, refer to the documentation prepared by the system installer.

Many of the addresses themselves can be altered in AMP. This table assumes the most common names (such as X and Z for the main axes). Alterable addresses are indicated by the note “AMP assigned”.

Later sections discuss these words in more detail, including variations in their meanings when they are associated with certain G-codes. All words discussed in this manual assume that the format and addresses in the following table have not been changed by the system installer.

Table 10.B
Word Formats and Descriptions

Address	Valid Range Inch	Valid Range Metric	Function
A	8.6	8.5	Rotary axis about X (AMP assigned)
	3.3	3.3	Angle in QuickPath Plus programming
B	8.6	8.5	Rotary axis about Y (AMP assigned)
	3.0	3.0	Second miscellaneous function (AMP assigned)
C	8.6	8.5	Rotary axis about Z (AMP assigned)
	8.6	8.5	Chamfer length in QuickPath Plus programming
D	3.0	3.0	Tool radius compensation number
	8.6	8.5	Fixed cycle parameter.
E	2.6	3.7	Thread lead
F	8.6	8.5	Feedrate function (F-word)
G	2.1	2.1	Preparatory function (G-code)
H	6.0	6.0	Tool length offset number
	8.6	8.5	Parameter for probing cycles.
I	8.6	8.5	X arc center in circular interpolation
	8.6	8.5	X lead in helical interpolation
	8.6	8.5	Parameter in fixed cycles (AMP assigned)
	8.6	8.5	Exit move vector in cutter compensation
J	8.6	8.5	Y arc center in circular interpolation
	8.6	8.5	Y lead in helical interpolation
	8.6	8.5	Parameter in fixed cycles (AMP assigned)
	8.6	8.5	Exit move vector in cutter compensation
K	8.6	8.5	Z arc center in circular interpolation
	8.6	8.5	Parameter in fixed cycles (AMP assigned)
	8.6	8.5	Exit move vector in cutter compensation
L	3.0	3.0	Number of repetitions
M	3.0*	3.0*	Miscellaneous function
N	5.0	5.0	Sequence number
O	5.0	5.0	Program name
P	5.0	5.0	Subprogram name
	5.3	5.3	Length of dwell in G04 and fixed cycles
Q	8.6	8.5	Parameter in fixed cycles
R	8.6	8.5	Arc radius
	8.6	8.5	Return point in fixed cycles
	8.6	8.5	QuickPath Plus radius designation

* Some of the system basic M-codes are 3.1 format however all system installer defined M-codes must remain 3.0. See Table 10.F on page for details.

Table 10.B
Word Formats and Descriptions

Address	Valid Range Inch	Valid Range Metric	Function
S	5.3	5.3	Spindle rpm function
	5.3	5.3	Spindle Orient
	4.3	3.3	CSS
T	6.0	6.0	Tool selection function
U	8.6	8.5	Incremental axis name (Lathe A only)
	5.3	5.3	Length of dwell in G04 and fixed cycles.
V	8.6	8.5	Incremental axis name (Lathe A only)
W	8.6	8.5	Incremental axis name (Lathe A only)
X	8.6	8.5	Main axis (AMP assigned)
	5.3	5.3	Length of dwell in G04
Y	8.6	8.5	Main axis (AMP assigned)
Z	8.6	8.5	Main axis (AMP assigned)

¹Important: The formats in this table indicate the maximum number of digits left and maximum number of digits right of the decimal point for each word. In many cases they are not valid together since the control allows a maximum of 8 total digits. Refer to the system installer's manual for specific formats.

10.4.1 Minimum and Maximum Axis Motion (Programming Resolution)

The maximum programmable value accepted by the control is 99,999,999. The minimum is .000001 inch or .00001 mm. However, the actual range of programmable values depends on specifications determined by the system installer.

By using AMP to establish the format of numeric values for words, the system installer sets the “programming resolution,” for axis motion, that is, the smallest programmable distance of axis motion.

Table 10.C
Programming Resolutions

Formats as set in AMP	_.3	_.4	_.5	_.6
Corresponding Resolution	0.001	0.0001	0.00001	0.000001

Refer to the system installer's documentation for the programming resolutions and ranges in a specific system.

10.5 Word Descriptions

This section describes general features of the words used in programming. Later chapters in this manual describe, in detail, how to use these words.

10.5.1 A_ L_ ,R_ ,C_ (Quick Plus and Radius-Chamfer Words)

To simplify programming an angle, corner radius, or chamfer between two lines, all that is necessary is the angle between the lines and the radius or chamfer size connecting them. This method of programming can be used to simplify the cutting of many complex parts.

Quick path words are made up of the addresses below followed by the desired numeric value.

,A ... angle
L ... length
,R ... corner radius
,C ... chamfer size

Important: A comma “,” must precede the ,A ,R and ,C address characters for the control to recognize them as radius or chamfer words.

For more details and examples using QuickPath Plus see chapter 15. For more details and examples using radius and chamfer words see chapter 16.

10.5.2 Axis Names

Axis words are made up of an axis name followed by the desired numeric value for that word.

For axis names, the system installer chooses from:

A B C U V W X Y Z \$X \$Y \$Z \$B \$C

These are assigned in AMP. This manual assumes primary axes one, two, and three to be labeled X, Y, and Z respectively. Integrant words for these axes are assumed by be I, J, and K respectively.

10.5.3 D - and H- Words (Tool Offsets)

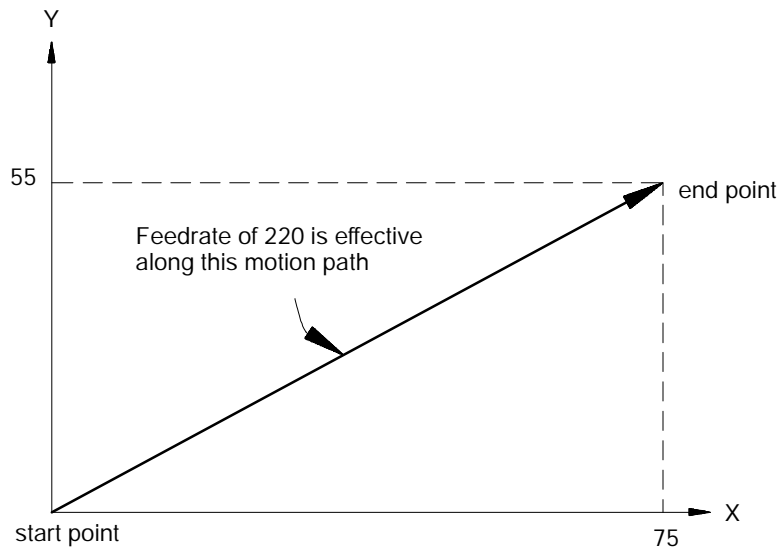
Specify a D address followed by a numeric value to call a value for the tool diameter from the offset tables discussed in chapter 3. The diameter called out from the table is used as the tool diameter when cutter compensation is active (G41, and G42). For details on compensation see chapter 21.

Specify an H address followed by a numeric value to call a value for the tool length offset from the offset tables discussed in chapter 3. The length called out from the table is used as the tool length offset when programming Z axes moves. For details on tool length offsets see chapter 20.

10.5.4
F- words (Feedrate)

An F-word with numeric values specifies feedrates for the cutting tool in linear interpolation (G01), and circular interpolation (G02/G03) modes. The feedrate is the speed along a vector of the commanded axes, as shown in the following figure.

Figure 10.5
Feedrate Vectors



The term “feed” refers to moving a tool at a specific velocity in a cutting path. “Feedrate” is the velocity programmed for the feed of a tool.

Feedrates are expressed by the distance of movement per interval. Depending on the mode of the control and the results wanted, the distance can be millimeters, inches, meters, or revolutions. The interval can be minutes or revolutions.

Table 10.D
Feedrate Units

Unit/Interval	Abbreviation	Typically Used For:
millimeters per minute	mmpm	linear axis feedrates
inches per minute	ipm	linear axis feedrates
revolutions per minute	rpm	rotary axis feedrates
millimeters per rev	mmpr	threading
inches per rev	ipr	threading

In a metric part program for a linear axis, a feedrate of 100 millimeters per minute (mmpm) typically would be written as F100.; (depending on the active word format).

For details on programming feedrates using the different feedrate modes, see chapter 18.

Important: Feedrates programmed in any of the feedrate modes (G93, G94, or G95) can be overridden by use of the switch.

10.5.5 G-codes (Preparatory Functions)

The preparatory function is designated by a G-code consisting of address G followed by a two-digit value. In some cases, the G-code may have an additional decimal digit. The G-codes are classified as modal and non-modal.

Modal	the G-code remains in effect until another G-code in the same G-code group is programmed.
Non-modal	the G-code is in effect only in the block in which it is programmed.

The system installer determines in AMP if M- and G-codes get reset every time the control executes an M02 or M30 end-of-program command. If the control does reset M- and G-codes, modal M- and G-codes default back to their power-up condition and non-modal M- and G-codes are reset to their default values. If M- and G-codes do not reset, modal M- and G-codes remain at their present value when the program is completed. A control reset is then necessary to reset the modal M- and G-codes.

Important: G-codes may also be expressed in terms of a parametric expression (for example G[#12+6]). For details, see section 28.2.

How the modal G-codes are executed is shown below, taking G00 and G01, both classified into the same G-code group.

Example 10.9
Modal G-code Execution

G00 X1. Y2. ;	G00 mode is effective
Y3. ;	G00 mode is in effect
G01 X2. Y1. F1;	G01 mode is made effective
X3. Y3. ;	G01 mode is in effect
G00 X1. Y2. ;	G00 mode becomes effective again
G01 G00 Y3, ;	G00 mode is in effect
G01 G91 Y2 ;	G01 and G91 both in effect

An example of non-modal G-code execution follows:

Example 10.10
Non-Modal G-code Execution

G00 X1. Y21 ;	G00 mode is effective
G28 X2. ;	G28 mode, this block only
X2. Y1. ;	G00 mode is effective
G04 P2. ;	G04 active this block only

Important: When programming more than one G-code from the same modal group in a block, the control will execute the block as the G-codes occur sequentially from left to right. Any mode that is being changed in a block only applies to the values to the right of the G-code. Characters to the left of the G-code still use the old operating mode.

Example 10.11
Changing Modes Mid-Block

N10G90X10G91Y10;	X10 is absolute (G90) Y10 is incremental (G91).
N20X20;	X20 is incremental (G91).
N30X30G90Y10;	X30 is incremental (G91), Y10 is absolute (G90).
N40M30	

Table 10.E breaks down the G-codes into their modal groups. For example, G-codes in group 01 are modal only with other G-codes in group 01. G-codes in the 00 group are non-modal and are effective only in the block in which they are programmed.

Table 10.E
G-codes

G-Code	Modal Group	Function	Type		
G00	01	Rapid Positioning	Modal		
G01		Linear Interpolation			
G02		Circular/Helical Interpolation (Clockwise)			
G03		Circular/Helical Interpolation (Counterclockwise)			
G04	00	Dwell	Non-Modal		
G05		Send Command and Wait for Return Status (for 9/Series Data Highway Communication Module)			
G05.1-G05.4		Send Command without Waiting for Return Status (for 9/Series Data Highway Communication Module)			
G09		Exact Stop			
G10L2		Setup Work Coordinate Offset Tables			
G10L3		Setup Tool Management Table			
G10L10		Setup Tool Length Values Geometry Table			
G10L11		Setup Tool Length Values Wear Table			
G10L12		Setup Tool Radius Values Geometry Table			
G10L13		Setup Tool Radius Wear Values Wear Table			
G10.1		Setup Random Tool Table			
G10.2L1		Communication Configuration Information (for 9/Series Data Highway Communication Module)			
G10.2L2		Output Command Table (for 9/Series Data Highway Communication Module)			
G10.2L3		Download Configuration Information (for 9/Series Data Highway Communication Module)			
G11		Setup Tool Management Table (Cancel)			
G12.1		21		Primary Spindle Controlling	Modal
G12.2				Auxiliary Spindle 2 Controlling	
G12.3				Auxiliary Spindle 3 Controlling	
G13				QuickPath Plus (Use First Intersect.)	
G13.1	QuickPath Plus (Use Second Intersect.)				
G14	19	Scaling (Disable)	Modal		
G14.1		Scaling (Enable)			
G15	15	Polar Coordinate Programming (Cancel)	Modal		
G16		Polar Coordinate Programming			
G16.1		Cylindrical Interpolation			
G17	02	Plane Selection	Modal		
G18		Plane Selection			
G19		Plane Selection			
G20	06	Inch System Selection	Modal		
G21		Metric System Selection			

G-Code	Modal Group	Function	Type	
G22	04	Programmable Zone 2 and 3, ON	Modal	
G22.1		Programmable Zone 3, ON		
G23		Programmable Zone 2 and 3, OFF		
G23.1		Programmable Zone 3, OFF		
G24	00	Feed to Hard Stop	Non-Modal	
G25		Adaptive Feedrate (torque mode)		
G26		Adaptive Depth		
G27		Machine Home Return Check		Non-Modal
G28		Automatic Machine Home		
G29		Automatic Return From Machine Home		
G30		Return to Secondary Home		
G31		External Skip Function 1		
G31.1		External Skip Function 1		
G31.2		External Skip Function 2		
G31.3		External Skip Function 3		
G31.4		External Skip Function 4		
G36		22	Short Block Acc/Dec (Disable)	
G36.1			Short Block Acc/Dec (Enable)	
G37	00	Tool Gauging Skip, Function 1	Non-Modal	
G37.1		Tool Gauging Skip, Function 1		
G37.2		Tool Gauging Skip, Function 2		
G37.3		Tool Gauging Skip, Function 3		
G37.4		Tool Gauging Skip, Function 4		
G38		Circle Diameter and Center Measurement		
G38.1		Parallel Probing Cycle		
G39		20		Cutter Diameter Comp (Linear Generated Block)
G39.1	Cutter Diameter Comp (Circular Generated Block)			
G40	07	Cutter Diameter Compensation (Cancel)		
G41		Cutter Diameter Compensation (Left)		
G42		Cutter Diameter Compensation (Right)		
G43	08	Tool Length Offset (Plus)	Modal	
G43.1		Tool Length Offset Selection (Plus)		
G44		Tool Length Offset (Minus)		
G44.1		Tool Length Offset Selection (Minus)		
G45	23	Disable Spindle Synchronization	Modal	
G46		Set Spindle Positional Synchronization		
G46.1		Set Active Spindle Speed Synchronization		
G47	24	Linear Acc/Dec in All Modes	Modal	
G47.1		S-Curve Acc/Dec for Positioning and Exact Stop Mode		
G47.9		Infinite Acc/Dec (No Acc/Dec) (Amp-selectable only)		

G-Code	Modal Group	Function	Type	
G48	00	Reset Acc/Dec to Default AMPed Values	Non-Modal	
G48.1		Acceleration Ramp for Linear Acc/Dec Mode		
G48.2		Deceleration Ramp for Linear Acc/Dec Mode		
G48.3		Acceleration Ramp for S-Curve Acc/Dec Mode		
G48.4		Deceleration Ramp for S-Curve Acc/Dec Mode		
G48.5		Programmable Jerk Value		
G49	08	Tool Length Offset Cancel)	Modal	
G50.1	11	Programmable Mirror Image (Cancel)	Modal	
G51.1		Programmable Mirror Image		
G52	00	Offsetting Coordinate Zero Point	Non-Modal	
G53		Motion in Machine Coordinate System		
G54	12	Preset Work Coordinate System 1	Modal	
G55		Preset Work Coordinate System 2		
G56		Preset Work Coordinate System 3		
G57		Preset Work Coordinate System 4		
G58		Preset Work Coordinate System 5		
G59		Preset Work Coordinate System 6		
G59.1		Preset Work Coordinate System 7		
G59.2		Preset Work Coordinate System 8		
G59.3		Preset Work Coordinate System 9		
G61		13		Exact Stop Mode
G62	Automatic Corner Override			
G63	Tapping Mode			
G64	Cutting Mode			
G65	00	Paramacro Call	Non-Modal	
G66	14	Paramacro Modal Call	Modal	
G66.1		Paramacro Modal Call		
G67		Paramacro Modal Call (Cancel)		
G68	16	Part Rotation	Modal	
G69		Part Rotation (Cancel)		
G73	09	Deep Hole Peck Drilling Cycle (With dwell)	Modal	
G74		Left-Hand Tapping Cycle		
G74.1		Left-Hand Solid Tapping Cycle		
G76		Boring Cycle (Spindle Shift)		
G80		Cancel or End Fixed Cycle		
G81		Drilling Cycle (No Dwell, Rapid Out)		Modal
G82		Drilling Cycle (Dwell, Rapid Out)		
G83		Deep Hole Peck Drilling Cycle		
G84		Right-Hand Tapping Cycle		
G84.1		Right-Hand Solid Tapping Cycle		
G85		Boring Cycle (No Dwell, Feed Out)		

G-Code	Modal Group	Function	Type
G86		Boring Cycle (Spindle Stop, Rapid Out)	
G87		Back Boring Cycle	
G88		Boring Cycle (Spindle Stop, Manual Out)	
G88.1	00	Pocket Milling Roughing Cycle	Non-Modal
G88.2		Pocket Milling Finishing Cycle	
G88.3		Pocket Milling Roughing Cycle	
G88.4		Pocket Milling Finishing Cycle	
G88.5		Hemispherical Milling (Roughing Cycle)	
G88.6		Hemispherical Milling (Finishing Cycle)	
G89	09	Boring Cycle (With Dwell, Feed Out)	Modal
G89.1	00	Irregular Pocket Milling (Roughing Cycle) ¹	Non-Modal
G89.2		Irregular Pocket Milling (Finishing Cycle) ¹	
G90	03	Absolute Mode	Modal
G91		Incremental Mode	
G92	00	Coordinate System Offset (Using Tool Positions)	Non-Modal
G92.1		Coordinate System Offset (Cancel)	
G92.2		Selected Coordinate System Offsets (Cancel)	
G93	05	Inverse Time Feed Mode	Modal
G94		Feed-per-minute mode	
G95		Feed-per-revolution Mode	
G98	10	Initial Level Return in Milling Cycles	Modal
G99		R-Point Level Return in Milling Cycles	

¹Important: The Irregular Pocket Milling Cycles feature (G89.1 and G89.2) is only available prior to system software release 12.xx. Any attempt to program a G89.1 or G89.2 in release 12.xx or later will result in the error message, "Illegal G-code".

A set of default G-codes become effective at power up, when the control is reset, or an emergency stop condition is reset. These default G-codes are selected by the system installer in AMP. These default G-codes can be seen on the status display screen after power up or control reset.

10.5.6 I, J, and K Integrand Words

Integrand words are typically used to define parameters that relate to a specific axis for a canned cycle, probing cycle, or circular motion block; though not limited to use only in these operations. For example, in circular motion blocks the axis integrands are used to define the center point of the arc being cut.

The system installer has the option of assigning either I, J, K, H or none as the axis integrand name for a specific axis. This manual makes the following assumption:

I	integrand name for the X axis
J	integrand name for the Y axis
K	integrand name for the Z axis
none	no integrand selected for the U axis.

Important: Refer to the system installers documentation to make sure the assumptions are true. If this assumption is not true it is important that the programmer realize that all examples and formats in this manual that use an I, J, or K need to have their letter replaced with the system installer's integrand words accordingly.

10.5.7 M - codes(Miscellaneous Functions)

The miscellaneous function is designated with an address M followed by a two or three-digit numeric value. Because many of these are set by industry standards, they are usually referred to as M-codes.

When a miscellaneous function is designated in a block containing axis motion commands, the control's PAL program determines whether the M-codes will be:

executed at the same time as the axis motion

executed before the axis motion

executed after the axis motion is completed

This order of execution may also be altered using the paramacro feature, system parameter #3003.

The system installer determines in AMP if M- and G-codes get reset every time the control executes an M02 or M30 end of program command. If the control does reset M- and G-codes, modal M- and G-codes default back to their power up condition and non-modal M- and G-codes are reset to their default values. If M- and G-codes do not reset, all modal and non-modal M- and G-codes remain at their present values.

The basic M-codes for the control are shown in Table 10.F. A part program block may contain as many basic M-codes as desired. If more than one M-code from any modal group is programmed in the same block, the rightmost M-code in that block for that modal group is the active M-code for the block.

The system installer may have defined additional M-codes in PAL. Up to four of these PAL M-codes may be activated in any one block. If more than four PAL M-codes are programmed in any one block the right most four in that block are activated. Other PAL M-codes in the block are ignored. Refer to documentation provided by the system installer for details on PAL defined M-codes and their operation. You may program as one basic control M-code from each modal group with up to four PAL M-codes in any one block.

Table 10.F
M-codes

M-code Number	Modal or Non-modal	Group Number	Function
M00	NM	4	Program stop
M01	NM	4	Optional program stop
M02	NM	4	Program end
M06	NM	4	Tool change
M30	NM	4	Program end and reset (tape rewind)
PRIMARY SPINDLE			
M03	M	7	Spindle positive rotation (cw)
M04	M	7	Spindle negative rotation (ccw)
M05	M	7	Spindle stop
M19	M	7	Spindle orient
AUXILIARY SPINDLE 2			
M03.2	M	11	Spindle positive rotation (cw)
M04.2	M	11	Spindle negative rotation (ccw)
M05.2	M	11	Spindle stop
M19.2	M	11	Spindle orient
AUXILIARY SPINDLE 3			
M03.3	M	12	Spindle positive rotation (cw)
M04.3	M	12	Spindle negative rotation (ccw)
M05.3	M	12	Spindle stop
M19.3	M	12	Spindle orient
M07	M	8	Mist coolant on
M08	M	8	Flood coolant on
M09	M	8	Coolant off
M48	M	9	Overrides enabled
M49	M	9	Overrides disabled
M58	M	10	Reserved for a lathe CSS feature. Can be used by a mill for any function.
M59	M	10	Reserved for a lathe CSS feature. Can be used by a mill for any function.
M98	NM	5	Sub-program call
M99	NM	5	Sub-program end and program jump

The following is a description of some of the basic M-codes provided with the control.

(Program Stop (M00))

When M00 is executed, program execution is stopped after the block containing the M00 is completed. At this time, the CRT displays the "PROG STOP" message. To restart the operation, press the {CYCLE START} button.

Optional Program Stop (M01)

The optional program stop function has the same effect as the program stop function, except that it is controlled by an external switch. That is, when the OPTIONAL PROGRAM STOP switch is placed in the OFF position, the M01 code in the program is ignored. This switch and the appropriate PAL programming are the responsibility of the system installer.

End of Program (M02)

If executing a program from control memory the M02 code acts the same as an M30, program execution is stopped and the control enters the cycle stop state. The program is reset to the first block and a <CYCLE START> will begin part program execution over again (see M99 for auto cycle start).

If executing a program from an external device (such as a tape reader), when M02 is executed, program execution is stopped and the control enters the cycle stop state. The M02 does not cause a tape rewind. The tape reader must be rewound using some other method before program execution can resume.

With some machines, the M02 code may also result in a spindle and coolant supply stop. For details, refer to the instruction manual prepared by the system installer.

Tool Change (M06)

An M06 typically is used to request a tool changer to perform some tool changing operation. Motion commands can not be programmed in an M06 block. Offset activation and tool change operation are dependent on your system installers PAL program and AMP configuration. More details on the tool change operation (M06) can be found on page 20-2.

End of Program, Tape Rewind (M30)

If executing a program from control memory the M30 code acts the same as an M02, program execution is stopped and the control enters the cycle stop state. The program is reset to the first block and a <CYCLE START> will begin part program execution over again (see M99 for auto cycle start).

If executing a program from an external device (such as a tape reader), when M30 is executed, program execution is stopped and the tape is automatically rewound to the beginning of the program just executed. After the tape is rewound the control enters the cycle stop state. A <CYCLE START> will begin part program execution over again (see M99 for auto cycle start).

With some machines, the M30 code may also result in a spindle and coolant supply stop. For details, refer to the instruction manual prepared by the system installer.

Overrides Enabled (M48)

When M48 is executed, the feedrate override, rapid feedrate override, and the spindle speed override functions become effective. Normally these will be enabled on power up without requiring this M-code to be executed. An M48 cancels an M49 and the system installer can choose which will be active upon power-up.

Overrides Disabled (M49)

Use the override cancel M-code (M49) to ignore any override set by the operator on the MTB panel. When the override setting is ignored, the axis feedrate, rapid feedrate, and the spindle speed override values are all set to 100 percent. An M49 cancels an M48 and the system installer can choose which will be active upon power-up. This override setting is ignored if you are using programmed motion.

Subprogram call (M98)

When M98 is executed, a subprogram is called and executed. This word may be used in any program including an MDI program. For details on programming an M98, refer to section 10.3.1.

End of Subprogram or Main Program Auto Start (M99)

M99 End of Subprogram or Paramacro program

When M99 is executed, subprogram execution is completed and program execution returns to the calling program. This word is not valid in an MDI command though it may be contained in a subprogram called by an MDI command. For details on programming an M99, refer to section 10.3.

M99 End of Main Program with Auto Start

If executing a program from memory, an M99 as the last block in a main program causes program execution to stop at that location. The program is reset to the first block and a <CYCLE START> automatically starts program execution for you.

If executing a program from an external device (such as a tape reader), when M99 is executed, program execution is stopped and the tape is automatically rewound to the beginning of the program just executed and a <CYCLE START> automatically starts program execution for you.



CAUTION: The M99 code is commonly used as the end of program for fully automated systems that automatically load the next part to be machined. Typically this code requires that some PAL interface be written that assures the part is fully loaded and ready for machining before block execution is allowed to restart. Failure to do so may cause injury to operators or damage to equipment.

For these systems some PAL interface should be written to assure that the part is fully loaded before program execution is restarted.

Simple Synchronization (M100-M149)

M100 - M149 – Simple Synchronization (dual-process system only)
These M-codes are for simple synchronization. When executed, this set of M-codes does not re-setup any program blocks that have already been read into program lookahead. See page 30-7.

Synchronization with Setup (M150-M199)

M150 - M199 – Synchronization with Setup (dual-process system only)

This set of M-codes cancels any information already in block look ahead and re-setup the blocks before process execution is resumed. This re-setup is only essential when shared information is being changed from one process to another, as in the case of the dual processing paramacro parameters. See page 30-7.

Important: You cannot use these M-codes when TTRC is active.

Other more specific M-codes are described in later sections that deal specifically with their functions.

Important: When you activate the MISCELLANEOUS FUNCTION LOCK feature, the control displays M-, B-, S-, and T-words in the part program with the exception of M00, M01, M02, M30, M98, M99 and M100-M150. This feature is described in chapter 7.

The operation of other M-codes are given in detail in sections that pertaining to their specific usage. Others, such as coolant and spindle control should be in documentation prepared by the system installer.

10.5.7.1
Auxiliary Miscellaneous
Function (B- word)

The B-word is commonly used when the number of M-codes is not sufficient for the available number of miscellaneous functions. Any alphabetic character which is not used for other functions may be used instead of B by setting the proper AMP parameter. For details, refer to the documentation prepared by your system installer, or your AMP reference manual. This manual assumes the B-word is used to call second auxiliary functions.

The B-word is designated by a two or three-digit numeric value following address B. Unlike M-codes, each block can contain only one B-word.

Important: When the MISCELLANEOUS FUNCTION LOCK feature is activated, the control displays M-, B-, S-, and T-words in the part program with the exception of M00, M01, M02, M30, M98, and M99. This feature is described in chapter 7.

10.5.8
N- words (Sequence
Numbers)

Each block in a part program can be assigned up to a five-digit numeric value following an N-address. These numbers are referred to as sequence numbers and are used to distinguish one block from another.

Sequence numbers can be assigned at random to specific blocks or to all blocks if desired. Blocks assigned sequence numbers can be called later by referencing their sequence number. Sequence numbers are necessary to make program jumps and to specify a block for subprogram calls and returns.

10.5.9
O- words (Program Names)

The O-word is used to define a program name. To use an O-word as a program name it must be the first block entered in a program. This block can be used to identify a program when reading from a tape (when program name is selected as "automatic" from the device setup menu). This is useful when many programs are placed together on a single tape. An O-word can have up to five numeric characters following it.

10.5.10
P ,L (Main Program Jumps
and Subprogram Calls)

When the same series of blocks are repeated more than once it is usually easier to program them using a subprogram.

Important: To make jumps, loops, or calculations within an executing program or subprogram use any of the paramacro features as discussed in chapter 28.

P-words in a subprogram call (M98) are used to designate the specific program being called. The P-address is followed by the program name being called.

L-words in a subprogram call (M98) are used to designate a repeat count for a subprogram. The number following the L-address designates the number of times a subprogram will be executed consecutively before execution is returned to the main program.

10.5.11 S-word (Spindle Speed)

Program spindle speeds (in RPM) using an S-word with up to five integer digits and three decimal digits. The actual legal format is defined in AMP by the system installer. A common S-word is used to program all of the spindles AMPed to be in the system.

The system installer sets a maximum spindle speed in AMP for each different gear range for each spindle configured in AMP. If an S-word is programmed requesting a spindle speed that exceeds this limit, the spindle speed will be held at the AMP defined maximum unless a gear change operation is performed.

When programming an S-word in a block that contains axis motion commands, the PAL program has the option to temporarily suspend the axis motion commands until the spindle reaches speed. The control has the ability to take the programmed spindle speed and automatically search for the gear range (configured in AMP) to allow the necessary RPM. The operation of gear changing and how it is implemented is very PAL dependant. Refer to the system installers documentation for details on how a gear change operation is performed.

Override programmed spindle speeds with the {SPINDLE SPEED OVERRIDE} switch on the MTB panel. This switch can be positioned in five percent increments within a range of 50 to 120 percent. For details, refer to documentation prepared by your system installer.

Use the override cancel M-code (M49) to ignore any override set on the MTB panel. When the override setting is ignored, the axis feedrate, rapid feedrate, and the spindle speed values are all set to 100 percent.

Important: When the MISCELLANEOUS FUNCTION LOCK feature is activated, the control displays M-, B-, S-, and T-words in the part program with the exception of M00, M01, M02, M30, M98, and M99. This feature may be activated through the front panel screen as described in chapter 7.

Cutting Speed

The term “cutting speed” refers to the velocity of the surface of the revolving cutting tool relative to the workpiece. Cutting speeds are determined by the spindle speed in revolutions per minute (rpm) and the diameter of the cutting tool in the following equation:

Metric Units	English Units
$V = \frac{3.14159 \times D \times N}{1000}$	$V = \frac{3.14159 \times D \times N}{12}$

Where :	Is :
V	cutting speed in meters or feet per minute
D	diameter of the tool in millimeters or inches
N	rpm of the spindle

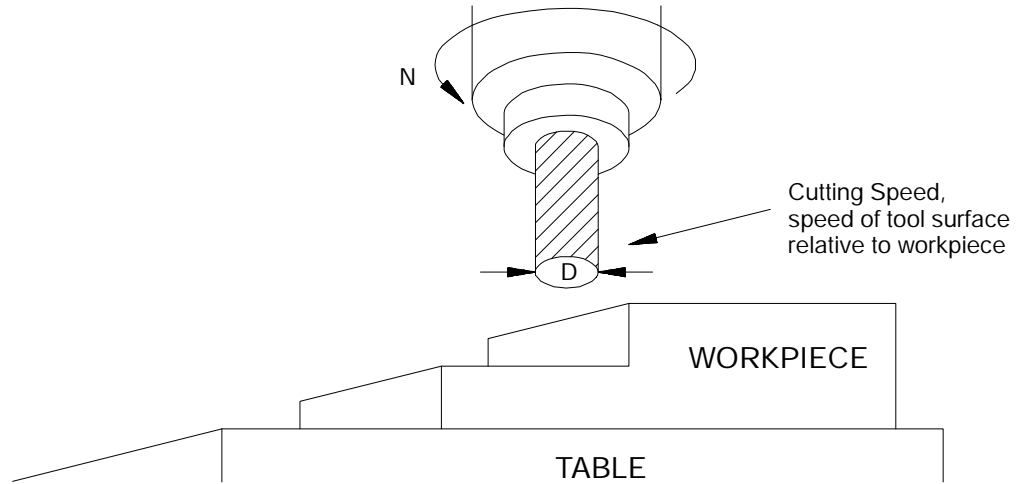
Or, stating the equations in terms of required spindle rpm:

Metric Units	English Units
$N = 318.30989 \frac{V}{D}$	$N = 3.81972 \frac{V}{D}$

From the metric equation, if a desired cutting speed is known, for example $V = 100$ surface meters per minute using a cutting tool with a diameter, $D = 100$ millimeters, the spindle speed, N is equal to approximately 318 rpm.

Spindle speed is programmed with an S-word. In this above example the S-word would be S318.

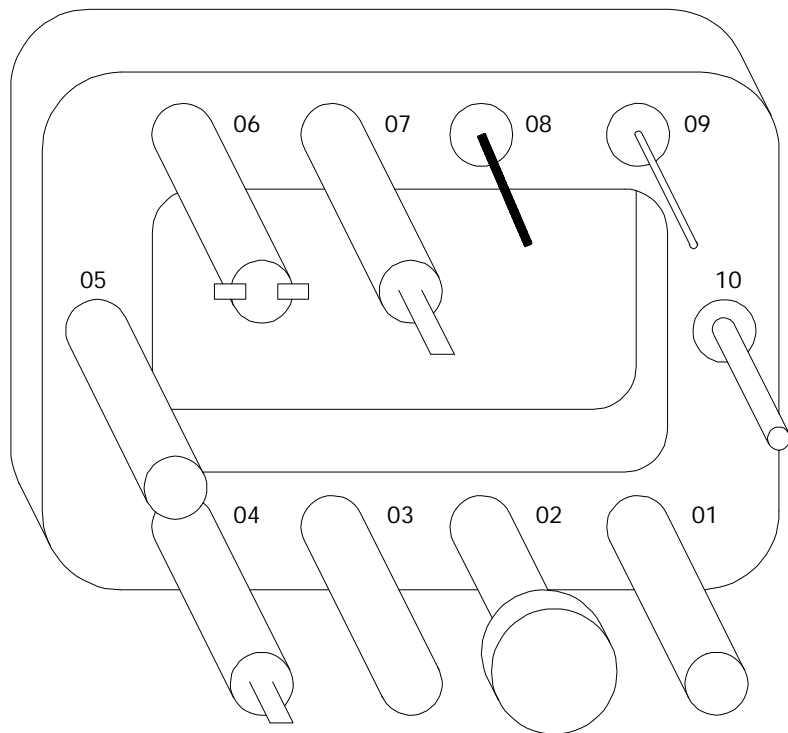
Figure 10.6
Cutting Speed



10.5.12
T- words (Tool Selection)

A workpiece usually requires different kinds of cutting processes, and usually there are cutting tools that correspond to each process. The cutting tools are typically stored in a tool magazine and are assigned tool numbers (see Figure 10.7).

Figure 10.7
Tool Magazine



A T-address followed by a numeric value programs a tool selection. When the control executes the T-word, it outputs a tool selection signal to a tool changer. The tool changer should perform a sequence of operations to deliver the proper tool in response to the tool selection signal. For example, to select a cutting tool that is assigned tool number "03", write "T03" in the part program. The system installer may require a M06 in the program to cause a tool change.

Since tool changers vary in style, size and function, the system installer is responsible for specific implementations through PAL. Refer to your PAL reference manual and the manual supplied by the system installer for more details.

Important: When changing cutting tools it is usually necessary to change the tool offset at the same time. This is done with an H- or a D-word. For details see section 10.5.3

Important: When the MISCELLANEOUS FUNCTION LOCK feature is activated, the control displays M-, B-, S-, and T-words in the part program with the exception of M00, M01, M02, M30, M98, and M99. This feature is described in chapter 7.

END OF CHAPTER

Coordinate Systems Offsets

11.0 Chapter Overview

This chapter covers the control of the coordinate systems. G-words in this chapter will be among the first programmed because they define the coordinate systems of the machine in which axis motion is programmed in. This chapter describes:

Information about:	On page:
Machine (absolute) coordinate system	11-1
Preset work coordinate systems; G54 - G59.3	11-4
External offsets	11-9
Offsetting the work coordinate systems	11-13
PAL Offsets	11-22

11.1 Machine Coordinate System (Absolute)

The control has two types of coordinate systems. One is the work coordinate system, which is based on the coordinate system in the part drawing for a part to be cut by the machine. The other is the machine coordinate system (often referred to as the Absolute Coordinator System), which is unique to the individual machine tool. Programs are usually written in the work coordinate system.

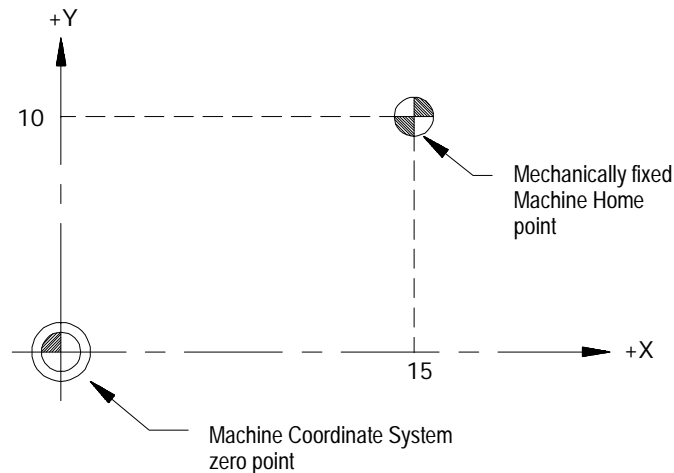
The machine coordinate system is the basic coordinate system set for every machine. It is established after completion of the machine homing operation. It can not be offset or shifted in any way. Its position is determined in AMP by the system installer.

Important: Before any coordinate system can be activated, the machine must be homed. The homing operation refers to the positioning of the axes to a machine-dependent fixed position, which is called the machine home. For more on machine homing refer to section 4.3.

The zero point of the machine coordinate system is referenced from the machine home point. This is done by assigning a coordinate location to the machine home point. Though frequently defined as X0Y0Z0, the home position for each axis can be given any legal coordinates such as 15.00, -20.0000, or -2.256.

Once established, the machine coordinate system is not affected by a control reset operation or any other programming or operator operation.

Figure 11.1
Machine Coordinate System, Home Coordinate Assignment



In Figure 11.1 the system installer has defined the zero point of the machine coordinate system by assigning the machine home point to have the coordinates $Y=10$ and $X=15$ in the machine coordinate system.

Note that the coordinate values assigned to the machine home point do not affect the position of machine home. The position of machine home is fixed by the system installer.

Important: Normally the control displays the current position of the axis in respect to an active work coordinate system. The position in the machine coordinate system may be displayed by selecting the Absolute screen as discussed in Chapter 8.

11.1.1 Motion in the Machine Coordinate System (G53)

Although axis motion is usually commanded in the work coordinate system, axis motion by referencing coordinate values in the machine coordinate system is possible when a G53 is programmed in a block.

```
G90G53X__Y__Z__ ;
```

The X, Y and Z words above specify coordinate positions in the machine coordinate system. These coordinate values indicate the end point of the next move in the machine coordinate system. The tool travels to this position in either G00 or G01 mode depending on which is active when the G53 block is executed. Any attempt to execute a G53 block in G02 or G03 mode will generate an error.

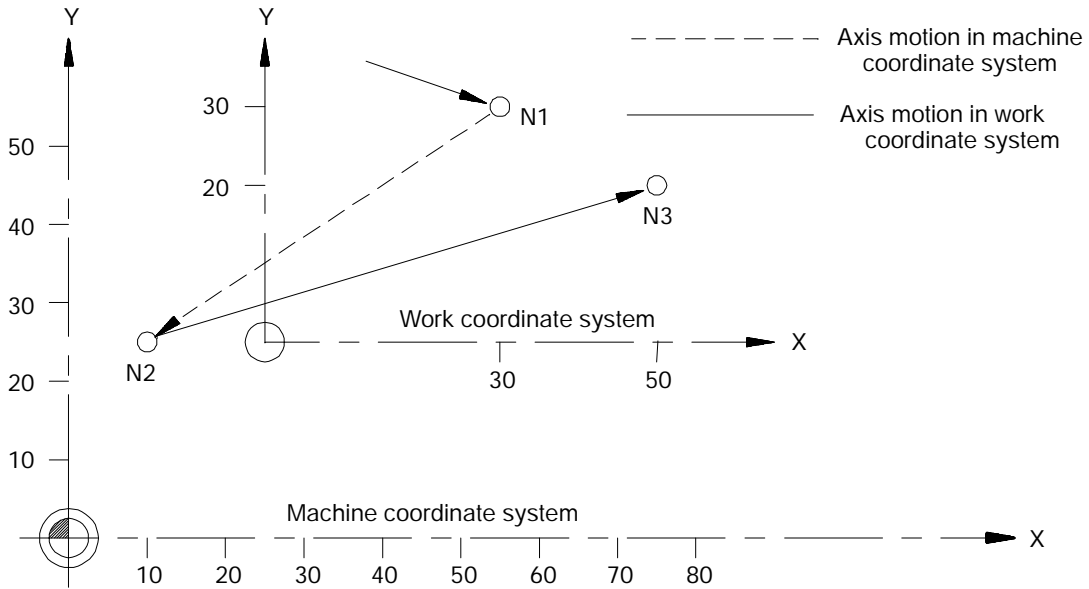
The G53 code is not modal and is effective only in the block in which it is called. After a G53 block the control returns to the positioning mode that was in effect prior to the G53 block's execution.

Important: The control must be in absolute mode (G90) when the G53 command is executed. If a G53 is executed while in incremental mode (G91), the G53 code and any axis words in the G53 block will be ignored by the control.

Example 11.1
Motion in the Machine Coordinate System

Program Block	Comment
N1G54G00X30. Y30. ;	axis motion in work coordinate system.
N2 G53X10. Y25. ;	axis motion in machine coordinate system.
N3 X50. Y20. ;	axis motion in work coordinate system.

Figure 11.2
Results of Example 11.1

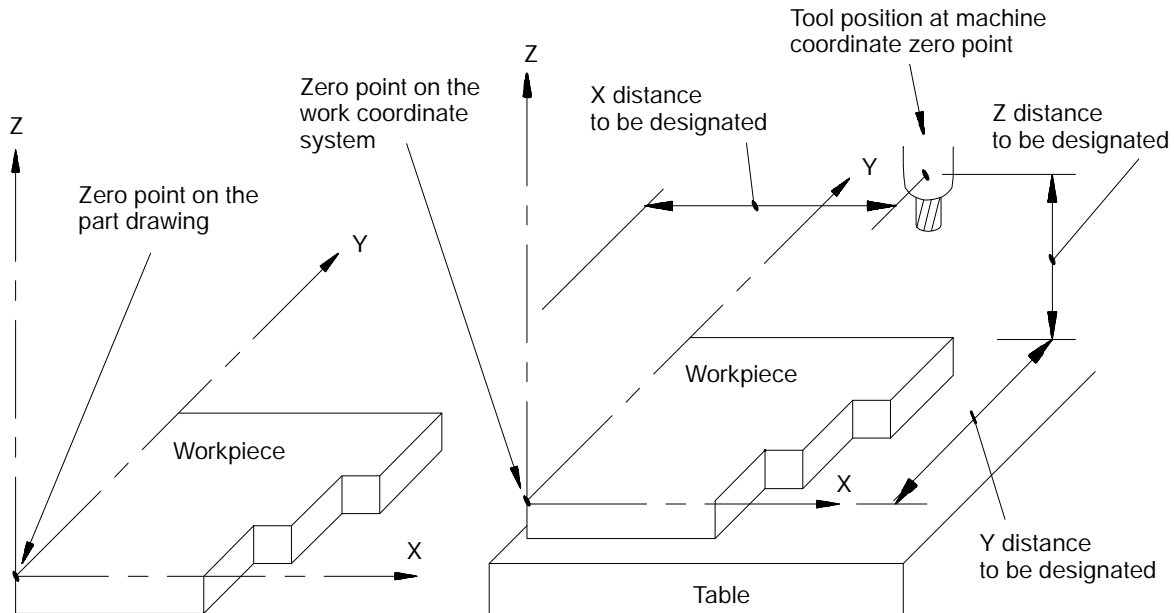


11.2 Preset Work Coordinate Systems (G54-59.3)

When cutting a workpiece using a part program made from a part drawing, it is desirable to match the zero point on the coordinate system of the part drawing with the zero point of the work coordinate system.

As shown in the illustrations in Figure 11.3, the work coordinate system is established by programming the distance between the desired zero point of the work coordinate system and the zero point of the machine coordinate system.

Figure 11.3
Work Coordinate System

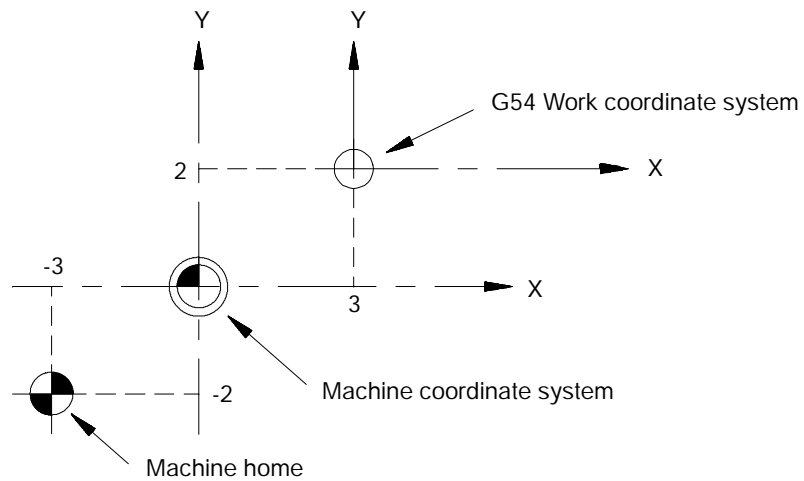


There are 9 preset work coordinate systems selectable using G54 - G59.3. The required work coordinate system can be selected by specifying any of these G-codes in the program.

Work coordinate systems called out by G54 - G59.3 have zero points that are entered in a work coordinate system table (see section 3.3). These zero points are in the form of offset values from the machine coordinate system zero point.

The machine coordinate system is established by the control immediately after the machine home operation is completed. The default work coordinate system, determined in AMP by the system installer, is activated simultaneously. The default work coordinate system is established upon execution of a control reset operation, E-STOP, G92.1, or power up. The default work coordinate system is the sum of the external offset value (if any), and the offsets of the default coordinate system selected in AMP (G54-G59.3 or none). If the default coordinate system is selected as none, the default work coordinate system will be simply the external offset (if any). This manual assumes G54 to be the default coordinate system and no external offset has been centered.

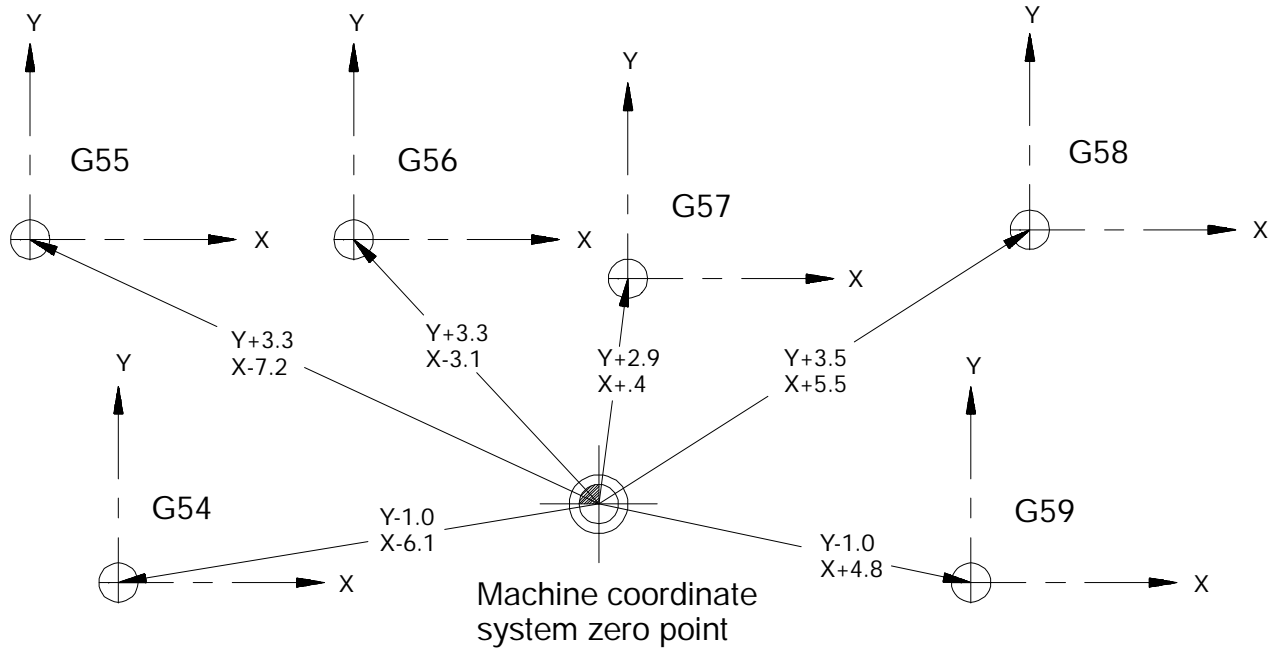
Figure 11.4
Work Coordinate System Definition



In the above figure the machine coordinate system was defined by declaring the fixed position machine home as the point $X=-3$, $Y=-2$. Then the G54 work coordinate system zero point is defined by the coordinates $X=3$, $Y=2$ in the machine coordinate system.

Coordinate positions in a part program are manipulated as coordinate values in the G54 default work coordinate system unless another coordinate system is selected by programming G54-G59.3.

Figure 11.5
Examples of Work Coordinate System Definition

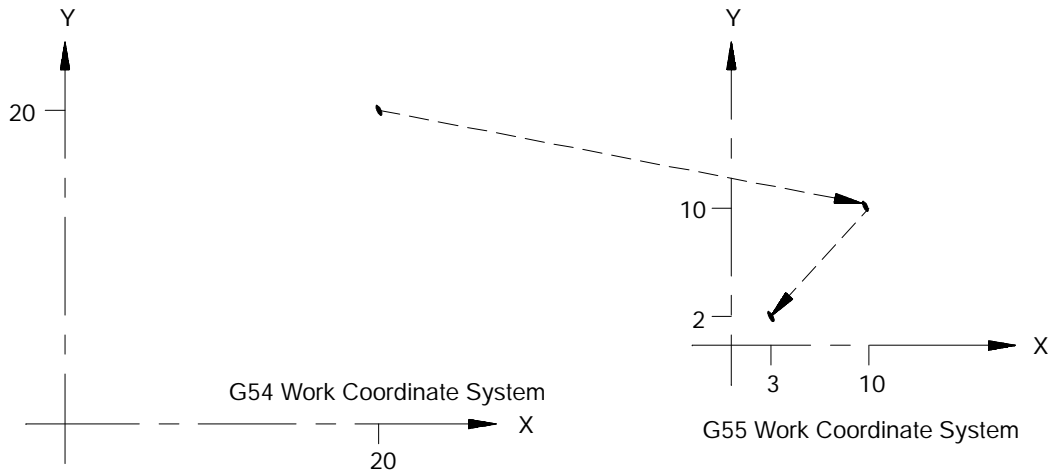


To change work coordinate systems simply specify the G-code corresponding to the desired work coordinate system in a program block. Any axis motion commands in a block that contains a change from one work coordinate system to another are executed in the work coordinate system specified in that block.

Example 11.2
Changing Work Coordinate Systems

Program Block	Comment
G54;	
G00X20. Y20. ;	axis motion in the G54 work coordinate system.
G55X10. Y10. ;	axis motion to the point X10. Y10. in the G55 work coordinate system.
X3. Y2. ;	axis Motion in the G55 work coordinate system.

Figure 11.6
Results of Example 11.2



11.2.1 Altering Work Coordinate Systems (G10L2)

There are 4 methods to change the value of a work coordinate system zero point in the work coordinate system table. Three methods can be found in the following sections:

- Manually alter the work coordinate system table as described in section 3.3.
- Alter the paramacro system parameter values 5221- 5386 as discussed in chapter 28.
- Alter through some external means defined by the system installers PAL program (refer to your PAL reference manual).

The fourth method, and the one discussed in this section, alters the work coordinate system table through G10 programming. Changing the values in the table using any of these methods does not cause axis motion; however, it does immediately shift the active coordinate system by the amount entered. The format for altering the work coordinate systems using G10 is as follows:

```
G10L2P__X__Y__Z__;
```

Important: The order of the words in this program block is important. The L and P words must be programmed before any axis words are programmed in the G10 block. Failing to follow this order can result in data being misinterpreted and loaded into the table incorrectly.

Where :	Is :
L2	tells the control that you want to alter the coordinate system tables.
P	specifies which coordinate system (G54 through G59.3) you want to work on. P1 through P9 correspond to the work coordinate systems G54 through G59.3. P1 = G54 work coord. system P6 = G59 work coord. system P2 = G55 work coord. system P7 = G59.1 work coord. system P3 = G56 work coord. system P8 = G59.2 work coord. system P4 = G57 work coord. system P9 = G59.3 work coord. system P5 = G58 work coord. system
X_Y_Z_	specify the location of the zero point of the specified work coordinate system relative to machine coordinate system.

Important: G10 blocks may not be programmed when TTRC is active.

Incremental/Absolute Mode and the G10L2 Command

When you program in incremental mode (G91), any values entered into the work coordinate system table using the G10 command are added to the currently active work coordinate system values. When you program in absolute mode (G90), any values entered into the work coordinate system table using the G10 command replace the currently active work coordinate system values.

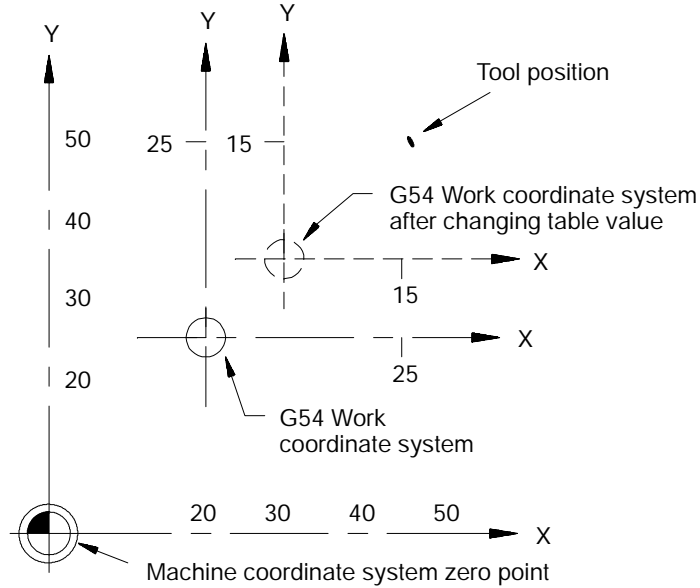
Example 11.3 and Figure 11.7 illustrate how the work coordinate system is shifted using G10.

Example 11.3
Work Coordinate System Shift Using G10

Program block	Work Coordinate Position	Absolute Coordinate Position
G54X25. Y25. ;	X25 Y25	X50 Y45
G91; G10L2P1X10. Y10. ;	X15 Y15	X50 Y45
G90; G10L2P1X30. Y35. ;	X15 Y15	X50 Y45

Important: This modification is permanent. The new table values for the work coordinate systems are saved even when the control power is turned off.

Figure 11.7
Results of Example 11.3



11.3 Work Coordinate System External Offset

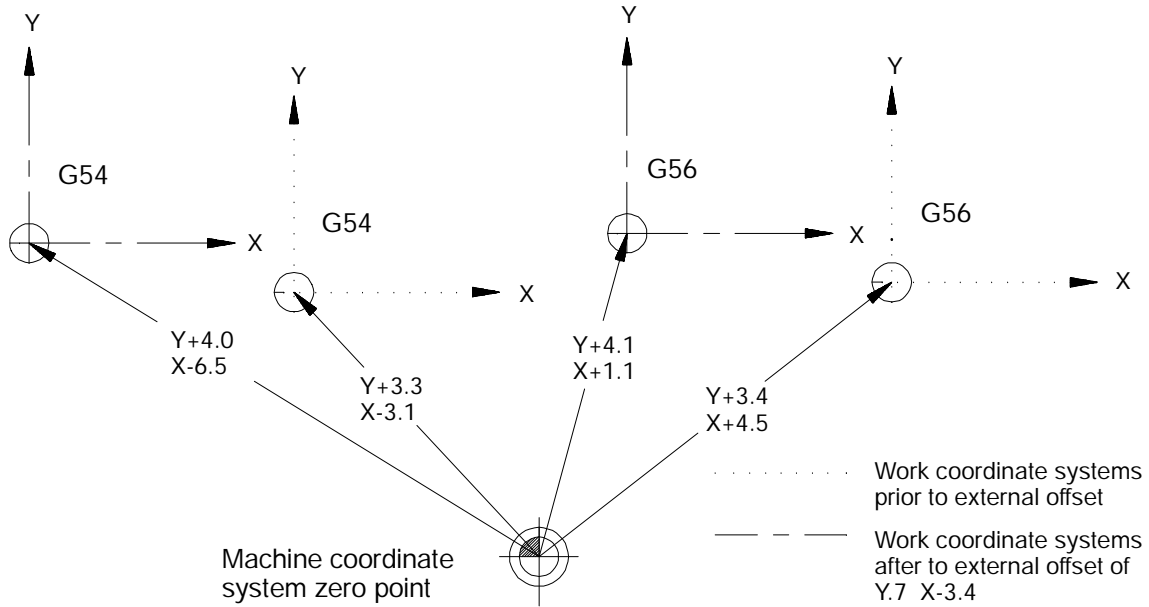
The external offset allows all work coordinate system zero points to be shifted simultaneously, relative to the machine coordinate system. This offset can compensate for part positioning shifts that result when a different tool is installed. It can also compensate for tool position shifts that result from a different tool fixture.

The external offset can also be used to match the work coordinate systems on mechanically different machines. The machines can then use the same part program with the same G54-G59.3 coordinate values. This allows part programs to be less machine dependant.

Four methods are available to modify the external offset:

Method:	Active Immediately after block execution	Active next Cycle Stop	Active next re-home or power cycle
Programming a G10	X		
System installer PAL offset		X	
Manually through table		X	
Paramacro Programming			X

Figure 11.8
External Offsets



Important: Once an external offset is entered into the coordinate offset table it cannot be canceled. This offset remains active even after power has been turned off. It becomes a permanent part of all work coordinate systems including the default work coordinate system.

11.3.1
Altering External Offset
(G10L2)

There are 4 methods used to change the value of an external offset in the work coordinate system table. Three methods can be found in the following sections:

- Manually alter the external offset value in the work coordinate system table as described in section 3.3.1.
- Alter the paramacro system parameter values 5201 - 5206 as discussed in chapter 28.
- Alter through some external means defined by the system installer's PAL program (refer to your PAL reference manual).

The fourth method, and the one discussed in this section, alters the external system table through G10 programming. Changing these values in the table using any of these methods does not cause axis motion; however, it does immediately shift the active coordinate system by the amount entered.

The values entered into the external offset are added to the work coordinate system zero point values each time a work coordinate system is called. The format for altering the external offset using G10 is as follows:

G10L2P0X__Y__Z__;

Where :	Is :
L2	tells the control that you want to alter the coordinate system tables.
P0	designates the external offset as the offset to update.
X_Y_Z_	specifies the location of the zero point of the specified work coordinate system relative to machine coordinate system.

When this block is executed, the control will immediately shift the currently active work coordinate system by the new external offset amount.

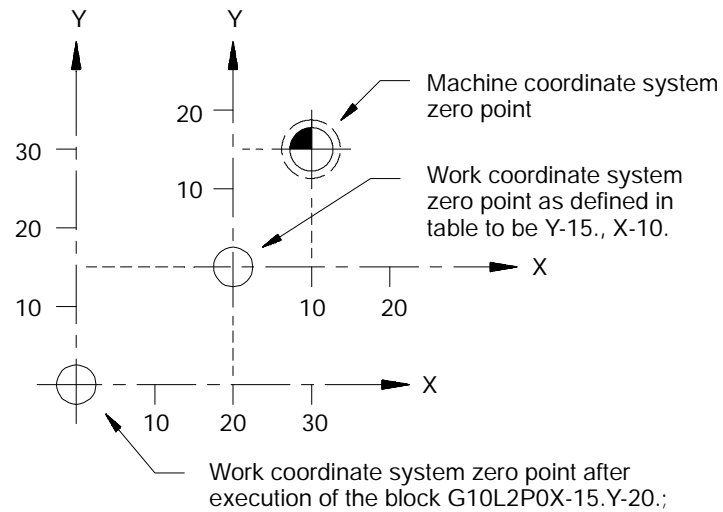
Example 11.4 and Figure 11.9 illustrate how the work coordinate system is shifted using G10.

Example 11.4
Changing the External Offset Through G10 Programming

Program Block	Comments
G10L2P1X-15.Y-10.;	defines work coordinate system zero point to be at X-15, Y-10 from the machine coordinate system zero point
G90;	
G10L2P0X-15.Y-20.;	sets external offset of X-15, Y-20 moving work coordinate system zero point to be at X-30, Y-30 from the machine coordinate system zero point

Important: This modification is permanent; the new table values for the work coordinate systems are saved even when control power is turned off.

Figure 11.9
Results of Example 11.4



11.4 Offsetting the Work Coordinate Systems

This section discusses the more temporary ways of offsetting the work coordinate systems. These offsets are activated through programming and are cancelled when an M02 or M30 is executed, a control reset is performed, or power to the control is turned off.

Important: All of these offsets are global in nature. This means that they will apply to all of the work coordinate systems. When changing work coordinate systems (Programming G54-G59.3), consideration should be given to the effects of these offsets on the new work coordinate system.

Tool geometry and wear offsets are not affected by an offset to the work coordinate system.

Important: It is recommended that tool offsets for geometry and wear be cancelled before any work coordinate system offsets are executed. If tool offsets are not cancelled, the work coordinate system offset will be added to the active tool offset. This can cause confusion when changing tool offsets later in the program (see chapter 20 on cancelling tool offsets).

11.4.1 Coordinate Offset Using Tool Position (G92)

The G92 command in a part program is used to offset the currently active work coordinate system relative to the current tool position. A G92 block in a program will offset the zero point of the work coordinate system a specified distance from the current tool position.

When a G92 command is executed in a program it cancels any other active work coordinate system offsets that may have been in effect including G52 offsets, jogged offsets, or set zero offsets. Note that external offsets are not affected. When the PAL flag \$INHR is set, it cancels G92.

Important: A tool offset is not automatically canceled when a G92 block is executed. Be aware that this may result in undesired effects on the work coordinate system when tool offsets are changed later.

The following G92 block offsets the work coordinate system so that the current tool position takes on the coordinate values programmed in the G92 block.

```
G92 X__ Y__ Z__;
```

For example specifying values of zero for all axes in a G92 block causes the current tool position to become the zero point of the current work coordinate system.

Execution of a G92 block does not produce any axis motion.

Important: Any axis not specified in the G92 block will not be offset, and the current coordinate position for that axis will remain unchanged.

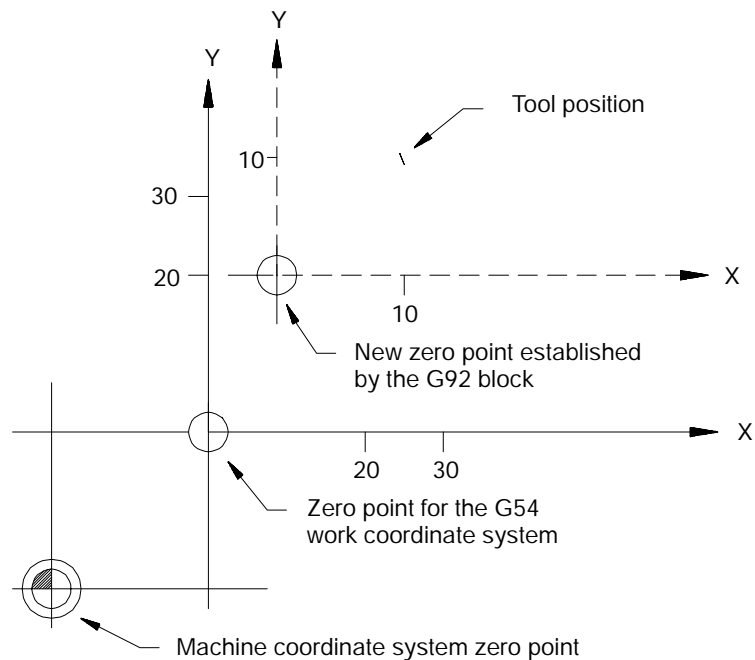
Once the work coordinate system is offset, all absolute positioning commands in the program are executed as coordinate values in the offset coordinate system.

Example 11.5
Work Coordinate System Offset (G92)

Program Block	Comment
X25. Y35. ;	rapid move to X25, Y35 in the G54 work coordinate system.
G92X10. Y10. ;	Redefines current axis position to have the coordinates X10, Y10

Note that the zero point of the offset G54 work coordinate system will be 10 units away from the current tool location in both the X and Y directions. If the Y value had not been entered in the G92 block, the Y coordinate location would have remained unchanged (Y35.)

Figure 11.10
Results of Example 11.5





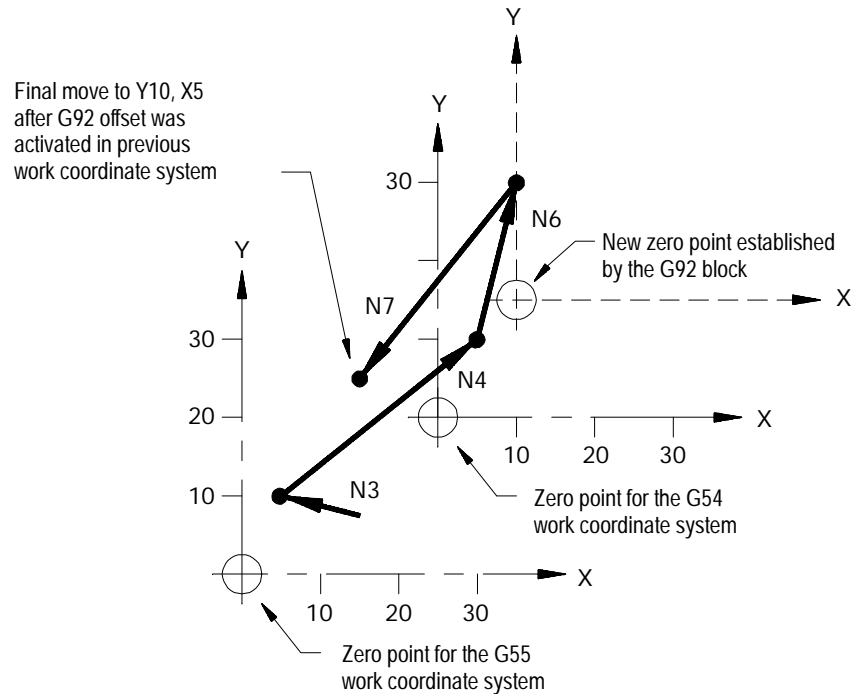
CAUTION: G92 offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59.3) unless the effects of the offset have been considered.

Example 11.6 shows the effect of changing work coordinate systems while the G92 offset is active.

Example 11.6
Changing Work Coordinate Systems With Offset Active

Program	Comment
N1 G10L2P1X0Y0;	Define G54 work coordinate system zero point to be positioned X0, Y0 away from the machine coordinate system
N2 G10L2P2Y20.X25.;	Define G55 work coordinate system zero point to be positioned Y20, X25 away from the machine coordinate system
N3 G55Y10.X5.;	Move to Y10, X5 in the G55 work coordinate system
N4 G54Y10.X5.;	Move to Y10, X5 in the G54 work coordinate system
N5 G92Y-5.X-5.;	Offset current tool position to be at Y-5, X-5
N6 Y15.X0.;	Move to Y15, X0 (offset still active)
N7 G55Y10.X5.;	Move back to Y10, X5 in the G55 work coordinate system with the G92 offset still active

Figure 11.11
Results of Example 11.6



In Example 11.6 and Figure 11.11, the offset entered for the G54 work coordinate system has also shifted the G55 coordinate system. Any offsets discussed in this section alter all of the work coordinate systems (G54 - G59.3) at the same time.

11.4.2 Offsetting Coordinate Zero Points (G52)

To offset a work coordinate system an incremental amount from its zero point, program a G52 block that includes the axis names and distances to be offset.

G52 X__ Y__ Z__ ;

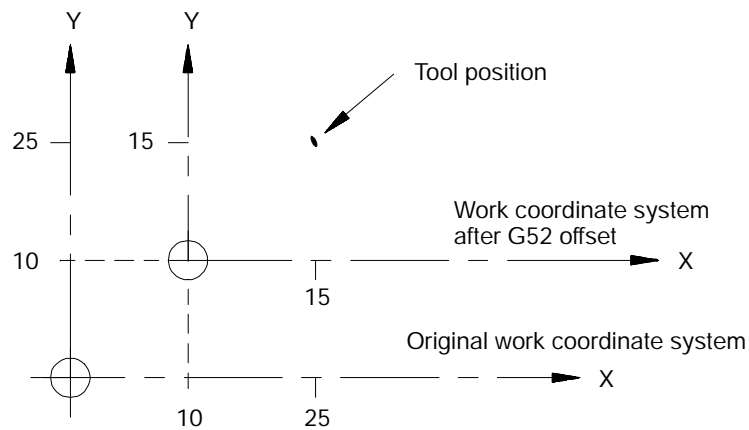
The above command offsets the current work coordinate system by the axis values that follow the G52 command.

In Example 11.7, no axis motion takes place when the G52 block is executed, however, the work coordinate system position values change. See Figure 11.12.

Example 11.7
Work Coordinate System Offset By G52

Program Block	Machine Coordinate Position	Work Coordinate Position
G01F55X25. Z25. ;	X25 Y25	X25 Y25
G52X10. Y10. ;	X25 Y25	X15 Y15

Figure 11.12
Results of Example 11.7



The G52 work coordinate system zero point offset can be canceled by programming a G52 block with zero values for the axes to be cancelled. The following block would cancel the work coordinate system offset for the X axis only.

```
G52 X0;
```

A G52 offset can also be canceled by executing a G92 or G92.1, performing a control reset or an E-STOP reset operation, or executing an end of program M30 or M02. A G92 command only cancels a G52 offset if one is active when the G92 block is executed. A G52 offset may be activated at some time after the G92 block is executed even if a G92 offset is still in effect.



CAUTION: G52 offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59) unless the effects of the offset have been considered.

11.4.3
{SET ZERO} Offset

When a Set Zero operation is performed the control shifts the current work coordinate system so that the current tools position is the zero point of the coordinate system. The axis that set zero is effective in is selected through PAL (refer to system installers Documentation) or by the current jog axis if using the {FRONT PANEL} option.

The Set Zero offset is similar to the execution of a G92 X0 Y0 Z0 block, with one exception. Unlike a G92, the set zero will not cancel a G52 offset. The G52 will remain active and continue to offset the current tool position in the work coordinate system. When the G52 offset is canceled later the coordinate system will shift.

The Set Zero offset can be canceled by programming a G92.1 command, executing a control reset operation, executing an E-STOP reset operation, or programming an end of program M30 or M02 command.



CAUTION: Set Zero offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59.3) unless the effects of the offset have been considered.

Example 11.8
Typical Set Zero Offset Application

Operation	Comment
-Manual jog-	axes are manually jogged to a location where the operator has determined that a special operation must be performed.
-Set Zero-	operator performs a Set Zero offset to establish the work coordinate system zero point at the current axis location.
-Run program-	a generic special operation program can now be executed from the axis coordinate position that resulted from the manual jog and Set Zero.

The set zero offset may be performed through an optional switch installed by the system installer or through the {SET ZERO} softkey as described in section 2.3 on using the {FRONT PANEL} softkey.



CAUTION: Set Zero offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59.3) unless the effects of the offset have been considered.

11.4.4 Jogging an Offset

The jog offset feature allows the operator to manually create a desired offset by jogging the axes during an automatic or MDI operation.

Important: This feature will function only if the system installer has supplied a special switch and the appropriate PAL programming. See the “Jog Offsets” and “Jog-on-the-fly” PAL flags in your PAL reference manual or else refer to the documentation supplied by the system installer.

This feature is activated by pressing a special switch after interrupting an automatic or MDI operation. Any manual jog moves made will be added to the current work coordinate position as an offset. When cycle start is pressed to continue execution, the jogged distance for each axis will remain as a coordinate offset for that axis.



CAUTION: Jog offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59.3) unless the effects of the offset have been considered.

The Jog offsets can be canceled by programming a G92.1 command, executing a control reset operation, executing an E-STOP reset operation, or programming an end of program M30 or M02 command.

To use this feature follow these steps:

1. Press <CYCLE STOP> or <SINGLE BLOCK> on the MTB panel to interrupt automatic or MDI operation.
2. Turn on the switch to activate the jog offset function (refer to documentation provided by the system installer).
3. Change to manual mode unless the control is equipped for the “Jog-on-the-Fly” feature which allows jogging in automatic or MDI modes (refer to documentation prepared by the system installer).
4. Jog the axes using any of the available jog types (with the exception of homing) as described in section 4.1. The control will add the amount of the jog move as an offset for each jogged axis.

5. Return to Automatic or MDI mode. When the <CYCLE START> button is pressed, execution will continue from the new tool location, at the jogged offset.

Important: When the jog offset move is made the axis position displays do not change on the screen (unless the currently active screen is displaying absolute position coordinates as described in section 8.1). This is because the coordinate values in the work coordinate system are being offset as the axes are being jogged.

11.4.5 Canceling Coordinate System Offsets (G92.1)

The G92.1 command cancels the following offsets:

- G92 work coordinate system offset
- G52 zero point offset
- {SET ZERO} offset
- Jog offset
- Reset G54 - G59.3 coordinate system to default condition

It will not cancel an external offset (see section 11.3).

The G92.1 block will also reestablish the default work coordinate system as set in AMP by the system installer. It will cancel or activate the coordinate system (G54-G59.3) as set in AMP to establish the default coordinate system.

The G92.1 block must be programmed with no axis words. Axis words in a G92.1 block will generate an error. When the G92.1 block is executed all G92, G52, {SET ZERO}, and Jog offsets are canceled on all axes. It is not possible to cancel the offsets on only one or more of the axes.

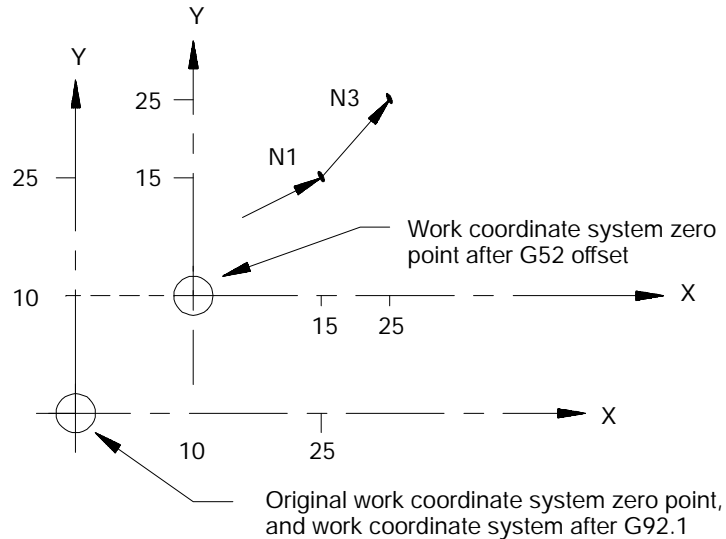
No axis motion takes place during execution of a G92.1 block. Axes remain at their last programmed position while the work coordinate system is adjusted to remove all offsets.

The following example demonstrates the G92.1 offset cancel.

Example 11.9
G52 Offset Cancelled By a G92.1

Program Blocks	Comment
N1 G01Y25. X25. ;	move to Y25, X25
N2 G52Y10. X10. ;	work coordinate system is offset by Y10, X10
N3 Y25. X25. ;	move to Y25, X25 in the offset coordinate system
N4 G92. 1;	G52 offset is cancelled, program position displays axis position at X35Y35.

Figure 11.13
Results of Example 11.9



11.4.6 Canceling Selected Coordinate System Offsets (G92.2)

The G92.2 command cancels the following offsets:

- G92 work coordinate system offset
- {SET ZERO} offset
- Jog offset

It will not cancel an external offset (see section 11.3), reset the current work coordinate system (G54-G59.3) or cancel a G52 offset.

The G92.2 block must be programmed with no axis words. Axis words in a G92.2 block will generate an error. When the G92.2 block is executed all G92, {SET ZERO}, and Jog offsets are canceled on all axes. It is not possible to cancel the offsets on only one or more of the axes.

No axis motion takes place during execution of a G92.2 block. Axes remain at their last programmed position while the work coordinate system is adjusted to remove these offsets.

11.5 PAL Offsets

The system installer has the option of activating, deactivating, or altering the value of the following offsets through PAL:

Work coordinate systems

External offset

Tool length offsets (geometry and wear)

Tool diameter offsets (geometry and wear)

These offsets may be modified through a PAL display page created by the system installer or through some other input to PAL.

Be aware that there may be an impact to the activation of offsets if a part program is already active for automatic execution. Typically any blocks that have been read into the controls look ahead buffer (see section 21.6.8) will be re-setup and use the newly modified offset value. However if a cutter compensation offset has been modified by PAL, be aware that the control will not update the look ahead buffer unless the offset is currently active. Refer to documentation supplied by the system installer for details on specific PAL offset operations.

END OF CHAPTER

Overtravels and Programmable Zones

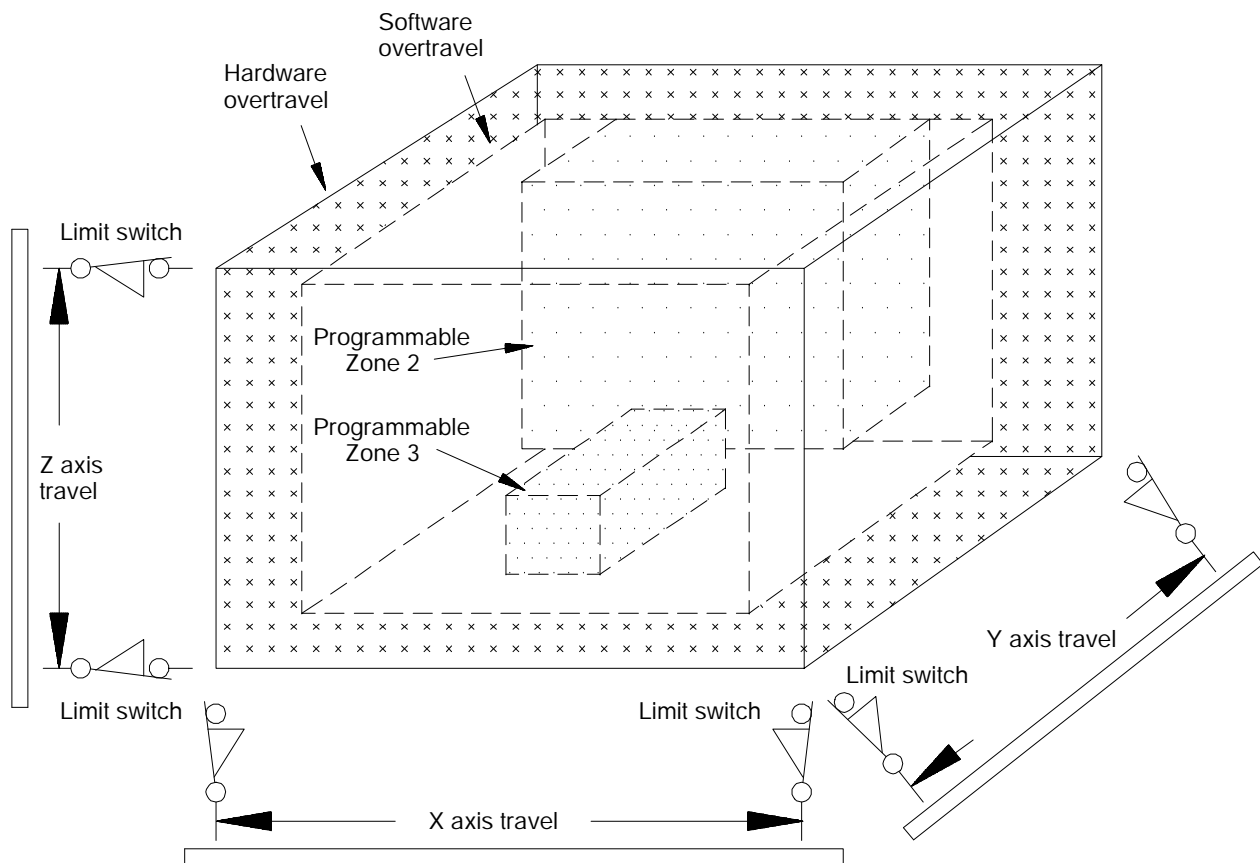
12.0 Chapter Overview

This chapter discusses overtravels and programmable zones.

12.1 Overtravels and Programmable Zones

Overtravels and programmable zones define areas that restrict the movable range of the cutting tool. The control is equipped to establish two overtravel areas and two programmable zones as illustrated in Figure 12.1.

Figure 12.1
Overtravels and Zones



There are two types of overtravels.

- **Hardware overtravels** -- Established by the system installer by mounting mechanical limit switches on the movable range of the axes.
- **Software overtravels** -- Established in AMP by the system installer designating coordinate values in the machine coordinate system.

There are two types of Programmable Zones.

- **Programmable Zone 2** -- Established by the operator, or person in charge of job setup. The machine coordinate system boundaries for this zone are entered in a table. Programmable zones may be turned on and off in the part program.
- **Programmable Zone 3** -- Established by the operator, programmer, or person in charge of job setup. The machine coordinate system boundaries for this zone are entered in a table or through programming. Programmable zones may be turned on and off in the part program.

12.1 Hardware Overtravels

When the machine tool is set up the system installer should have installed a set of two mechanical limit switches on each axis. These limit switches are installed in a position such that when the machine attempts to move beyond a range determined by the system installer the limit switch will be tripped. When the limit switch is tripped axis motion stops. The area defined by these limit switches is referred to as the hardware overtravel.

These switches are installed to prevent the machine from motion that exceeds a range that might cause damage to the machine. Frequently the system installer will wire the hardware overtravel directly into the E- stop string. This will stop all motion and disable the axis drives. Refer to the literature provided by the system installer for instructions on moving axes out of hardware overtravel.

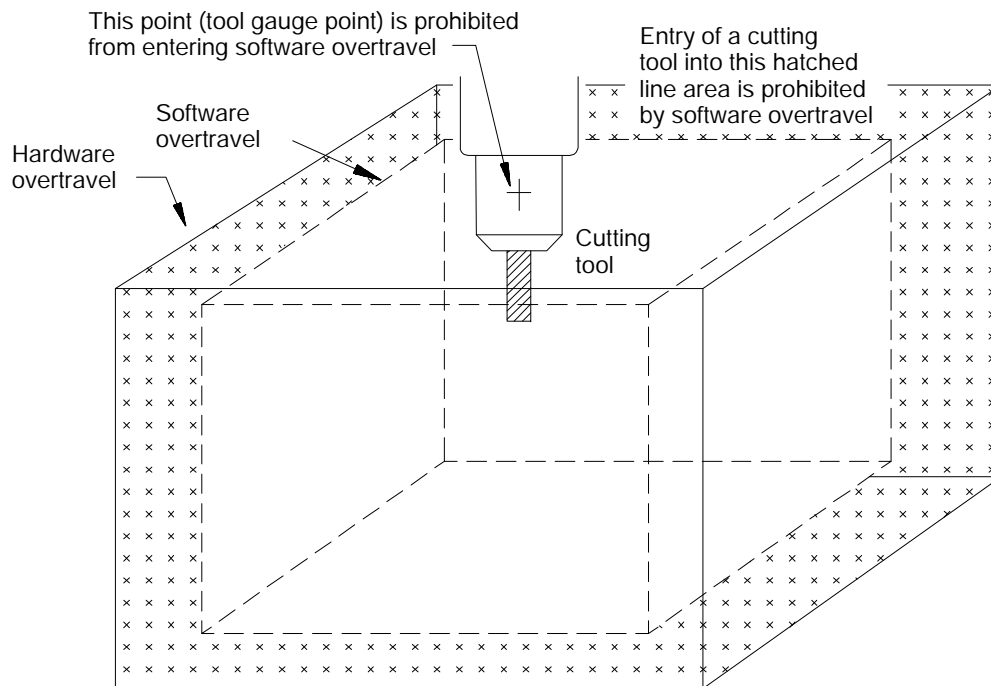


CAUTION: The area defined by a hardware overtravel does not take in to account tool offsets. This may allow the actual tool to enter the restricted area without the axes entering it.

12.2 Software Overtravels

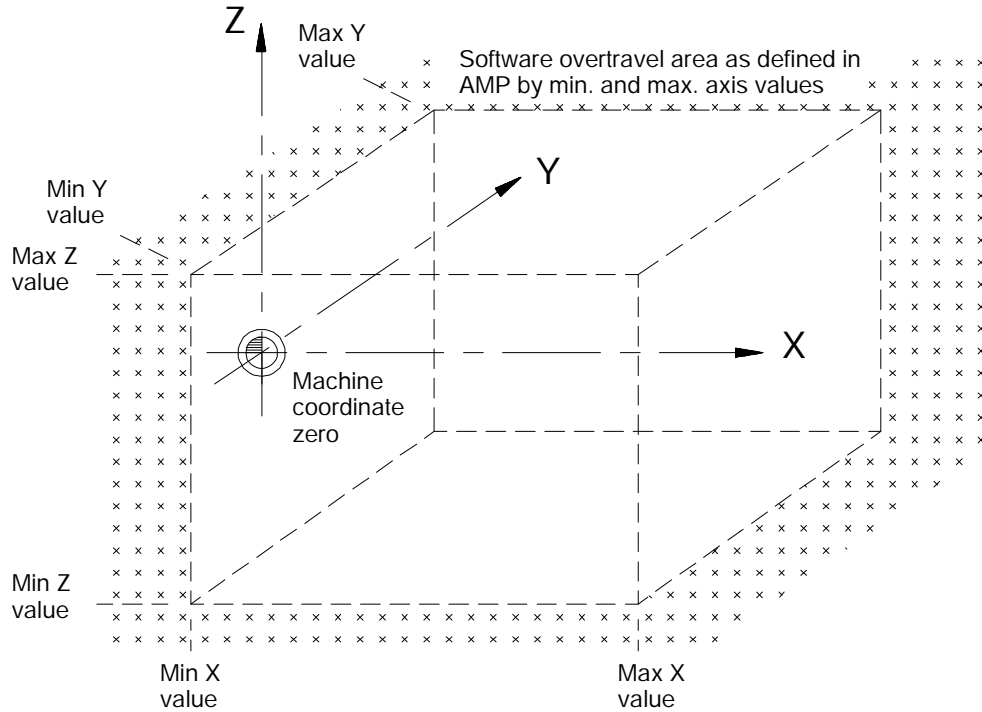
The coordinate values of the points defining the software overtravels are set in AMP by the system installer. This overtravel may only be disabled by the system installer in AMP. If the system installer has enabled the software overtravels the control will not be allowed to exit the area defined by the software overtravels.

Figure 12.2
Software Overtravels Established in AMP



The system installer selects values that represent a maximum and a minimum value in the form of coordinate values for each axis. These coordinate values define points on the machine coordinate system. Axes are then not allowed to move past the coordinate value representing the maximum and minimum value on each axis. This limited range of motion is referred to as the software overtravels.

Figure 12.3
Area Defining Software Overtravel



Typically the software overtravels are located within the hardware overtravels (maximum axis travel defined by the limit switches on each axis) and are used to keep the axes within the range the system installer determines is usable for that particular machine's application.

The area defined by the software overtravels becomes effective after completion of the initial homing operation at power up. For details on how the control reacts to a entry into an overtravel area see section 12.5.



CAUTION: The area defined by a software overtravel does not take in to account any tool offsets. This may allow the actual tool to enter the restricted area without the axes entering it. Make sure this is considered when the software overtravel is established.

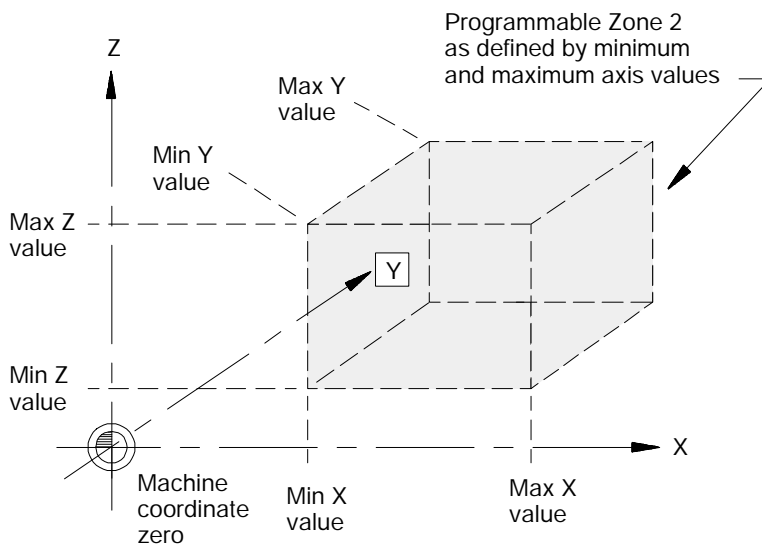
12.3
Programmable Zone 2 (G22,
G23)

Programmable zone 2 defines an area which the tool axes may not enter. Generally, zones are used to protect some vital area of the machine or part located within the software overtravels.

Important: Programmable zones are defined using coordinates in the machine coordinate system. They are not affected by any changes in the work coordinate system, including external offsets. They do however take into account any active tool offset.

Values for programmable zone 2 are entered in the programmable zone tables as discussed in section 3.5. These values represent a maximum and a minimum value in the form of machine coordinate values for each axis. The area defined by these points establishes the boundaries for programmable zone 2.

Figure 12.4
Area Defining Programmable Zone 2



Programmable zones 2 and 3 become active when a G22 block is executed and are cancelled when a G23 is executed. Both G22 and G23 are modal commands.

Important: You must home your axes first before the control will enable the programmable zones.

Important: When made active the current tool location must be outside of the area defined by programmable zone 2.

G22 programmable zone 2 and 3 active
G23 programmable zone 2 and 3 inactive

G23 is normally automatically made active at power up though this is determined by the system installer in AMP. Any zone that is activated in a program or MDI block, remains active even after a control reset, E- STOP reset, or end of program block (M02 or M30).

Important: If programming a G22, any axis words included in the block will be stored as the coordinates for programmable zone 3 (see section 12.5).

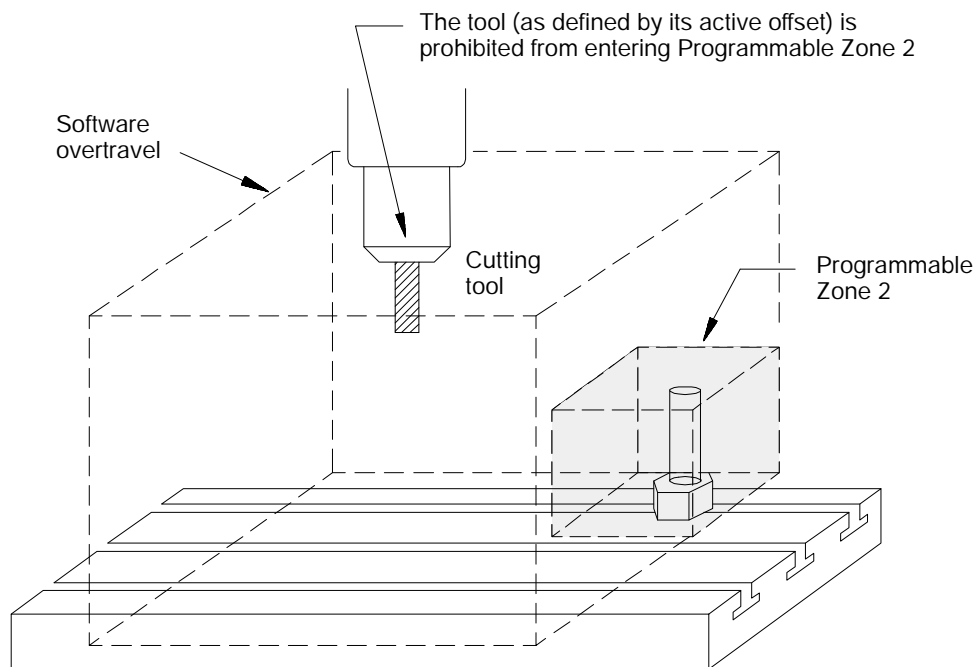
If an attempt is made to designate some other command in a G22 or G23 block, other than a G code in the same modal group, for example:

```
G22 G01 X12. ;
```

the control issues the error message:

“UNUSABLE WORDS IN ZONE BLOCK”

Figure 12.5
Programmable Zone 2



Programming this G-code:	turns Zone 2:	turns Zone 3:
G22	On	On
G22.1	Off	On
G23	Off	Off
G23.1	No Change*	Off

* A G23.1 turns on programmable zone 2 if it is the default power up condition configured in AMP (also activated at a control reset). G23.1 does not turn on programmable zone 2 when it is activated in a part program.

Your system installer can also turn zones on and off with PAL. Refer to your system installer's documentation for more information.

For details on how the control reacts to entry into a prohibited area see section 12.5.

12.4 Programmable Zone 3 (G22.1, G23.1)

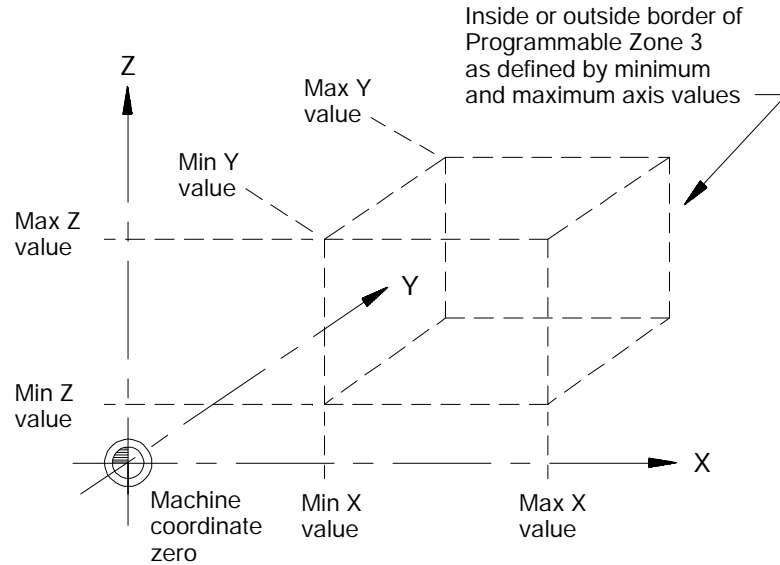
Important: G22.1, G23, and G23.1 must be programmed in blocks without other commands. If programming a G22, any axis words included in the block will be stored as the coordinates for programmable zone 3.

Programmable zone 3 can define an area which the tool may not enter or exit. Generally, zones are used to protect some vital area of the machine or part located within the software overtravels.

Important: Programmable zones are defined using coordinates in the machine coordinate system. They are not affected by any changes in the work coordinate system, including external offsets. They do however take into account any active tool offsets.

Values for programmable zone 3 are entered either in the programmable zone table (discussed in section 3.5) or through a G22 program block. A maximum and a minimum coordinate value (in the machine coordinate system) is assigned for each axis. The resulting coordinates define the boundaries for programmable zone 3.

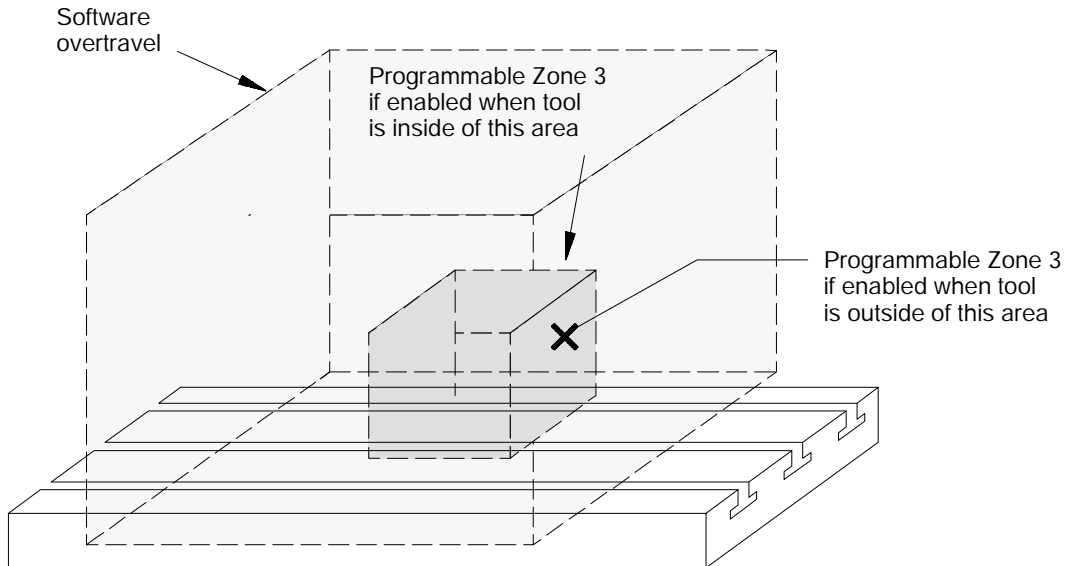
Figure 12.6
Area Defining Programmable Zone 3



Unlike the software overtravels, which define the perimeter of an area from which the cutting tool may not exit, programmable zone 3 can define either an area that the cutting tool may not exit or an area that the cutting tool may not enter. This is determined by the current tool location when programmable zone 3 is made active.

- It defines an area that the cutting tool may not exit if the tool is currently inside the zone when the zone is activated.
- It defines an area that the cutting tool may not enter if the tool is currently outside the zone when the zone is activated.

Figure 12.7
Programmable Zone 3 Zero Point (Machine Coordinate System)



Programmable zone 3 becomes active when either the G22 or G22.1 code is executed. It is made inactive when the G23 or G23.1 code is executed.

Important: You must home your axes first before the control will enable the programmable zones.

Program G-code:	To turn on these zones:	To turn off these zones:
G22	2 and 3	not applicable
G22.1	3	2
G23	not applicable	2 and 3
G23.1	not applicable	3

G22.1 and G23.1 are modal (G22.1 cancels G23.1 and G23.1 cancels G22.1). G22 and G23 belong to a different modal group than G22.1 and G23.1. This means that programmable zone 2 may be activated without activating programmable zone 3 if a G23.1 is executed.

G23 is automatically active at power up, control reset, or E-STOP reset as the default G code for this modal group.

Your system installer can also turn zones on and off with PAL. Refer to your system installer's documentation for more information.

Programming zone 3 values (3 or less axes)

You can reassign values for the parameters that establish programmable zone 3 by programming axis words in a G22 program block. Two methods are available. This section discusses programming values for zone 3 when 3 or less axes have been configured on the system (this does not include any spindle).

Define values for programmable zone 3 using the G22 command followed by axis words in the following format:

G22 X__ Y__ Z__ I__ J__ K__;

Where:	Defines:s
Primary axis words (normally X, Y, and Z)	maximum zone limits
Integrand words (normally I, J, and K)	minimum zone limits

These axis words can vary. Refer to your system installer's documentation.

The zone values entered in a G22 block always reference coordinate values in the machine coordinate system.

If a value for a maximum axis parameter is less than the value set for an axis current minimum parameter, or if a value for a minimum axis parameter is set greater than the value set for an axis current maximum value, the control displays the message:

“INVALID VALUE (MAX < MIN) FOR ZONE 3 AXIS (X)”

This message displays the name of the axis that has been set incorrectly. It does not indicate if it is the minimum or maximum value that is incorrect.

If the same integrand word is assigned in AMP by the system installer to more than one axis, that integrand word will set the lower zone 3 limit for all axes with that integrand.

Programming zone 3 values (4 or more axes)

You can reassign values for the parameters that establish programmable zone 3 by programming axis words in a G22 program block. Two methods are available. This section discusses programming values for zone 3 when 4 or more axes have been configured on the system (this does not include any spindle).

This method differs from the three axis method in that the same integrands can be used again for different axes (necessary since the control only supports three integrand words). Assume the following AMP configuration:

Primary Axis name	X	Y	Z	U	V	W	A	B	C
Axis Integrand	I	J	K	I	J	K	I	J	K

These axis words can vary. Refer to your system installer's documentation.

Define values for programmable zone 3 using the G22 command followed by axis words in the following format:

```
G22 X__ Y__ Z__ I__ J__ K__ ;
G22 U__ V__ W__ I__ J__ K__ ;
G22 A__ B__ C__ I__ J__ K__ ;
```

Where:	Defines:s
Primary axis words	maximum zone limits
Integrand words (normally I, J, and K)	minimum zone limits

Using this method, the same integrand word assigned in AMP to more than one axis correspond only to the primary axis words programmed in the G22 block. Integrand words cannot be programmed alone (without a primary axis word in the G22 block). The following example assumes a machine with axes configured as shown above.

These blocks:	Results in:
G22 X10 I-10 Y14 J-14 Z1 K-1; G22 U5 I-5 V13 J-2 W11 K10; G22 A3 I2 B7 J-7 C12 K11;	upper and lower zone 3 limits for all 9 axes are changed. Zones 2 and 3 are both activated when the first block in this series of blocks is executed.
G22 X1 Y2 Z3 U4 V5 W6 A7 B8 C9;	upper zone 3 limits are changed for all 9 axes. Zones 2 and 3 are both activated.
G22 X1 Y2 Z3 U4 V5 W6 A7 B8 C9 I-1 J-2 K-3;	upper and lower zone 3 limits for all 9 axes are changed. (I sets lower for X, U, and A; J sets lower for Y, V, and B; K sets lower limits for Z, W, and C). Zones 2 and 3 are both activate.
G22 K-10;	error is generated. Current status of zones remains in current state (on or off).



WARNING: When using multiple blocks to set the zone 3 limits, keep in mind zone 3 is activated after the first G22 block. This will result in zone 3 being activated before you have completed changes to the zone 3 values. This can cause the control to miss-interpret zone 3 as an internal or external zone, depending on the tool location at the time of the zone activation.

The zone values entered in a G22 block always reference coordinate values in the machine coordinate system.

If a value for a maximum axis parameter is less than the value set for an axis current minimum parameter, or if a value for a minimum axis parameter is set greater than the value set for an axis current maximum value, the control displays the message:

“INVALID VALUE (MAX < MIN) FOR ZONE 3 AXIS (X)”

This message displays the name of the axis that has been set incorrectly. It does not indicate if it is the minimum or maximum value that is incorrect.

12.5 Resetting Overtravels

The control stops tool motion during overtravel conditions. Overtravel conditions may occur from 3 causes:

- hardware overtravel -- the axes reach a travel limit, usually set by a limit switch or sensor mounted on the axis. Hardware overtravels are always active.
- software overtravel -- commands cause the axis to pass a software travel limit. Software overtravels are active only after the axis has been homed provided the feature has been activated in AMP by the system installer.
- Programmable Zone Overtravel -- The tool reached a travel limit established by independent programmable areas. Programmable Zones are activated through programming the appropriate G code.

In all cases, the control issues an error message.

When an overtravel condition occurs, all axis motion is forced to a stop, the control is placed in cycle stop and one of the following applicable error messages is displayed.

Where :	Is :
"HARDWARE OVERTRAVEL (-) BY AXIS (X)"	This message indicates that the specified axis has tripped either the + or - hardware limit switch mounted on the machine.
"SOFTWARE OVERTRAVEL (+) BY AXIS (X)"	This message indicates that the specified axis has entered the overtravel area defined by the softlimits in either a positive or negative direction.
"VIOLATION OF ZONE (2) BY AXIS (X)"	This message indicates that a tool has reached the specified axis overtravel area defined by either programmable zone 2 or 3.

When an overtravel of any type has taken place, axis feed in the same direction as the feed causing the overtravel is not allowed. Only axis feed in the reverse direction is possible.

How a hardware overtravel condition is reset depends on the ESTOP circuit design and the way PAL was programmed by the system installer.

To reset a software or programmable zone overtravel condition:

1. Determine whether the control is in E-STOP. If it is not, go to step 4.
2. Look for and eliminate any other possible conditions that may have caused emergency stop, then make sure that it is safe to reset the emergency stop condition.
3. Press the button to reset the emergency stop condition. If the E-STOP does not reset it is a result of some cause other than overtravel causing E-STOP.
4. Make sure it is safe to move the axis away from the overtravel limit.
5. Use any of the jog features discussed in section 4.1, except homing to manually move the axis away from the limit.

END OF CHAPTER

Coordinate Control

13.0 Chapter Overview

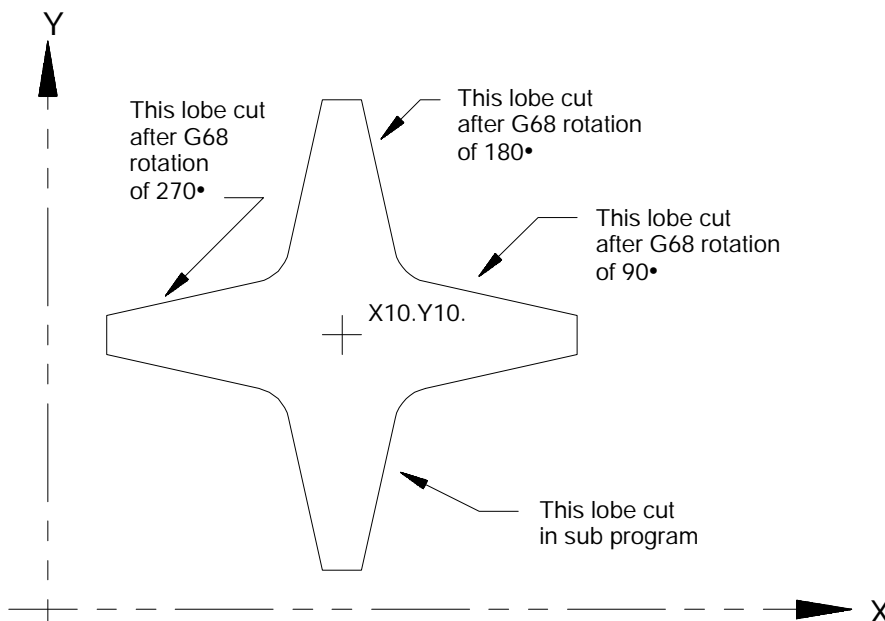
This chapter describes:

How to:	On page:
rotate a coordinate system	13-1
select a plane	13-11
use absolute and incremental modes	13-12
apply inch and metric measures	13-13
use scaling	13-14

13.1 Rotating the Coordinate Systems

The control has a feature (G68) that can rotate the work coordinate system. There is also a feature called External Part Rotation which rotates all work coordinate systems by simulating a rotation of the machine coordinate system. Rotating the coordinate systems can prove to be useful when cutting a part that contains symmetrical geometries (see Figure 13.1).

Figure 13.1
Part With Symmetrical Geometry



13.1.1
Rotating the Current Work
Coordinate System (G68,
G69)

To rotate the current work coordinate system, program the following command.

G68 X__ Y__ Z__ R__;

Where :	Is :
X, Y, Z	Specify the center of rotation using <u>only the two</u> axis words that are in the current active plane (G17, G18, or G19). The value entered with these axis words represent a position in the current work coordinate system. The values specified with the axis words are always absolute coordinate values, the center of rotation cannot be specified as an incremental position.
R	Specify the angle of rotation that the coordinate system is to be rotated at. Enter a value in units of degrees. R is always measured parallel to the first major axis in the current plane. Positive R is measured counter clockwise, and a negative R is measured clockwise. If you do not specify an angle using an R word in the G68 block, the control uses the value of the "Programmable Part Rotation" angle field from the rotation table shown in figure Figure 13.6.

Important: If the first motion command following the G68 command is an incremental move, the center of rotation as described in the G68 command is ignored and the coordinate system is rotated about its zero point (see Example 13.1). The first motion command following a G68 block cannot be a circular move.

Any unnecessary parameters in the G68 block are ignored. Any G codes other than a plane selection (G17, G18, or G19) or a change from absolute or incremental mode (G90, or G91) specified in a G68 block will result in an error.

The G68 block will not create any axes motion. Position displays change due to the alteration of the work coordinate system.

Any rotation of the work coordinate system by programming a G68 command will rotate only the currently active work coordinate system. When changing to a different work coordinate system, the rotation will not be applied to the new work coordinate system. When changing back to the rotated coordinate system the rotation will still be in effect.

If you do not program an R word in the G68 block, the value of the angle for programmed rotation is taken from the part rotation screen. Access this screen as described for external part rotation on page 13-8. The last field on this screen is the programmable part rotation angle. This angle is only used when the R word is excluded in the G68 block.

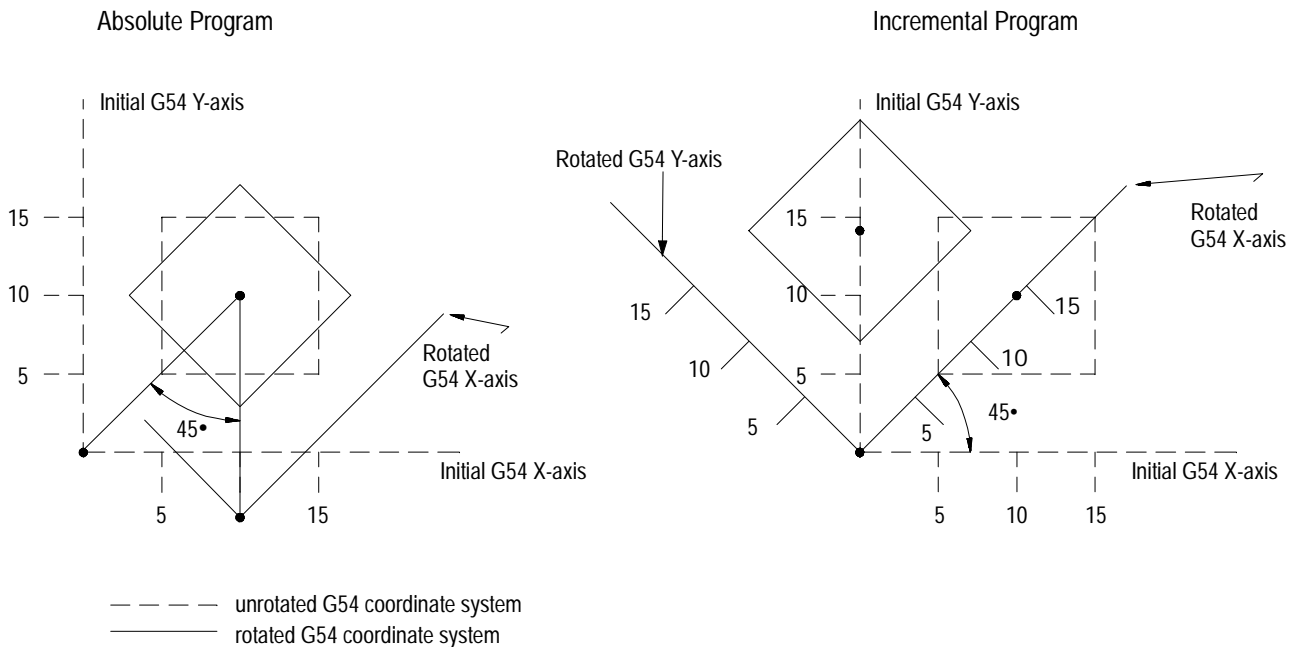
Example 13.1
Rotating the Active Work Coordinate System (G68)

These program blocks cause the rotation of the active work coordinate system as shown in Figure 13.2.

ABSOLUTE PROGRAM	INCREMENTAL PROGRAM
N1 G54 G17 G00;	N1 G54 G17 G90;
N2 G90 X0. Y0. F500;	N2 G00 X0. Y0. ;
/N3 G68 X10 Y10 R45;	/N3 G68 X10 Y10 R45;
N4 G90 G00 X5. Y5. ;	N4 G91 G00 X5. Y5. ;
N5 G01 X15. F100;	N5 G01 X10 F100;
N6 Y15. ;	N6 Y10;
N7 X5. ;	N7 X-10;
N8 Y5. ;	N8 Y-10;
N9 M30;	N9 G69;
	N10 M30; G54 G00;

If optional block delete 1 is set "ON", the control will cut the part shown with a dashed line in Figure 13.2. If optional block 1 is set "OFF" the control will cut the part shown with a solid line in Figure 13.2.

Figure 13.2
Results of Example 13.1



Note that in the preceding figure the center of rotation programmed in the G68 block is ignored when the block immediately following the G68 is an incremental motion block.

Angles and centers of rotation for G68 blocks are modal and remain in effect for following G68 blocks until a new center of rotation or angle is specified with a G68 command.

Important: It is possible to rotate all of the work coordinate systems at once by using the external part rotation.

If rotating the coordinate system again in the same plane using another G68 command:

- while in incremental mode, the angle of rotation is taken from the current rotated coordinate position (see Figure 13.3)
- while in absolute mode, the angle of rotation is taken from the original position

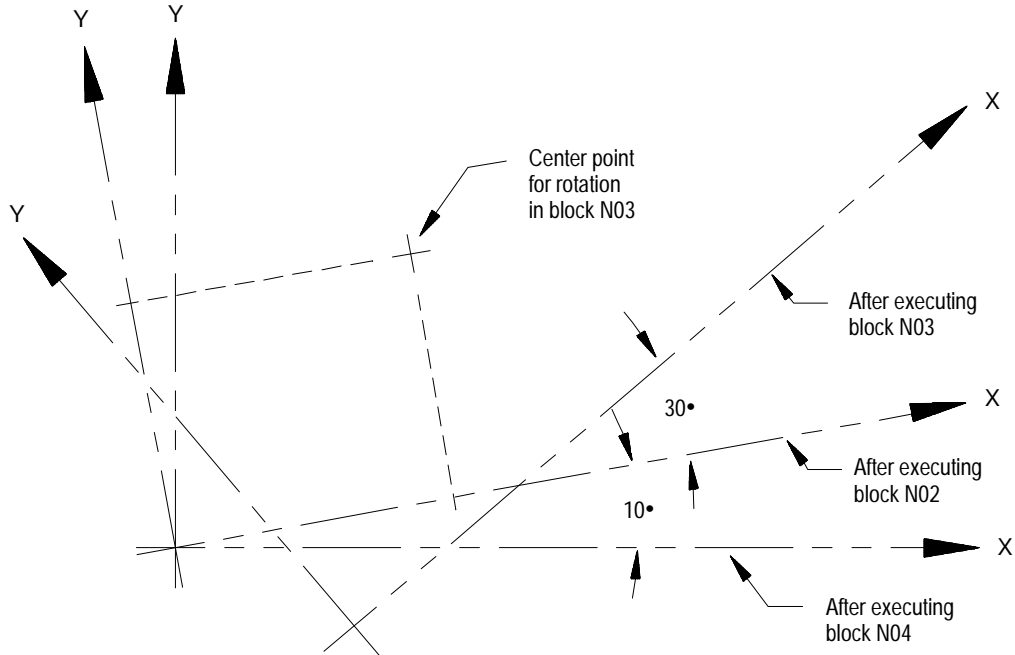
Rotating the coordinate system again in a different plane using another G68 is not allowed.

Executing a G69 cancels all G68 rotations and returns the coordinate system back to its original orientation. Local rotation of a work coordinate system using the G68 command is also canceled when the control executes an M30 or M02 code in a program.

Example 13.2
Multiple Rotation of the Coordinate System While in Incremental Mode

Program Block	Comment
N01 G54 G91;	Incremental mode
N02 G68X0Y0R10;	Rotates the current work coordinate system 10 degrees.
N03 G68X5. Y4. R30;	Rotates the current work coordinate system 30 degrees about a center point of X5. , Y4. for a total rotation from its original position of 40 degrees.
N04 G69;	Returns the work coordinate system to its original position of 0 degrees.

Figure 13.3
Results of Example 13.2

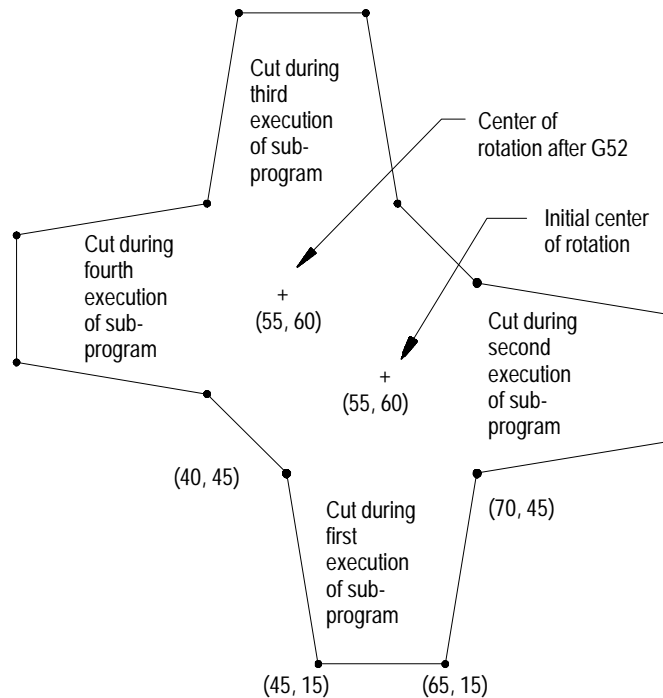


Rotating the work coordinate system can be helpful anytime a part has a repetitive shape. This feature combined with the G52 work coordinate system shift can reduce the size of a part program appreciably. The following program is an example of this.

Example 13.3
G68 Coordinate Rotation With G52 Coordinate System Shift

Main Program	Subprogram 1000
G17 G90 G00 X0 Y0;	G01 X45. Y15. F500. ;
G00 G90 X40 Y45. ;	X65. ;
M98 P1000 L4;	X70. Y45. ;
M30;	G68 X55. Y60. R90. ;
	M99;

Figure 13.4
Results of Example 13.3



13.1.2 External Part Rotation

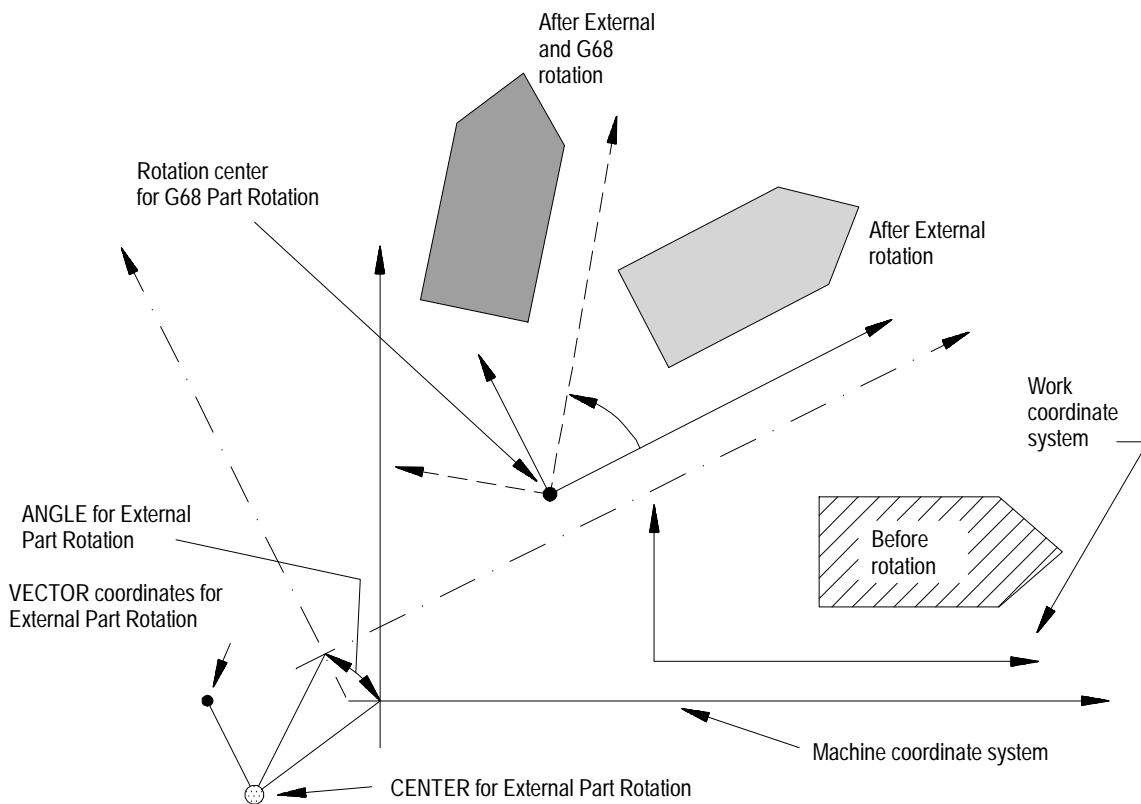
The external part rotation feature simulates a rotation of the machine coordinate system. Since all work coordinate systems are referenced from the machine coordinate system, rotating it would effectively rotate all work coordinate systems. However, software overtravels, programmable zone limits, homing, and positioning in the machine coordinate system are not affected.

When external part rotation is activated all work coordinate systems are rotated together by a specified amount, about a specified point.

External Part Rotation can be executed before or after rotation of the work coordinate system using the G68 command (as described in section 13.1.1). If a G68 is programmed to rotate the current work coordinate system, an additional rotation of coordinates will result as shown in Figure 13.5.

Any work coordinate system rotation that is to be done using the external rotation feature must be performed before program execution begins. Program execution may not be interrupted to perform a external part rotation. If an attempt is made to interrupt a program to perform an external part rotation the rotation will not become effective until the end of program (M02 or M30) command is read, a control reset, or E-STOP reset is performed.

Figure 13.5
External Part Rotation Followed By G68 Work Coordinate Rotation



Important: This feature only simulates rotation of the machine coordinate system for the purpose of establishing the preset work coordinate systems. Software overtravels, programmable zones, and any other positioning referenced on the machine coordinate system will remain unaffected by this rotation, but a program originally written within the limits can now be outside these limits.

Activating the External Part Rotation Feature

To activate the External Part Rotation feature, follow these steps:

1. Place the control in E-STOP and press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
--	------------------	--------	----------------	----------------	------------------	---

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→
--	----------------	-----------------	---------------	-----------------	--	---

2. Press the {COORD ROTATE} softkey. This will display the external part rotation parameters screen as shown below.

(softkey level 2)

↑	WORK CO- ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
---	-----------------	--------------	----------------	----------------	----------------	---

↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→
---	-----------------	------------------	--------	--	--	---

Figure 13.6
Typical External Part Rotation Parameter Screen

ENTER VALUE:		E- STOP				
		MODE = [MM]				
EXTERNAL	PART	ROTATION	[OFF]			
PLANE		X	Z			
CENTER		-2.440	-2.600			
VECTOR		0.000	0.000			
ANGLE		15.000				
PROGRAMMABLE		PART	ROTATION			
ANGLE			0.000			
↑	EXTERN ON/OFF					

3. Move the cursor to the desired parameter to be changed by pressing the up, down, left, right cursor keys. The selected parameter will be shown in reverse video.
4. Enter the new value for the desired parameter using the keys on the keyboard. The entered value will be shown on the input line (lines 2 and 3) of the CRT. The value may be modified on the input line as described in section 2.10.
5. When the desired value is displayed on the input line of the CRT, press the key. The value on the input line will replace the old parameter value.
6. Repeat until all parameters display the desired values.
7. Activate the external part rotation feature by pressing the {EXTERN ON/OFF} softkey. The external offset feature will toggle between on and off each time this softkey is pressed.

The work coordinate systems are all rotated as soon as the external rotation feature is activated. The current work coordinate system can be changed while an External Part Rotation is active. If changed, the new work coordinate system will be rotated as described by the External Part Rotation parameters.

The “PROGRAM” and “TARGET” position displays (as discussed in section 8.1) will not reflect an external part rotation since these values are relative to the active work coordinate system.

Since “ABSOLUTE” position displays always show coordinates relative to the machine coordinate system, they will reflect an external part rotation.

External Part Rotation Parameters

This table describes the parameters you use for external part rotation:

If you want to:	Use this parameter:	In this way:
select a plane	plane	Enter the axis names that define the plane to be rotated. Two separate values must be entered on this line. The first axis entered here is the axis that the angle of rotation is measured from. This parameter must match the active plane. If the external and program-selected planes do not match, the control generates a block decode error.
define the center of rotation	center	Enter a coordinate value for each axis in the selected plane. The center of rotation is a point on the machine coordinate system about which all the work coordinate systems will be rotated. The default value for the center of rotation is (0, 0).
define the vector	vector	Enter a coordinate value to define the “head” of the vector. The “tail” of this vector is the center of rotation. This parameter is optional. If you use the angle parameter, that value overrides this value. The resulting angle between this vector line and the first plane axis is the angle of rotation for the work coordinate system. All work coordinate systems rotate about the point defined by the center parameter. The default values are (0, 0). With the values of this parameter, the control calculates the angle of rotation and displays this value in the angle parameter.
define the angle of rotation	angle	Enter the angle at which that you want to rotate the work coordinate system. This value overrides the value of the vector parameter. Enter a positive value for clockwise rotation from the primary axis of the selected plane. If you do not enter a value the control displays the angle defined by the vector parameter.

13.2 Plane Selection (G17, G18, G19)

The control has a number of features that operate in specific planes. For that reason it is frequently necessary to change the active plane using a G17, G18, or G19.

Some of the features that are plane dependant are:

- Circular interpolation
- Cutter compensation
- Work Coordinate system rotation
- Many fixed cycle operations

Important: The system installer determines the planes defined by G17, G18, and G19 in AMP. Axes may not be assigned to the planes exactly as listed below. Refer to the documentation prepared by the system installer.

Typical axis names and their corresponding plane assignment are shown below:

- G17 -- plane defined by the X and Y axes (or axes parallel to X and Y)
- G18 -- plane defined by the Z and X axes (or axes parallel to Z and X)
- G19 -- plane defined by the Y and Z axes (or axes parallel to Y and Z)

Planes can be altered to accommodate additional axes parallel to the principle axes by programming those axes in the G17, G18, or G19 block. See Example 13.4.

Example 13.4 Altering Planes for Parallel Axes

Assuming the system installer has made the following assignments in AMP:

```
G17    -- the XY plane.
U axis -- parallel to X axis
V axis -- parallel to Y axis
```

Program block	Plane selected	Axis Motion
G17;	selects XY plane	None
G17 U0;	selects UY plane	U axis moves to zero
G17 V0;	selects XV plane	V axis moves to zero
G17 U0V0;	selects UV plane	U & V axes move to zero

Important: Any axis word in a block with plane select G-codes (G17, G18, G19) causes axis motion on that axis. If no value is specified with that axis word, the control assumes a value of zero or generates an error depending on how your system is AMPed.

13.3 Absolute/Incremental Modes (G90, G91)

There are two methods for programming axis positioning commands, absolute positioning and incremental positioning.

In the absolute mode, coordinates are referenced from the zero point of the active coordinate system. Absolute mode is established by programming a G90.

```
G90X40. Z20. ;
```

In the above block the control will move the axes to a position X40, Z20 as referenced on the active coordinate system.

G90 is a modal G code and remains active until cancelled by a G91.

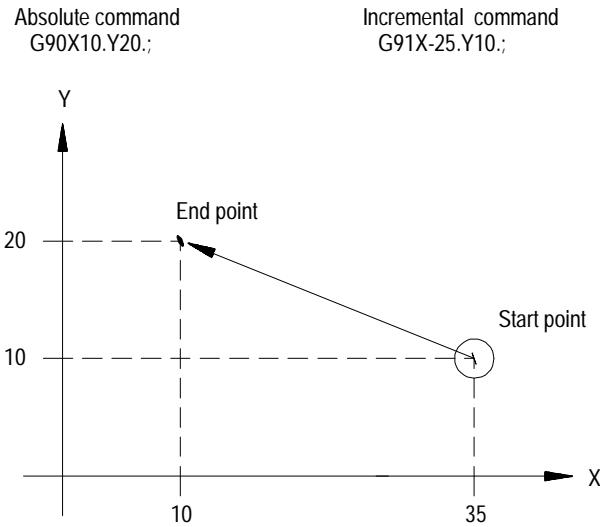
In the incremental mode, coordinates are referenced from the current axis position. Incremental mode is established by programming a G91.

```
G91X40. Z20. ;
```

In the above block the control will move the cutting tool a distance of 40 units on the X axis and 20 units on the Y axis away from the current axis position.

G91 is a modal G code and remains active until cancelled by a G90.

Figure 13.7
Incremental and Absolute Commands.



13.4 Inch/Metric Modes (G20, G21)

The selection of a unit system (inch or metric) can be done by programming either G20 for the inch system or G21 for the metric system. These unit system G codes should be among the first blocks written in a program.

Both G20 and G21 are modal, and cancel each other. The default unit system selected by the control at power-up is determined in AMP by the system installer.

The currently active unit system is usually displayed on the screen for softkey level 1 in lines 3 or 4 between the [] symbols. If the screen selected for display of softkey level 1 is the status screen the active system G code (G21 or G20) will be displayed among the active system G codes.

These functions are affected by the active unit system (inch or metric).

- Position commands
- Feedrate commands
- Axis feed amount for fixed amount feed operation
- Unit system for hand pulse generator (HPG)

13.5 Scaling

Use the Scaling feature to reduce or enlarge a programmed shape. Enable this feature by programming a G14.1 block as shown below:

G14.1 X_Y_Z_P_;

Where :	Is :
X, Y, Z	the axis or axes to be scaled and the center of scaling for those axes
P	the scaling magnification factor for the specified axes.

The axes programmed in the G14.1 block determine which axes will be scaled. The corresponding axis word values specify the center of scaling for each axis. This position is the axis position around which the scaling operation is performed.

The scaling magnification factor (P) is the amount of scaling to apply to the programmed axes. Each scaled axis may have a different scale factor by programming them in separate G14.1 blocks. The scaling range is from 0.00001 to 999.99999. A scale factor less than one will reduce a programmed move while a scale factor greater than one will enlarge a programmed move.

If no P word is programmed or if P0 is programmed in the G14.1 block, the default magnification factor is used. If the programmed P word value is out of range, an error message will be displayed on the CRT.

When absolute mode (G90) is active, scaling moves are referenced from the programmed center of scaling.

Example 13.5
Scaling with Absolute Mode Active

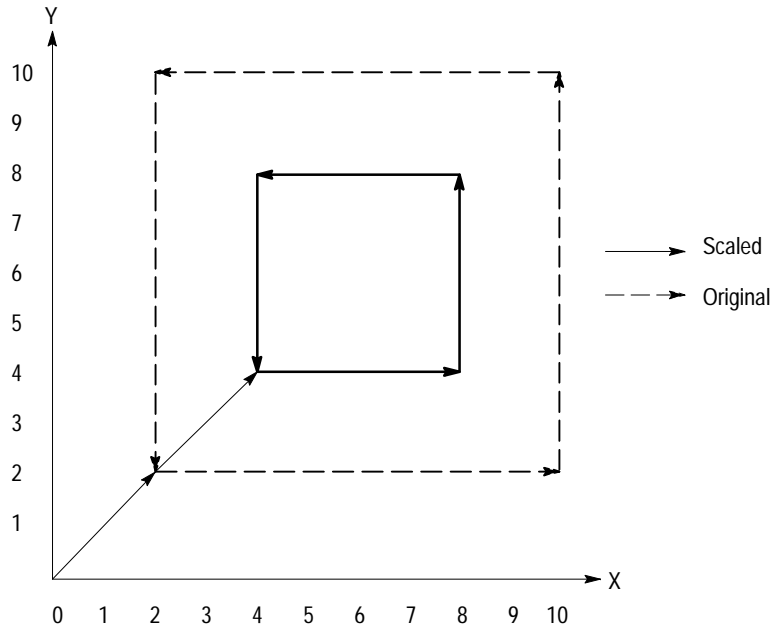
Program

```

N01 G14.1 X6 Y6 P0.5;
N02 G90 X2 Y2 F100;
N03 X10;
N04 Y10;
N05 X2;
N06 Y2;
N07 M30;

```

Figure 13.8
Results of Example 13.5



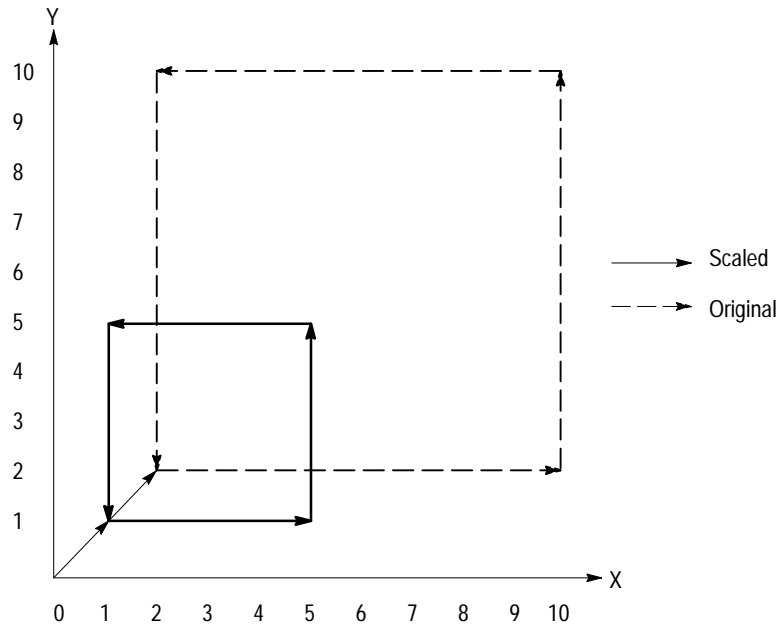
When incremental mode (G91) is active, the control ignores the programmed centers of scaling. The control performs scaling on the axes programmed in the G14.1 block, but the scaling moves are referenced from their current axis positions not the programmed center of scaling or the active coordinate zero point.

Important: The center of scaling may be specified in either incremental or absolute mode (G90/G91) in the G14.1 block. But, unlike other features in the control, both modes may not be programmed in the same block.

Example 13.6
Scaling with Incremental Mode Active

Program block	Comment
N01 G91;	incremental mode
N02 G14.1 X6. Y6. P0.5;	scale X and Y by .5 (X, Y values ignored)
N03 G91 X2 Y2 F100;	axis positioning move
N04 X8. ;	feedrate move X
N05 Y8. ;	feedrate move Y
N06 X-8. ;	feedrate move X
N07 Y-8. ;	feedrate move Y
N08 G14;	cancel scaling

Figure 13.9
Results of Example 13.6



G14 disables scaling on all axes. When scaling is disabled, the center of scaling and any scaling magnification factors are cleared. The next time scaling is enabled these values must be reset. In addition to G14, M99 in the main program, M02, M30, and a control reset operation will disable scaling. The system will power up with scaling disabled.

The system installer specifies in AMP, on an axis by axis basis, whether scaling is allowed. Refer to the literature provided by your system installer for additional information.

The control provides the PAL program with the option of monitoring which axes are currently being scaled, on an axis by axis basis, through the PAL flag \$SCAX. Refer to either to your PAL reference manual for additional information.

13.5.1 Scaling and Axis Position Display Screens

When scaling is enabled for a particular axis, the letter “P” will be displayed next to the axis name on all axis position display screens. The following screen shows scaling enabled on all axes.

Figure 13.10
Axis Position Display Screen Showing Scaling Enabled

PROGRAM[MM]			E-STOP		
			F	0.000 MPM	
P	X	1234.567	S	00	
P	Y	9876.000	T	0	
P	Z	2468.000	(ACTIVE PROGRAM NAME)		
MEMORY			MAN STOP		
PROGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPPORT	→

13.5.2 Scaling Magnification Data Screen

The scaling magnification data screen lists the currently active scaling magnification amount, the current center of scaling, and the default scaling magnification amount for all axes. The currently active scaling magnification amount and the current center of scaling for the axes can be monitored through this screen. The default scaling magnification amount for the axes can be monitored and/or changed through this screen.

The default scaling magnification values should only be changed when the control is in a stopped state. If the default values are changed, the new default values will not become active until the next G14.1 block is executed.

The scaling magnification data screen is accessed through these steps:

1. Press the {OFFSET} softkey on the main menu screen.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the {SCALNG} softkey to display the scaling magnification data screen.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

Figure 13.11
Scaling Magnification Data Screen

SCALING MAGNIFICATION			
	CENTER	CURRENT	DEFAULT
X	• <u>1</u> 23.45678	2.00000	1.00000
Y	• <u>1</u> 23.45678	2.00000	1.00000
Z	• <u>1</u> 23.45678	2.00000	1.00000
A	• <u>4</u> 56.78901	0.50000	1.00000
B	• <u>4</u> 56.78901	0.50000	1.00000
C	• <u>4</u> 56.78901	0.50000	1.00000

↑	REPLCE VALUE					
---	-----------------	--	--	--	--	--

Important: If an axis is configured as a rotary axis, the scaling magnification display screen will display dashes instead of numbers for that axis. Rotary axes cannot be scaled.

The left column lists the current center of scaling for each axis. When scaling is cancelled, the current center of scaling for each axis is set to zero. The format of this value is determined by the word format of the selected axis.

The middle column lists the currently active scaling magnification value for each axis. When scaling is cancelled, the current scaling magnification value for each axis is set to 1.00000.

The right column lists the current scaling magnification default value for each axis. This value is used if P is not programmed or if P0 is programmed in the G14.1 block. The range of the default value is 0.00001 to 999.99999 with a word format of 3.5. The default values are stored in RAM or flash memory when the control is powered down. When the control is powered up, these values are restored.

3. Use the up or down cursor keys to move the block cursor to the default value to be changed. The selected default value will be shown in reverse video.
4. To replace stored default scaling magnification value, key-in the new default value and press the {REPLCE VALUE} softkey.

13.5.3 Scaling Restrictions

While scaling is enabled, these restrictions apply:

- Scaling affects only programmed axis motion. All manual axis motions and PAL axis mover motions are performed at full scale
- Scaling does not affect M, F, S, T, and B word functions. The F word will be scaled if the control is in inverse time mode (G93). Scaling while in inverse time mode will be applied as follows:

$$\text{Scaled F word (when in G93 mode)} = \frac{\text{Programmed F word}}{\text{Largest Scale Factor}}$$

- Scaling is disabled during G27, G28, and G30 automatic home operations. For a G29 automatic return from home operation, scaling is re-enabled after the intermediate point is reached
- When changing work coordinates (G54-G59.3), the center of scaling is transferred from the old work coordinate system to the new work coordinate system. The offset distance from the tool position in the old work coordinate system to the tool position in the new work coordinate system is not scaled

- Scaling is applied to G52 and G92 offsets. The center of scaling will be shifted when the work coordinate systems are shifted by a G92 offset or by changing coordinate offset values. When using a G52 offset, the center of scaling will be adjusted to the new local coordinate system
- Scaling is not applied to external offsets, tool wear, tool geometry, tool radius, or tool length offsets.
- Scaling will not be applied to blocks containing dwells (G04), data setting codes (G10, G10.1), or macro calls (G56, G66, G66.1). In the case of macro calls, the data passed via local parameters will not be scaled unless the data is used inside of the macro for motion.
- G22, programmable zone 2 check on and data setting, will not be scaled
- G53, absolute positions moves, will not scaled
- Rotary axes cannot be scaled
- Polar coordinates are not scaled
- In circular mode, the scale factors for the axes of the active plane have to be the same. The control generates an error if the scale factors of the axes are not equal
- Scaling will be applied to these fixed cycles as shown below:

- G31, G31.1 - G31.4

Gxx X__Y__Z__
 X (scaled)
 Y (scaled)
 Z (scaled)

- G37, G37.1 - G37.4

Gxx Z__
 Z (scaled)

- G73, G74, G76, G82, G83, G84G85, G86, G87, G88, G89

- Gxx X__Y__Z__R__I__Q__K__P__F__L__
 X (scaled)
 Y (scaled)
 Z (scaled)
 R (scaled)
 I (not scaled)
 Q (not scaled)
 K (not scaled)
 P (not scaled)
 F (not scaled)
 L (not scaled)

Important: R uses the scale factor associated with the axis that is perpendicular to the active plane

- G38

G38 H _ R _ D _ E _ F _
 H (scaled)
 R (scaled)
 D (scaled)
 E (not scaled)
 F (not scaled)

Important: The active plane scale factors must be equal. H, R, and D use the scale factor associated with the active plane

- G38.1

G38.1 X _ Y _ I _ J _ R _ D _ E _ F _
 X, Y (scaled)
 I (scaled)
 J (scaled)
 R (scaled)
 D (scaled)
 E (not scaled)
 F (not scaled)

Important: The active plane scale factors must be equal. R and D use the scale factor associated with the active plane

- G88.1, G88.2

G88.x X _ Y _ Z _ I _ J _ (,R or,C) _ P _ H _ D _ L _ E _ F _
 X, Y (scaled)
 Z (scaled)
 I, J (scaled)
 ,R ,C (scaled)
 P (not scaled)
 H (not scaled)
 D (scaled when scale factor is less than 1)
 (not scaled when scale factor is greater than or equal to 1)
 L (scaled when scale factor is less than 1)
 (not scaled when scale factor is greater than or equal 1)
 E (not scaled)
 F (not scaled)

Important: The active plane scale factors must be equal. ,R and ,C use the scale factor associated with the active plane

- G88.3, G88.4

G88.x X_Y_Z_I_J_Q_(,R or,C)_P_H_D_L_E_F_
X, Y (scaled)
Z (scaled)
I, J (scaled)
Q (scaled)
,R ,C (scaled)
P (not scaled)
H (not scaled)
D (scaled when scale factor is less than 1)
(not scaled when scale factor is greater than or equal to 1)
L (scaled when scale factor is less than 1)
(not scaled when scale factor is greater than or equal 1)
E (not scaled)
F (not scaled)

Important: The active plane scale factors must be equal. Q, ,R, and ,C use the scale factor associated with the active plane:

- G88.5, G88.6

G88.x X_Y_Z_R_Q_P_H_D_L_E_F_
X, Y (scaled)
Z (scaled)
R (scaled)
Q (not scaled)
P (not scaled)
H (not scaled)
D (scaled when scale factor is less than 1)
(not scaled when scale factor is greater than or equal to 1)
L (scaled)
E (not scaled)
F (not scaled)

Important: The Irregular Pocket Milling Cycles feature (G89.1 and G89.2) is only available prior to system software release 12.xx. Any attempt to program a G89.1 or G89.2 in release 12.xx or later will result in the error message, “Illegal G-code”.

Important: The active plane scale factors must be equal. R uses the scale factor associated with the active plane. L uses the scale factor associated with the axis that is perpendicular to the active plane:

- G89.1, G89.2

G89.x X_Y_Z_P_Q_H_E_F_L_

X, Y (scaled)

Z (scaled)

Q (not scaled)

P (not scaled)

H (not scaled)

E (not scaled)

F (not scaled)

L (not scaled)

END OF CHAPTER

Axis Motion

14.0 Chapter Overview

This chapter describes the group of G-words that generates axis motion or dwell data blocks. Major topics include:

Information about:	On page:
Positioning axes	14-1
Polar coordinate programming	14-21
Automatic machine home	14-29
Dwell (G04)	14-35
Programmable mirror image	14-36
Axis clamp	14-39
Feed to Hard Stop	14-40

14.1 Positioning Axes

Use these 4 basic G-codes to produce axis motion:

- G00 Positioning
- G01 Linear interpolation
- G02 Circular/Helical interpolation (clockwise)
- G03 Circular/Helical interpolation (counter-clockwise)

After the execution of a positioning command, the program proceeds to the next block only after an in-position check function confirms that all commanded axes have reached the in-position band. The system installer sets the in-position band width in AMP.

14.1.1 Rapid Positioning Mode (G00)

The format for the rapid positioning mode is as follows:

```
G00X__ Y__ Z__ ;
```

The G00 code establishes the positioning mode. In positioning mode, the cutting tool is fed along a straight line to a location designated by the programmed axis words.

The axes to be moved are determined by the axis names in the G00 block. The end point of the move to be generated is determined by the values programmed with the axis names.

Rapid positioning can be performed in the absolute mode (G90) or the incremental mode (G91).

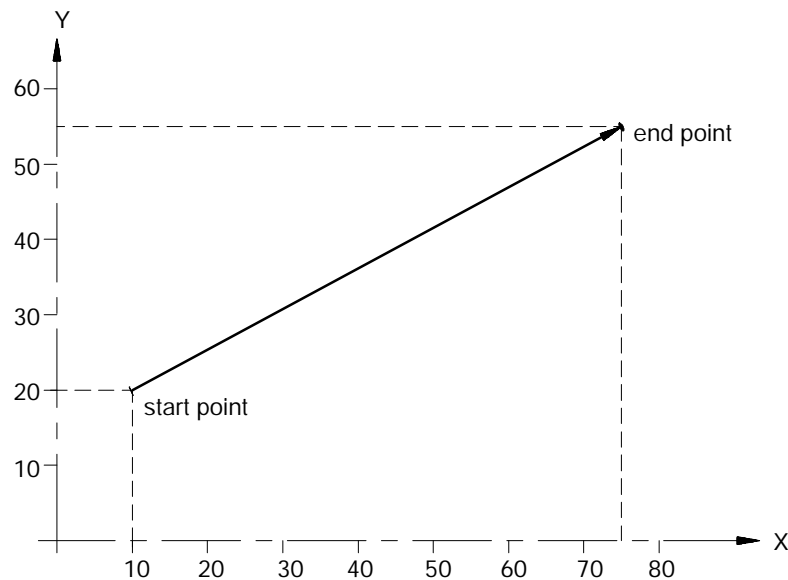
The system installer specifies a rapid feedrate individually for each axis in AMP. The feedrate of a positioning move that drives more than one axis is limited by the rapid rate set for the slower axis. The slower axis is driven at its rapid rate while the feedrate for other axes is reduced to maintain a linear move. This also assures that all axes start and stop at the same time.

G00 is a modal command and remains in effect until cancelled by a G-code of the same group. For a listing of G-code groups see the table in section 10.5.5.

Example 14.1
Positioning (G00)

Absolute Command	Incremental Command
G90G00X75. Y55. ;	G91G00X65. Y35. ;
M30;	M30;

Figure 14.1
Results of G00 Positioning Example



Important: Any F-word designated in the positioning mode is stored as the active feedrate in control memory but is ignored during positioning mode (G00).

14.1.2 Linear Interpolation Mode (G01)

The format for the linear interpolation mode is as follows:

```
G01X__ Y__ Z__ F__ ;
```

G01 establishes the linear interpolation mode. In linear interpolation mode, the cutting tool is fed along a straight line at the currently active or programmed feedrate.

The axes to be moved are determined by the axis names in the G01 block. The end point of the move to be generated is determined by the values programmed with the axis names.

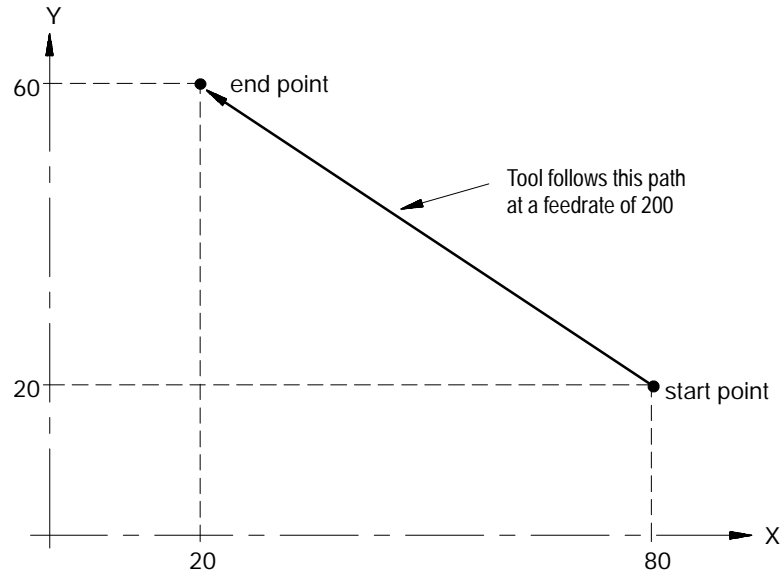
The F-word represents the feedrate for axis moves that take place in the G01, G02 and G03 modes. The F-word does not have to be programmed in the G01 block, however, if not programmed a feedrate must have been made active in some previous block.

Linear interpolation can be performed in the absolute mode (G90), or the incremental mode (G91).

Example 14.2 Absolute Versus Incremental Interpolation

Absolute Command	Incremental Command
G90G01X20. Y60. F200;	G91G01X- 60. Y40. F200;
M30;	M30;

Figure 14.2
Results of Linear Interpolation (G01) Example



Once the feedrate, F, is programmed it remains effective until another feedrate is programmed (F is modal). It is possible to override programmed F-words. For information on overriding feedrates, see chapter 18.

Example 14.3
Modal Feedrates

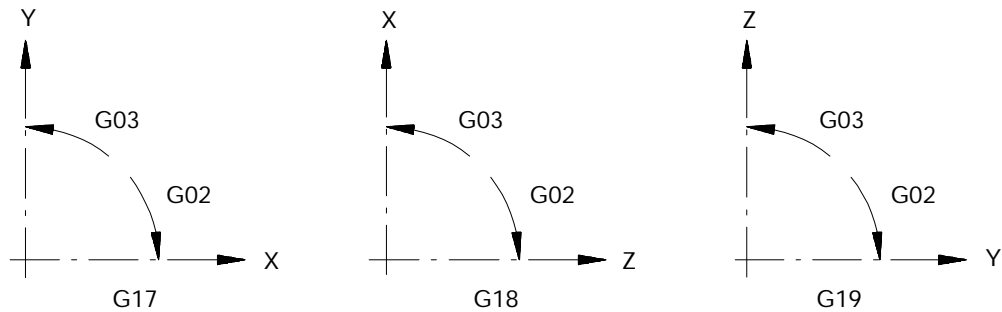
Program Block	Comment
G91G01X10. Y20. F200;	F200 is effective until another feedrate is programmed
Y35. ;	
X40. Y35. ;	
Y44. F50;	F50 is effective
M30;	

The feedrate for a multi-axis move is specified as the vectorial feedrate. The control will adjust the individual axis feeds to obtain the programmed feedrate. For information on feedrates, see chapter 18.

14.1.3 Circular Interpolation Mode (G02, G03)

G02 and G03 establish the circular interpolation mode. In G02 mode, the cutting tool moves along a clockwise arc; in G03 the tool moves along a counterclockwise arc. Figure 14.3 shows clockwise and counterclockwise orientation relative to the positive X, Y, and Z axes.

Figure 14.3
Circular Interpolation Direction



Circular interpolation can be performed in the absolute (G90) or incremental (G91) mode.

Important: S-Curve Acc/Dec mode is not available with circular interpolation mode.

A plane must first be established before the control will perform the correct arc.

The system installer selects a default plane that the control assumes when power is turned on, E-STOP is reset, or when the control is reset. In order to change planes, it is necessary to program either G17, G18, or G19. G17, G18, and G19 are modal and remain in effect until cancelled by each other. For details on plane selection, see chapter 13.

The system installer determines which axes are assigned to each plane in AMP. This manual assumes the axes are assigned to the planes as indicated below:

Circular Interpolation in XY plane

$$\begin{array}{c} G17\{G02\} X_ Y_ \{I_ J_ \} F_ ; \\ G03 \qquad \qquad \qquad R_ \end{array}$$

Circular Interpolation in ZX plane

$$\begin{array}{c} G18\{G02\} Z_ X_ \{K_ I_ \} F_ ; \\ G03 \qquad \qquad \qquad R_ \end{array}$$

Circular Interpolation in YZ plane

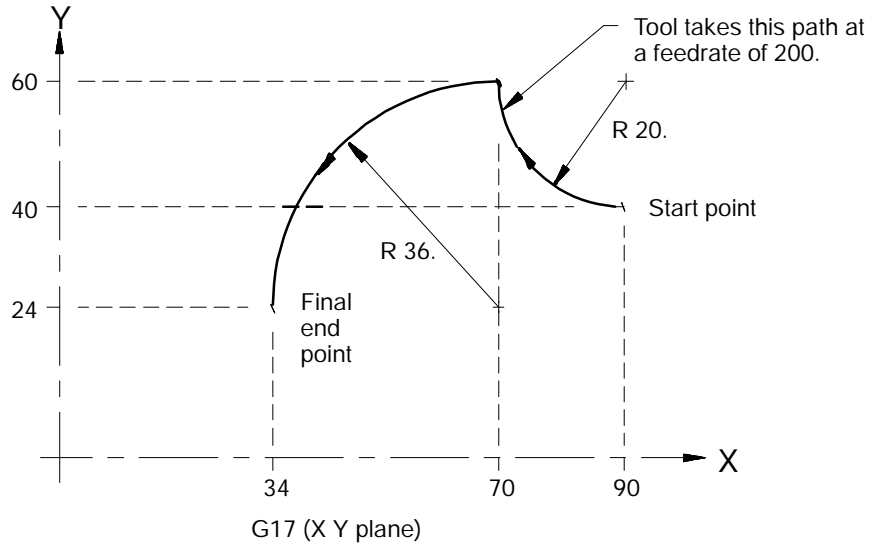
$$\begin{array}{c} G19\{G02\} Y_ Z_ \{J_ K_ \} F_ ; \\ G03 \qquad \qquad \qquad R_ \end{array}$$

Where :	Is :
X, Y, Z	In absolute (G90) mode, these are the coordinate values of the end-point. In incremental (G91) mode, these are the positions of the end-point in reference to the start-point CAUTION: If no axis word is specified for the end-point of one of the axes in the currently defined plane, the control will assume a value for that axis equal to the current tool location. Specifying the wrong plane or wrong axis word for a circular move can inadvertently generate a helical move (see section 14.1.4).
I, J, K	These determine the position of the arc center in reference to the start-point. These values are always incremental regardless of the established positioning mode (absolute or incremental). I is parallel to X axis, J is parallel to Y axis, and K is parallel to Z axis in this manual. Refer to the system installer's documentation for integrand words.
R	Rather than defining a center with I, J, K, the option exists to define an arc radius using R. The sign of this entry determines the arc center-point location. If R is programmed as a positive value, the center-point will be located such that an arc less than or equal to 180° is generated. If R is programmed as a negative value, the center-point will be located such that an arc greater than 180° is generated. Refer to Figure 14.5 for an example.
F	Another option is to enter a feedrate tangent to the arc. If omitted, the control will use the feedrate active prior to this block.

Example 14.4
Circular Interpolation

Absolute Mode	Incremental Mode
G17;	G17;
G00X90Y40;	G91G02X-20.Y20.J20.F200;
G02X70.Y60.J20.F200;	G03X-36.Y-36.J-36.;
G03X34.Y24.J-36.;	M30;
M30;	
or	or
G17;	G17;
G00X90Y40;	G91G02X-20.Y20.R20.F200;
G90G02X70.Y60.R20.F200;	G03X-36.Y-36.R36.;
G03X34.Y24.R36.;	M30;
M30;	

Figure 14.4
Results of Circular Interpolation Example

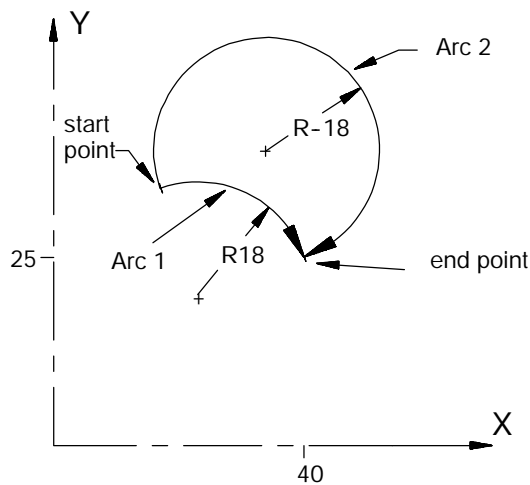


When programming an arc using the radius (R) value, two arcs are possible (Figure 14.5). Program the R word with a positive or negative value to distinguish between these arcs.

Example 14.5
Arc Programmed Using + or - Radius

Arc 1 center angle less than 180 degrees	Arc 2 center angle greater than 180 degrees
G00X15Y30;	G00X15Y30;
G90G02X40. Y25. R18. F200;	G90G02X40. Y25. R- 18. F200;
M30;	M30;

Figure 14.5
Results of Arc Programmed Using Radius Example



If the end point of the arc is not specified, or if the end point is the same as the start point, do not use R. Only J, I, and K can specify the center point in these cases.

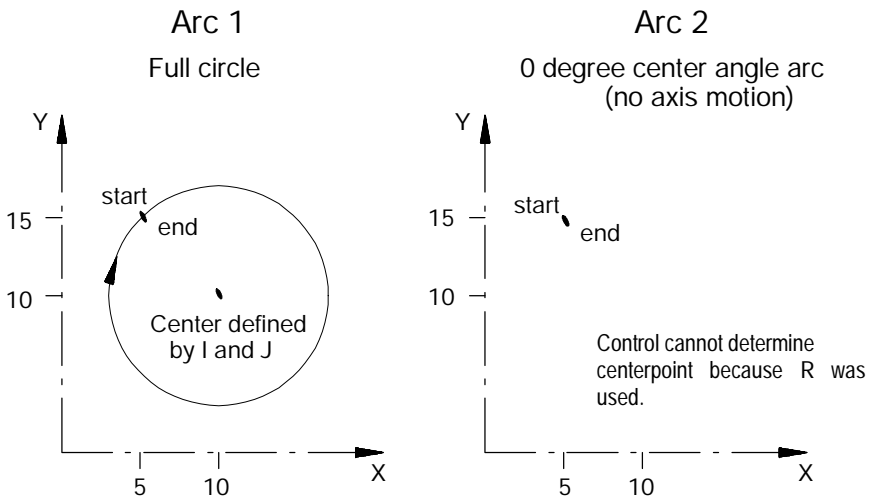
Important: Any axis in the current plane that is not specified when programming a circle defaults to the current axis position values. This results in the end point of an arc having the same coordinate value as the start point of the arc for that axis.

- If I, J, and/or K is used to program the arc center the control will cut a full circle.
- If R is used to program the radius of the arc the control will not move the axis. This is because the control defines an arc with a 0 degree center angle.

Example 14.6
Arc End Points Same As Start Points

Arc 1 - Full Circle	Arc 2 - No Motion
G00X5. Y15;	G00X5. Y15;
G02X5. Y15. I5. J-5. F100;	G02X5. Y15. R7. 07. F100;
M30;	M30;

Figure 14.6
Arc with End Point Equal To Start Point



If programming a radius command, R, in the same block as I, J, and/or K, the control gives the R priority. The I, J, and/or K words are then ignored.

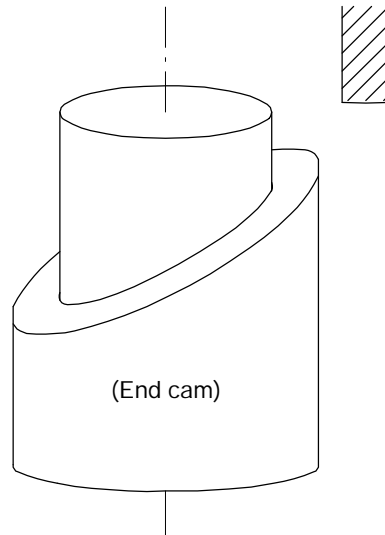
When programming I, J, and/or K words for the arc center, the words that have a zero value can be omitted.

Important: The system installer can specify the maximum allowed difference between the starting radius of the arc and the ending radius of the arc. If the difference exceeds the allowed value set in AMP, an error message occurs.

14.1.4
Helical Interpolation Mode
(G02, G03)

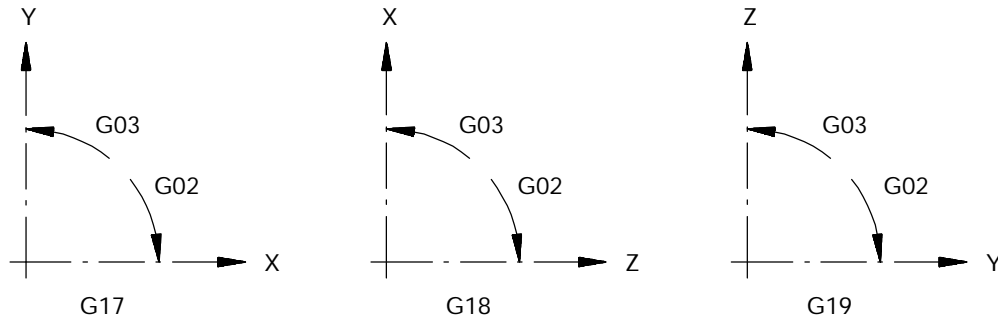
G02 or G03 may also be used to perform helical interpolation.
Figure 14.7 shows how a part may be cut with helical interpolation.

Figure 14.7
Helical Interpolation (G02, G03)



Use G02 or G03 to add a third axis to the circular interpolation command block. The direction of the helical interpolation that results depends on whether a G02 or G03 was used. Refer to Figure 14.8.

Figure 14.8
Helical Interpolation Direction



Helical Interpolation in the XY Plane with the Z axis normal.

G17{G02} X__ Y__ Z__ {I__ J__} F__ ;
G03 R__

Helical Interpolation in the XZ Plane with the Y axis normal.

G18{G02} X__ Z__ Y__ {I__ K__} F__ ;
G03 R__

Helical Interpolation in the YZ Plane with the X axis normal.

G19{G02} Y__ Z__ X__ {J__ K__} F__ ;
G03 R__

Where :	Is :
X, Y, Z	In absolute (G90) mode these are the coordinate values of the end point. In incremental (G91) mode these are the positions of the end point in reference to the start point The axis which is normal to the circular interpolation plane produces the "lead" of the helix. Again, all axes start and stop at the same time to produce helix motion.
I, J, K	These determine the position of the helix center in reference to the start point. These values are always incremental, regardless of the established positioning mode (absolute or incremental)
R	Rather than defining a center with I, J, K, the option exists to define an arc radius using R. The sign of this entry determines the arc centerpoint location. If R is programmed as a positive value, the centerpoint will be located such that an arc less than 180° is generated. If R is programmed as a negative value, the centerpoint will be located such that an arc greater than 180° is generated. Refer to Figure 14.5 for an example.
F	Another option is to enter a feedrate tangent to the tool path. If omitted the control will use the feedrate active prior to this block.

In helical interpolation, the feedrate is the same as in circular interpolation, that is, the feedrate is tangent to the tool path.

Important: Cutter diameter compensation is effective only for the arc portion of helical interpolation. Tool length offsets may be active during a helical move, however, changes to the tool length offset are allowed only if it does not affect either of the two circular axes in the move.

14.1.5 Positioning Rotary Axes

A rotary axis is a non-linear axis that typically rotates about a fixed point. A rotary axis is not the same as a spindle which uses an M19 to orient to a specific angle. A rotary axis is a fully positionable axis that is capable of interpolated motion when programmed in a block with other axes.

The system installer determines which axes are rotary axes in AMP, and determines the address that is used to command those axes. This manual assumes that the C-word is used to program a rotary axis. Refer to the system installer's documentation for the rotary axis words used in a specific system.

A rotary axis is programmed in units of degrees. This manual assumes that the system installer has configured the rotary axis to "rollover" at 359.99 degrees. Rollover means that after the rotary axis exceeds 359.99 degrees of rotation, its position displays rollover to 0 degrees and starts increasing. If the axis rotates to a position less than 0 degrees its position displays rollover to 359.99 degrees and starts decreasing.

Typically a rotary axis is programmed in a block by itself or with linear moves (rapid G00 or cutting G01 moves). If necessary it is possible, however, to program a rotary axis in a block that contains circular moves (G02 or G03).

Programming in absolute or incremental

Rotary axes may be programmed in absolute or incremental mode.

In absolute mode (G90) the rotary axis is programmed to angular positions. These positions are programmed between 0 and 359.99 degrees. The sign given to this angular position determines the direction that the rotary axis will travel to reach the programmed angle. For example programming:

```
G90C25;
```

in a part program will cause the rotary axis C, to rotate to an angle of 25 degrees (referenced from a position 0 determined by the system installer). It will rotate in the positive direction to reach this position. Programming:

```
G90C-25;
```

in a part program will cause the rotary axis C, to rotate to the same position (25 degrees) except that it will rotate in the negative direction to reach this position.

In incremental mode (G91) the rotary axis is programmed to move an angular distance (not to a specified angle as in absolute). The maximum incremental departure depends on the programming format selected in AMP by the system installer. The sign of the angle determines the direction the rotary axis will rotate. For example if the current C axis position is 25 degrees and the following block is programmed:

```
G91C50;
```

the C axis would rotate 50 degrees in the positive direction. The new C axis position would be 75 degrees. If the current C axis position is 25 degrees and the following block is programmed:

```
G91C-50;
```

the C axis would rotate 50 degrees in the negative direction. The new C axis position would be 335 degrees.

In incremental mode (G91), programming a value greater than the rollover amount results in the rotary axis making one or more complete revolutions.

In absolute mode (G90), you cannot program a rotary axis move greater than the rollover amount.

In circular interpolation mode (G02 or G03), you cannot program a rotary axis move unless these conditions are met:

- the rotary axis cannot be in the active plane
- the rotary axis must be programmed in the same block as a valid circular move made with the axes in the active plane

Important: The largest move that may be programmed with a rotary axis is equal to the rollover amount. Any attempt to program a move that will generate more motion than the rollover amount, will be truncated and moved to the position that has the same numerical endpoint as the programmed position. For example if the following incremental move is programmed from a position of 10 degrees:

```
G91C370;
```

the actual endpoint of the above move will still be 20 degrees; however, the rotary axis will not get there by revolving one revolution. Instead, it will position itself directly to 20 degrees without passing 20 once, as would be expected.

Determining Rotary axis feedrates

The feedrate for a rotary axis is determined in much the same way as linear axes.

When the control is in rapid mode (G00) the feedrate for the rotary axis is the rapid feedrate for that axis as set in AMP. Remember that if other axes are moving in the same block the feedrate for the block is limited by the axis which will take the longest time to complete its programmed move at its rapid speed (see chapter 18 for details).

When the control is in one of the cutting modes (G01, G02, or G03) the control uses the programmed feedrate to calculate the angular velocity of the rotary axis. This feedrate is still limited to the maximum cutting feedrate (feedrate clamp) as determined in AMP.

When programming in G95 feed per revolution mode the rotary axis will perform the required amount of angular rotation for every revolution of the spindle. If programmed with other axis motions in G93 or G94 modes. The resulting feedrate for the block is applied to the total distance to move the axes including the rotary axis

In any event if a rotary axis is programmed in a block with other axis moves in either rapid (G00) or cutting (G01, G02, or G03) modes, all axes will reach their destinations at the same instant.



CAUTION: Keep in mind when programming a rotary axis that the programmed feedrate is in units of angular velocity. This means that the actual cutting feedrate will be dependant on the tool's distance from the center of rotation of the rotary axis.

14.1.6 Cylindrical Interpolation

The cylindrical interpolation feature coordinates the motion of a rotary axis with the linear machine axes to machine contours on the side of a cylindrical workpiece as shown in Figure 14.9. Cylindrical interpolation mode is turned on using a G16.1 block and turned off with a G15 block.

A mill control with a minimum of two linear and one rotary axes is required for cylindrical interpolation. Typically there will be three linear and one rotary axis.

Cylindrical interpolation requires that axes be defined in AMP as a cylindrical interpolation rotary axis, a linear axis, a park axis, and a feed axis. The coordinates of the park and feed axes that define the rotary axis center-line must also be specified. Refer to the information provided by your system installer.

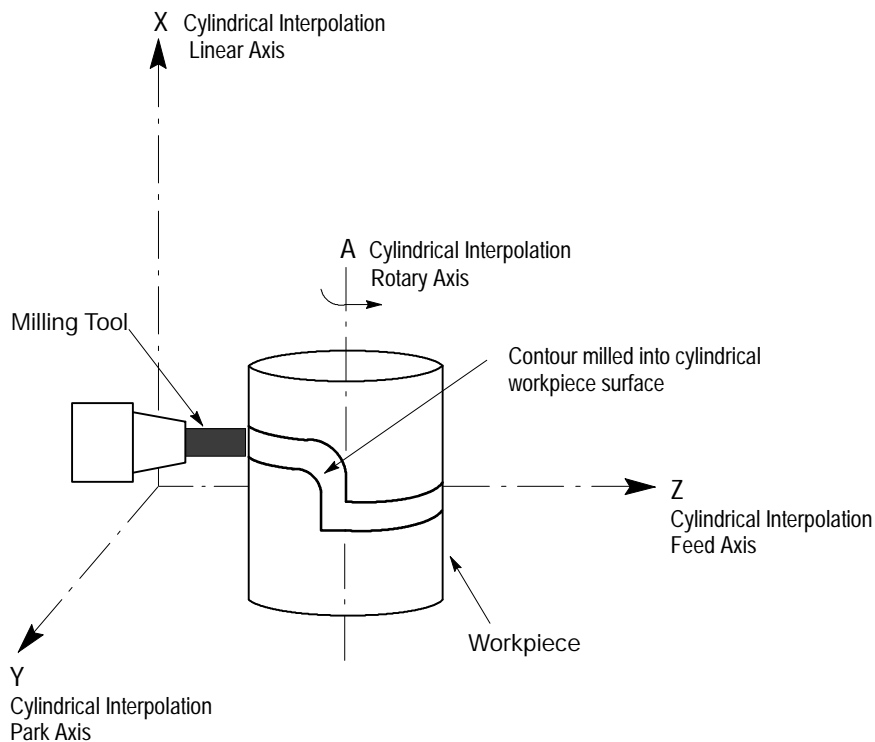
Important: Cylindrical interpolation requires that the cylindrical interpolation rotary axis rollover value be 360 degrees.

This discussion assumes the following AMP axis name assignments. Refer to the literature provided by your system installer for the axis names used by your machine.

- Feed axis is Z
- Park axis is Y
- Linear axis is X
- Rotary axis is A

Figure 14.9 shows a typical mill configuration for cylindrical interpolation.

Figure 14.9
Typical Mill Configuration for Cylindrical Interpolation



Important: The center of the rotary axis must coincide with the center-line of the workpiece on which contours are machined during cylindrical interpolation.

Cylindrical Interpolation Block Format

The block used to activate cylindrical interpolation has the following format:

G16.1 R__ X__ Z__ A__ F__

Where :	Is :
R	The radius at which the feed axis (typically the Z axis) will be positioned at the start of cylindrical interpolation. Can be used to alter the feed axis depth if programmed in a G16.1 block <u>during</u> cylindrical interpolation.
X	The coordinate (if in G90 absolute mode) or the linear distance (if in G91 incremental mode) to which the X axis is to move.
Z	The coordinate (if in G90 absolute mode) or the linear distance (if in G91 incremental mode) to which the Z axis (feed axis) is to move.
A	The angular coordinate (if in G90 absolute mode) or the angular distance (if in G91 incremental mode) to which the A rotary axis is to move.
F	The feedrate to be used by the X and Z axes when commanded to move while G16.1 is active. It also controls the A rotary axis speed as. Refer to chapter 18.

These parameters and their application are described in detail in the paragraphs that follow:

Important: R must be programmed in the initial G16.1 block. If R is not programmed in the initial G16.1 block, the error message “CYLINDER RADIUS IS ZERO” will appear. At power turn-on, program-end (M02, M30, or M99) or control reset the cylindrical interpolation feature is turned off and the R value is set to zero. It must then be re-entered in the next G16.1 block.

The radius specified by the R parameter is modal and does not need to be included in subsequent cylindrical interpolation blocks. Programming a G16.1 block with a different R value will modify the feed depth to the new radius. Feed depths cannot be changed using the Z parameter when G16.1 is active. Programming a Z will generate the error message “FEED AXIS MOTION NOT ALLOWED”.

Figure 14.9 illustrates the tool position if the AMP parameter Feed Axis Park Location is selected as “Nearest to Machine Zero”. If “Farthest from Machine Zero” were selected, then the tool would be positioned for cutting into the part from the positive side of the Z axis. Refer to the information provided by your system installer.

An A or X axis position may be programmed with the R parameter in the initial G16.1 block. However, once G16.1 mode is established, only the X parameter can be programmed in the same block as the R parameter. When it is, the X axis motion will be executed first followed by feed axis motion to radius R.

If an A axis position is programmed, the A axis will be rotated to the specified angle. If the A and X axes are programmed together in the same block, then a vector motion will result. around the circumference of the part.

If G02 or G03 circular interpolation is made active while in G16.1 cylindrical interpolation mode, a circular cut can be made around the circumference of the part (such as the contour cut in Figure 14.9). This is accomplished by programming the A and X axis endpoints along with the desired circle radius R as described in section 14.1.3 . Note that the R parameter now defines the radius of the circular path to be cut, not the feed axis position.

Important: When programming circular interpolation in G16.1 mode, only radius programming (using R) may be used. Integrand programming (using I, J, K) is not allowed and will generate the error message “CIRCLE PROGRAMMING ERROR”. Refer to section 14.1.3 .

Important: A axis motion is programmed as an angular value. When programming circular interpolation in G16.1 mode, this angular value has to be derived from an A axis arc length (based on the cutting radius). Refer to Example 14.7.

To perform G02/G03 circular interpolation while in G16.1 mode, the linear axis (X) and the rotary axis (A) must move to the endpoint of the arc of radius R made on the side of the cylinder.

In incremental mode (G91) the A axis arc length along with the programmed X move length, must position the A and X axes at a legal endpoint for the arc radius defined by the R value in the G02/G03 block.

In absolute mode (G90) the coordinate defined by the A axis arc along with the coordinate programmed for the X axis, must position the A and X axes at a legal endpoint for the arc radius defined by the R value in the G02/G03 block.

When cylindrical interpolation is activated, the circle plane is set to XA. The A and X axes become the two axes of the circle plane and remain so, as long as the G16.1 mode is active. If the active plane is changed, the change will not become effective until the G16.1 mode is cancelled, and will be superseded if the G16.1 plane is reactivated.

Canceling Cylindrical Interpolation

Cancel cylindrical interpolation by programming a G15. The G15 program block can not contain axis words. Note that the G15 program block can cancel other modal group 15 functions such as polar programming (refer to appendix C for a complete listing of modal group 15 G-codes).

Cylindrical Interpolation Operation

When cylindrical interpolation is activated, the control will position the tool on the cylindrical work surface with two distinct moves. In the first move, all programmed axis moves in the initial G16.1 block (including the A axis) will be executed. At the same time, the park axis (Y) is positioned to the park axis coordinate as specified in AMP (refer to the documentation provided by your system installer).

Once the tool is positioned at the AMP specified park coordinate, the control locks or “parks” the park axis at it’s current position. This prevents additional commands from moving the tool off the rotary axis center-line. This first move takes place at the rapid feedrate for the axes.

In the second move, the feed axis (Z) moves at the active cutting feedrate to the radius specified by R.

The blocks following the G16.1 block determine the contour to be machined on the side of the cylindrical workpiece. The moves of the rotary axis (A) and the machine axes are interpolated to produce the programmed contours.

The following example makes a series of circumferential and circular cuts into the side of a cylindrical workpiece. The A axis angle in the G02 block of this program was derived from the equation that follows this example. Figure 14.10 illustrates the results.

Example 14.7
Cylindrical Interpolation Example

Program Block	Comment
N01 G01 X0 Y0 Z0 F100;	
N02 G16.1 X100 R100;	set cylindrical cutting radius at 100
N03 G01 A40;	make circumferential cut of 40 degrees
N04 G02 X80 A51.459 R20;	make arc cut of radius 20
N05 G01 X60;	make linear cut of 60
N06 G16.1 R90;	change cylindrical cutting radius to 90
N07 G01 A160;	make circumferential cut of 160 degrees
N08 G01 X100 A270;	make linear and circumferential cut
N09 G01 A0	finish with circumferential cut
N10 G15;	cancel cylindrical interpolation
N11 M30;	

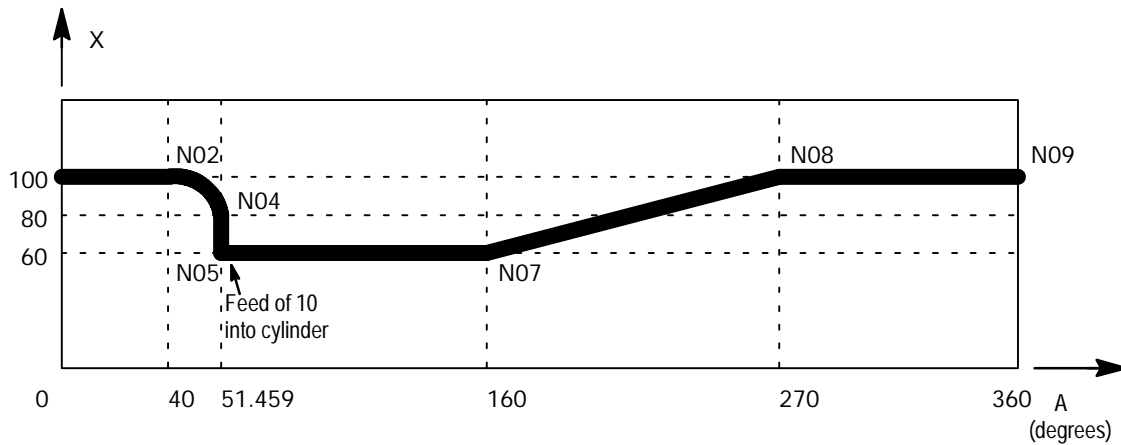
The angle for the A move in the G02 block above was determined using the following equation, with L = 20 and R = 100.

$$\bullet = \frac{360 (L)}{2 \bullet (R)}$$

Where :	Is :
•	The angle to be programmed for the A axis.
L	The length of the arc along the circumference of the cylinder, as required to define a legal endpoint for the arc programmed in the G02/G03 block.
R	The radius at which the feed axis is positioned. This is the active R value programmed in the initial G16.1 block, <u>not</u> the R radius for the G02/G03 block.

Figure 14.10 illustrates the results of Example 14.7 with a two-dimensional plot of the circumferential tool path.

Figure 14.10
Results of Cylindrical Interpolation, Example 14.7



Cylindrical Interpolation Programming Restrictions

When the cylindrical interpolation feature is enabled the following programming restrictions apply:

- Work coordinate system offsets (G52, G54-G59, and G92) for the park and feed axes (Y and Z) will be temporarily cancelled when in G16.1 mode. Offsets for other axes will not be affected
- Tool offsets and cutter compensation/TTRC offsets are allowed on the cylindrical circle plane during cylindrical interpolation programming
- Activation of offsets through PAL is disabled
- Jog on the fly is disabled
- Cylindrical interpolation cannot be activated during part rotation G88/G89
- Integrand circle/arc programming is not permitted during G02/G03 blocks. Only direct radius (R) programming is allowed
- Automatic motion to and from home G27, G28, G29, or G30 is not permitted.
- Work coordinate changes and shifts G53, G54-G59, G59.1, G59.2, G59.3, G50/G92, G52, G92.1 are not permitted
- Cavity and Irregular pocket cycles cannot be used
- Only the primary spindle (selected via G12.1) may be used in coordination with cylindrical interpolation. If the system uses Auxiliary Spindle(s), and the auxiliary spindle is the controlling spindle when cylindrical interpolation is selected, an error message appears.

14.1.7 PAL Axis Mover

If desired, the system installer has the option of controlling selected axes through the PAL program. When an axis is under PAL control, the operator and part program have no control on that axis. Jog commands as well as part program commands are typically ignored unless PAL is written to manipulate these values in some manner.

Be aware that it is possible to disable axis position displays on the CRT for an axis under PAL control. Refer to the documentation provided by your system installer for details on an axis controlled by PAL.

Important: S-Curve Acc/Dec mode is not available with PAL axis Mover.

14.2
Polar Coordinate
Programming (G15, G16)

Polar programming allows a programmer to use polar coordinates (using angles and distance specified with a radius) as a means of establishing the end point of a move rather than specifying the normal cartesian coordinates of the end point. G16 and G15 are modal G-codes used to start and stop polar coordinate programming respectively.

After a G16 block in a part program the control will interpret the axis words as polar programming commands. Cancel polar programming with a G15 block in a part program. The G15 program block can not contain any axis words. Note that the G15 program block can cancel other modal group 15 functions such as cylindrical interpolation (refer to appendix C for a complete listing of modal group 15 G-codes).

Axis words in the current plane (selected by G17, G18, and G19) are used to program angle and radius values. The order in which they are assigned to a plane in AMP is significant in determining their use.

Specifying the Radius:

The first axis word that is used to describe the current plane is used to specify radius values. Negative radius designations are measured 180 degrees from the current angle designation.

Specifying the Angle:

The second axis word that is used (to describe the current plane) is used to specify angle designations. The angle is specified in units of degrees. Positive angles are measured counter clockwise and negative angles are measured clockwise.

For example if the current plane is G17 (defined in AMP to be the X, Y plane) during polar programming:

Any X word in a program is used as a radius value for all following moves until re-specified using another X word or polar programming is cancelled with a G15 block.

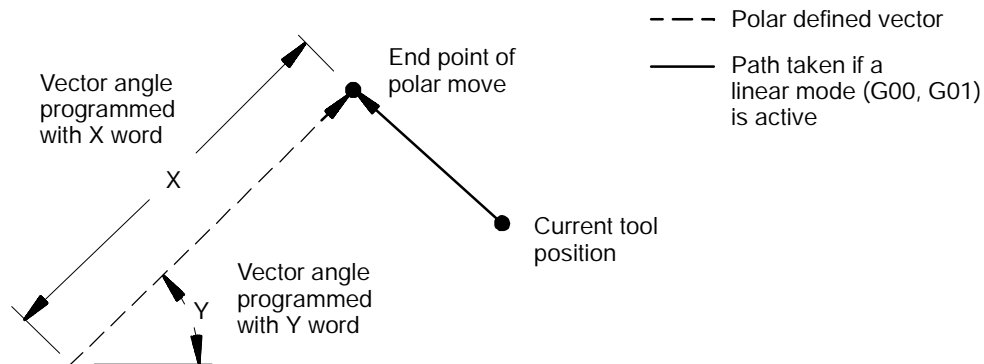
Any Y word in a program is interpreted to be the angle for all following moves until re-specified using another Y word or polar programming is cancelled with a G15 block.

For the purpose of explanation this section assumes that the X,Y plane (G17) is the currently active plane for all polar programming examples and figures. Any other axis word that is specified in a polar block and is not in the current plane is interpreted by the control as a normal cartesian coordinate value.

Polar positioning is done by defining a vector using a radius and angle value. The head (or end) of the vector defined by the radius and angle values is used as the end point of a polar move.

In both incremental and absolute mode the cutting tool will follow a path starting at the end point of the last move and ending at the head of the vector defined by the radius and angle. How the tool reaches that endpoint is determined by the current positioning mode (G00, G01, G02, or G03).

Figure 14.11
Polar Vector Defining End Point Of Tool Path



The control's interpretation of the specified angle and radius values used to define the vector is dependant on whether programming is in incremental mode or absolute mode. It is possible to mix incremental and absolute modes in polar programming, though for clarity this practice is not recommended.

If programming in incremental mode (G91):

- The radius is measured from the current tool position at the specified angle to define a vector.
- The angle is referenced relative to the last programmed angle (note the last programmed angle defaults to zero degrees upon entry to polar programming mode).

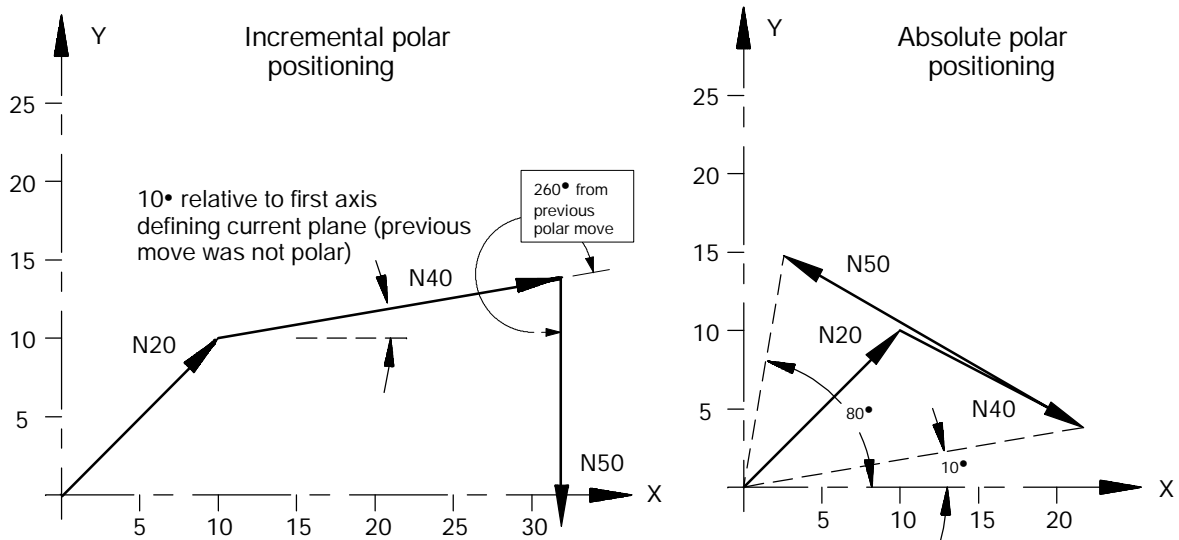
If programming in absolute mode (G90):

- The radius is measured from the zero point of the currently active work coordinate system at the specified angle and defines a vector. This vector is independent of the current tool position.
- The angle is referenced from the first axis that is used to define the currently active plane and is independent of the previous move.

Example 14.8
Incremental Versus Absolute Polar Blocks

Incremental	Absolute
N10 G00 X0 Y0 F150;	N10 G00 X0 Y0 F150;
N20 G91 G01 X10 Y10;	N20 G90;
N30 G16;	N30 G01 X10 Y10;
N40 X22 Y10;	N40 G16;
N50 X15 Y260;	N50 X22 Y10;
N60 G15;	N60 X15 Y80;
N70 M30;	N70 G15;
	N80 M30;

Figure 14.12
Results of Incremental Versus Absolute Polar Programming Example



Angles may be entered in a polar block with positive or negative values. Angles are referenced counter-clockwise if specified as positive and clockwise if negative. Clockwise and counterclockwise orientation for the X, Y, and Z axes is shown in Figure 14.3.

Angle values greater than 360 degrees are permitted. Programming 365 degrees or 725 degrees will have the same result as if 5 degrees were programmed.

Radius values may be programmed as positive or negative values. When specifying a radius value as a negative amount it is referenced in a direction 180 degrees from the currently specified angle.

The axis position displays will not show the polar coordinate values during polar programming. These displays will always show the current cartesian coordinate position as described in section 8.1

Important: Polar programming mode has no effect on axis words that are programmed with any of the following G-codes. Any axis words in any of these blocks are executed as if polar programming was not active. Axis words specified in these blocks have no effect on the current angle or radius active in polar programming.

- Coordinate system offset G52
- Coordinate system offset G92
- Work coordinate system rotation G68
- Dwell G04
- G10 blocks that modify tables
- Programmable zone blocks G22
- Programmable mirror image G50.1
- Motion in the machine coordinate system G53

Though polar programming has no effect on programming these G-codes, many of these G-codes have a significant impact on the execution of polar programming.

For example a G68 work coordinate system rotation will have a major impact on angles specified in a polar programming block. When the coordinate system is rotated the polar angle is referenced from the rotated axis. Work coordinate system offsets will have a similar effect on the work coordinate system zero point thus moving the location of the radius vector start point.

Axis words that are programmed with any of the return to/from home G-codes (G27 - G30) to specify an intermediate point, are interpreted as polar coordinate values when in polar programming mode.

14.2.1
Polar Programming Special
Cases

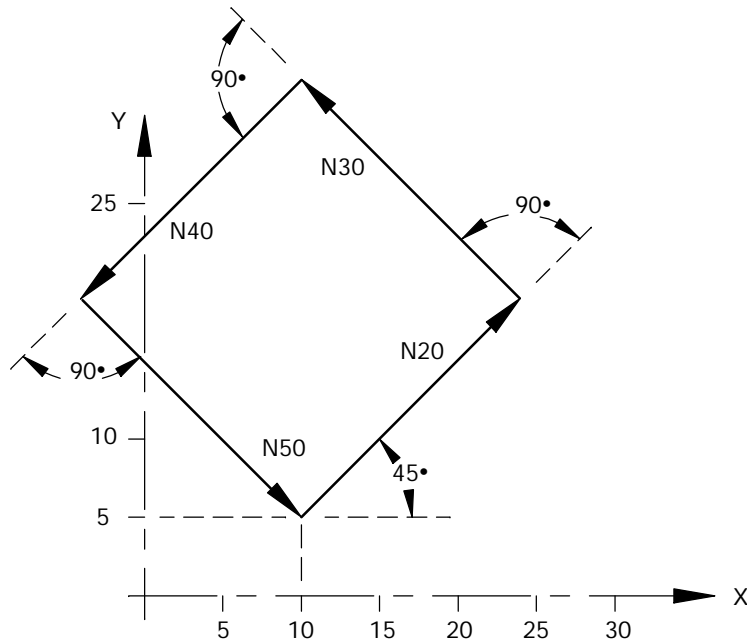
When programming using polar blocks the values programmed with the axis words are stored much as if they had been position commands. Normally, programming an incremental move of Y1.3 would position the Y axis 1.3 units from its previous position. The X axis position would not change. This also holds true for polar programming.

Programming Y20. with polar programming active specifies an angle of 20 degrees. This angle remains in effect for all subsequent blocks until a new value for Y is specified. This is also true for radius specifications. Keep in mind, however, that either a polar distance, a polar angle, or both must be programmed in the polar block.

Example 14.9
Polar Blocks with only Polar Angle Programmed

G00X10Y5;	G00X0Y0;
N10G00X10Y5;	G01G91G16F100;
N20G01G91G16F100;	N10G00X0Y0;
N30X20Y45;	N20G01G91G16F100;
N40Y90;	or
N50Y90;	N30X20Y45;
N60Y90;	N40Y90;
N70Y90;	N50X20;
	N60X20;
	N70M30;

Figure 14.13
Results of Example 14.9, Polar Programming Blocks



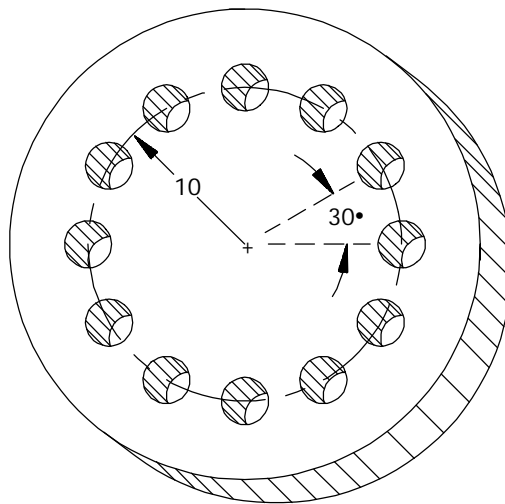
It is possible to change from incremental to absolute or absolute to incremental modes during polar programming if desired. The axis word is interpreted by the control in the mode that it was specified in. Mixed combinations such as angles designated in absolute and radii designated in incremental are possible.

Example 14.10 is used to illustrate this.

Example 14.10
Changing Between Incremental and Absolute During Polar Moves

```
N10G01X0Y0Z0F100;  
N20G16;  
G90X10. Y0. ;  
G81G91Y30. Z10. R5. L12;  
G15;  
M30;
```

Figure 14.14
Bolt Hole Pattern, Results of Example 14.10



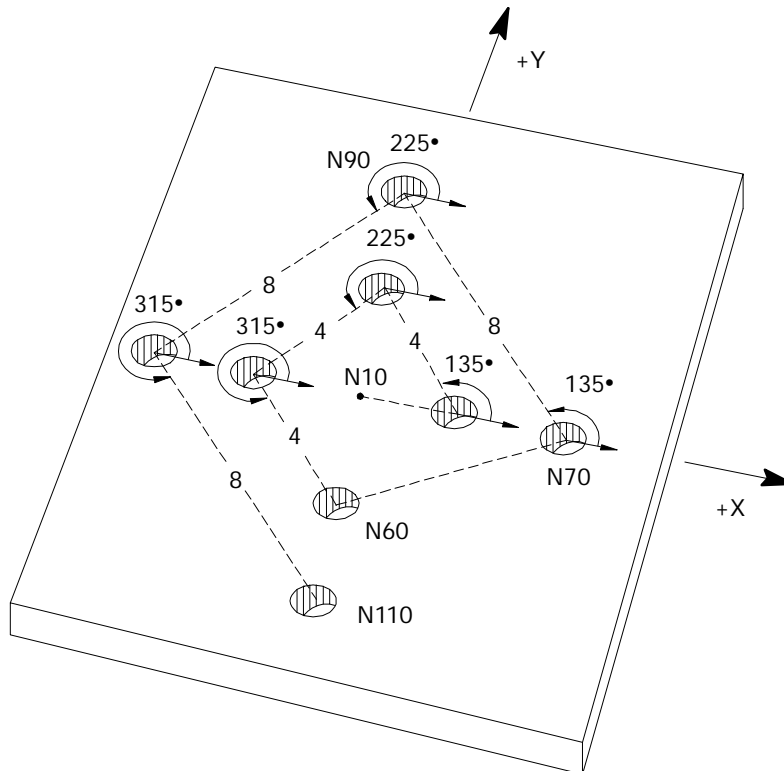
When programming an arc using polar programming the angle and radius values still define the end point of the next move. The center point of the arc must still be designated in the circular block using either the I, J, or K words or by designating a radius value (see the section 14.1.3 on circular interpolation).

It is also possible to use polar programming when the angles are programmed in absolute mode and the radii are in incremental. See Example 14.11 and Figure 14.15.

Example 14.11
Polar Programming - Angle in Absolute, Radii in Incremental

N10 G00 X0Y0 F500;	rapid move to X0 Y0
N20 G90 G81 X3.Y0 R3. Z10.;	drilling cycle at X3 Y0
N30 G16;	polar programming
N40 G91 X4. G90 Y135.;	radius of 4 at 135 deg abs
N50 Y225.;	still radius of 4 at 225 deg abs
N60 Y315.;	still radius of 4 at 315 deg abs
N70 G15 X6. Y0	cancel polar, move to X6 Y0
N80 G16;	polar programming
N90 G91 X8. G90 Y135.;	radius of 8 at 135 deg abs.
N100 Y225.;	still radius of 8 at 225 abs.
N110 Y315.;	still radius of 8 at 315 abs.
N120 M30;	end

Figure 14.15
Results of Example 14.11

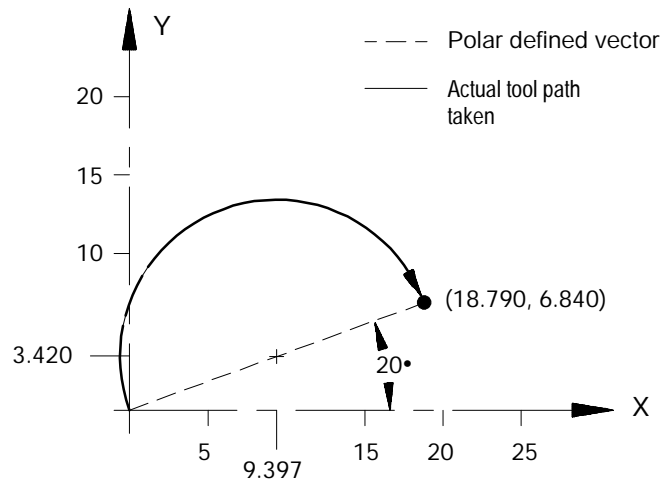


When programming an arc using I, J, or K words the control does not use these values as polar coordinates. Program the center of the arc in the same manner as normal circular programming described in section 14.1.3 . I, J, and K are always cartesian coordinate values.

Example 14.12
Circular Polar Programming

```
G00X0. Y0. ;  
G91G16F100;  
G02X20. Y20. I9. 397J3. 42;  
G15;  
M30;
```

Figure 14.16
Results of Circular Polar Programming Example



14.3 Automatic Motion To and From Machine Home

Machine tools have a fixed machine home position that is used to establish the coordinate systems. The control offers two different methods for homing a machine after power up.

- Manual machine home operation that uses switches or buttons on the MTB panel provided solely for this purpose. Manual homing is discussed in detail in section 4.3.
- Automatic machine home operation that uses a programmed machine home code.

14.3.1 Automatic Machine Homing (G28)

Automatic homing is accomplished through the use of a G28 code. When programmed as the first motion block in a part program (or through MDI) a G28 will cause an automatic homing operation if the axes have not yet been homed. Only axes that have their axis words programmed in the G28 block are homed.

Homing will follow the sequence of homing events described in manual homing (see chapter 4).

The coordinate values which are programmed with the axis words in a G28 block are stored by the control as intermediate point values (described in the next section).

If all the axes programmed in the G28 block have already been homed when the G28 code is executed, then the control will consider it an “Automatic Return to Machine Home” as described in the next section.

Important: When a homing request is made the feedback device for the axis (typically an encoder) must encounter at least one marker before tripping the homing limit switch. If the axis is close to the home limit switch you should jog the axis away from this switch before attempting a homing operation.

Automatic Machine Homing (G28) with Distance Coded Markers

The following outlines automatic machine homing (G28) for an axis with DCM feedback if the axis has not already been homed:

1. The axis moves at a speed and direction defined in AMP by G28 Home Speed and G28 Direction to Home, respectively.

The axis will come to a stop once the axis crosses three consecutive markers on the DCM scale.

Important: To determine an absolute position using DCMs, you must encounter at least three consecutive markers. Thus, if the axis position will not accommodate this assumption, the axis must be moved to another position before attempting a homing operation.

2. When the output command equals 0 (i.e., the axis stops), the control will determine the absolute position. Refer to your AMP manual for more information about DCM Homing for Absolute Position.

If your axis is already homed, refer to the Automatic Return to Home (G28) section later in this chapter.

Important: DCM axis homing must be performed manually or by programming a G28. Attempting to program any motion command other than a G28 will result in the decode error "MUST HOME AXIS".

14.3.2 Automatic Return to Machine Home (G28)

When a G28 is executed in a part program (or through MDI) after the axes have already been homed, it causes a return to machine home. In this case, the axes specified in the G28 block simply go to their respective home position in the machine coordinate system, after moving to a programmed intermediate point. They do not repeat the homing routine of moving to the limit switches and searching for the encoder marker. For example, executing the block:

```
G28 X__ Y__ Z__ ;
```

In either absolute or incremental mode would return the axes automatically to the machine home via an intermediate point. The control stores the intermediate point specified by the axis words (X, Y, Z) in memory to be used as the point of return for the automatic return from machine home operation called out by G29.

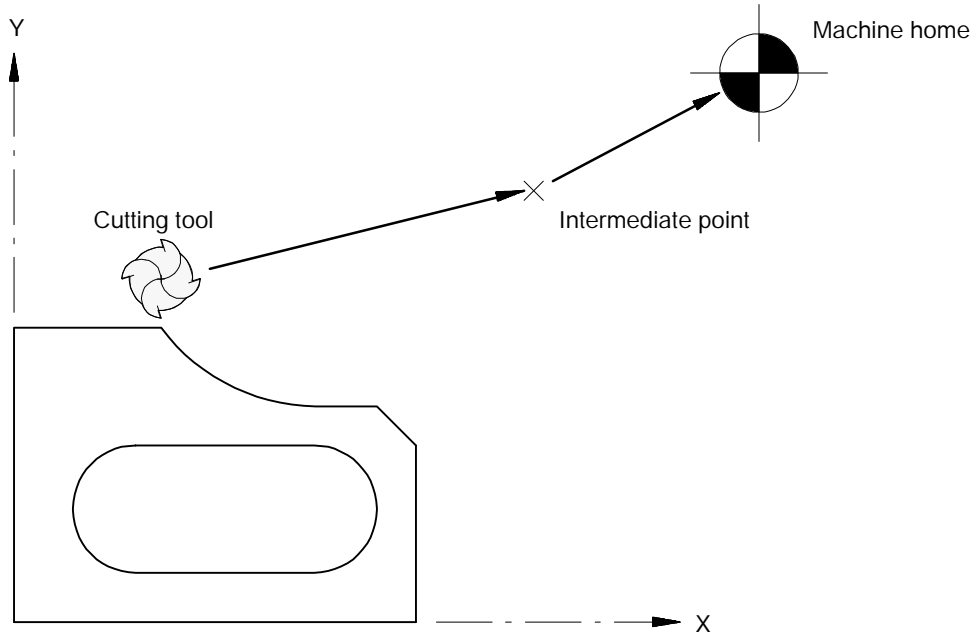
The return operation generates two axis moves both executed at the rapid feedrate. The first move travels to the intermediate point and the second travels to the axis home position.

Although this command moves the axes at rapid feedrate as if in G00 mode, it is not modal. If G01, G02, or G03 modes are active, they will only be temporarily canceled for the return to home moves.

Only the axes specified in the G28 block are moved. For example:

N1 G28 X4. 0;	the X axis is moved to home after moving to 4.0
N2 G28 X4. 0 Y2. 0;	the X and Y axes are moved to home after moving to (4.0 ,2.0)

Figure 14.17
Automatic Return To Machine Home (G28)



Usually a G28 is followed by a G29 (automatic return from machine home) in a part program; however, the control will store the intermediate point in memory for be use with any subsequent G29 block executed before power down. Only one intermediate point is stored for each axis. When a G28 is programmed with a new intermediate point, any axis not programmed in that block will remain at the old value. For example:

N1 G28 X4. 0 Y3. 0;	Intermediate point X=4 Y=3
N2 G28 Y2. 0;	New intermediate point X=4, Y=2

Important: When the control executes a G28 or G30 block it temporarily removes any tool offsets and cutter compensation during the axis move to the intermediate point. The offsets and/or cutter compensation are automatically reactivated during the first block containing axis motion following the G28 or G30 unless that block is a G29 block. If a G29 follows, the offsets and/or cutter compensation remain deactivated on the way to the intermediate point and are re-activated when the axis moves from the intermediate point to the point indicated in the G29 block.

14.3.3 Automatic Return From Machine Home (G29)

When a G29 is executed in a part program (or through MDI) the axis or axes will move first to the intermediate point, and then to the position indicated in the G29 block. If a G28 was just executed, then this will have the effect of returning the axis from machine home. For example, executing the block:

```
G29 X7.0 Y.2 Z-14.0 ;
```

in absolute mode would move the axes to (X7.0, Y.2, Z- 15.0) after passing through the intermediate point stored in control memory. In incremental mode this block would move the axes to a position that is X7, Y2, and Z14 units away from the home point.

The intermediate point is stored in control memory after a G28 return to machine home or a G30 return to alternate home is executed. A G29 block is usually executed after a G28 or G30 block, typically to return the cutting tool to the part after a tool change.

Although this command moves the axes at rapid feedrate as if in G00 mode, it is not modal. If G01, G02, or G03 modes are active, they will be temporarily canceled for the return from home moves.

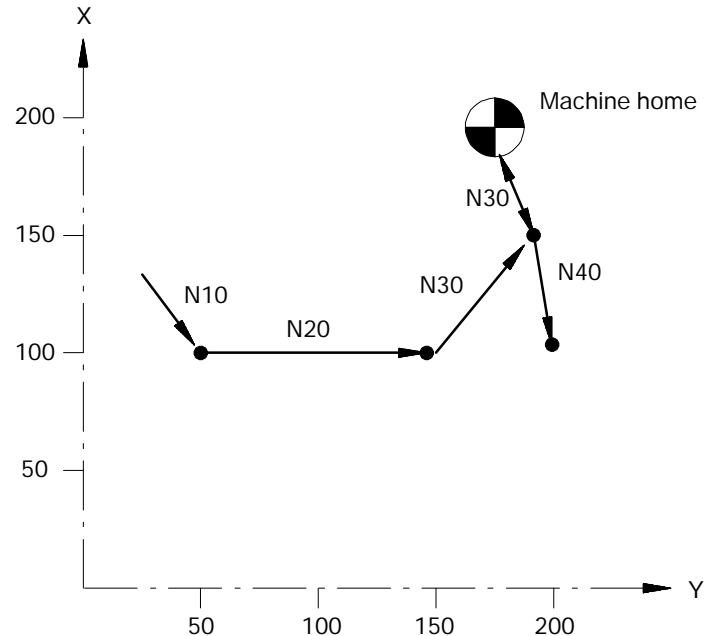
Only the axes specified in the G29 block are moved. For example:

N1 G28 X5.0 Z1.0;	X and Z axes are moved to home after moving to X=5.0, Z=1.0 -- Y is not moved.
N2 G29 X3. ;	X moves to X=5.0 then to X=3.0 --Y and Z do not move.

Example 14.13 Automatic Return From Machine Home Example

```
N00010 G00X0. Y50F500;
N00020 X100;
N00030 Y100X150;
N00040 X150. Y50;
N00050 M30;
```


Figure 14.18
Automatic Return From Machine Home, Results of Example 14.13



Important: When a G29 is executed, tool offsets and/or cutter compensation will be deactivated on the way to the intermediate point and are re-activated when the axis moves from the intermediate point to the point indicated in the G29 block.

14.3.4 Machine Home Return Check (G27)

A G27 causes the control to move the axes at rapid directly to the machine home position. Only the axes included in the G27 block will be moved.

G27 X__ Y__ Z__ ;

The value entered with the axis name in the G27 block must be the machine home coordinate for that axis. If it is not, no axis motion will take place and the control will issue the error message:

“INVALID ENDPOINT IN G27 BLOCK”

Aside from this endpoint check, the only difference between a G27 block and a G00 block requesting a move to the machine home coordinates is that the G27 is not modal. If G01, G02 or G03 modes were active before the G27 was executed, they will be reactivated immediately after the G27 block is completed.

G27 block commands are usually given after tool offset modes have been cancelled.

If an attempt is made to execute a G27 before the axes have been homed the control will go to cycle stop and the following error message will be displayed:

“MACHINE HOME REQUIRED OR G28”

14.3.5 Return to Alternate Home (G30)

The G30 command is similar to the G28, with the main difference being that the axis or axes move to an alternate home position instead of machine home. The command format determines whether the axes return to a second, third, or fourth alternate home position. Any axis programmed in the G30 block must have been homed prior to G30 execution.

The alternate home positions, in reference to the machine coordinate system, are predefined for each axis in AMP by the system installer.

To use the G30 command follow this format:

```
G30 X__ Y__ Z__ ;
      or                (second alternate home position)
G30 P2 X__ Y__ Z__ ;
G30 P3 X__ Y__ Z__ ; (third alternate home position)
G30 P4 X__ Y__ Z__ ; (fourth alternate home position)
```

Important: The control generates the error “P VALUE OUT OF RANGE” if the P value is illegal. For example, a P1 or P5 would be illegal and generate the error.

The axis words in the above block establish the intermediate point in the same manner as the G28 code. That is, the axes will move to the intermediate point defined in the G30 block prior to moving to the alternate home position. When intermediate values are programmed in a G28 block they replace G30 intermediate point values and vice-versa. This intermediate point is used by the G29 automatic return code.

Only those axes included in the G30 block are sent to the alternate home position. For example:

G30 X5.6	The control moves the X axis to second home after moving to 5.6 on the X axis. The Z and Y axes are not moved.
G30 P3 X1.0 Z4.0	The control moves the X and Z axes to third home after moving to 1.0 on the X axis and 4.0 on the Z axis. The Y axis is not moved.

A typical application for the G30 command would be if the automatic tool changer were located at a position other than machine home.

If an axis included in the G30 block has not been homed, block execution will stop and the following error message will appear:

“MACHINE HOME REQUIRED OR G28”

Important: When the control executes a G28 or G30 block it temporarily removes any tool offsets and cutter compensation during the axis move to the intermediate point. The offsets and/or cutter compensation are automatically re-activated during the first block containing axis motion following the G28 or G30 unless that block is a G29 block. If a G29 follows, the offsets and/or cutter compensation remain deactivated on the way to the intermediate point and are reactivated when the axis moves from the intermediate point to the point indicated in the G29 block.

14.4 Dwell (G04)

The G04 command delays the execution of the next data block. Dwell length is specified in either of two types.

- Seconds
- Number of spindle revolutions

The type used is normally dependant on the feedrate mode (G93, G94 or G95) active at the time. The type can also be permanently fixed to “seconds,” regardless of G93, G94 or G95 mode, by setting the proper AMP parameter.

14.4.1 Dwell - Seconds

In the G93 (inverse time feed) and G94 (feed per minute) modes, G04 suspends execution of the commands in the next block for a programmed length of time in seconds.

```
G94G04 P__; X__; U__;
```

Specify the required dwell time by either a P, X, or U word in units of seconds. It does not matter which of these three words are used, as long as only one appears in the same block. The allowable dwell time is 0.001 - 99999.999 seconds.

When programming a dwell in seconds the system installer has the option of writing PAL to allow a portion of the dwell to be skipped. If this feature is used, when the appropriate signal is sent to PAL (from a switch or other device) the control will automatically skip any portion of the dwell that has not been executed and proceed on to the next block in the program. The position of the axes when the skip signal is sent to PAL is recorded and stored as system parameters #5071 - #5076 (see specifics on the G31 skip cycles for details).

14.4.2 Dwell - Number of Spindle Revolutions

In the G95 mode (feed per revolution) G04 suspends execution of commands in the next block for the time it takes the controlling spindle to turn a designated number of revolutions.

```
G95G04 P__; X__; U__;
```

Specify the required dwell length by either a P, X, or U word in units of spindle revolutions. It does not matter which of these three words are used, as long as only one appears in the same block. The allowable range is 0.001 - 99999.999 revolutions.

14.5 Programmable Mirror Image (G50.1 - G51.5)

The programmable mirror image feature is used to mirror (duplicate yet reversed) axis motion commands about some defined plane. This feature is activated using the G51.1 code and cancelled using the G50.1 code. Mirroring takes place about the axis position specified in the G51.1 code, it is independent of the plane selected using G17, G18, or G19.

The format for the G51.1 code is as follows:

```
G51.1 X__Y__Z__ ;
```

The axis motion commands in any following blocks are executed with the motion direction reversed (including incremental moves) as if a mirror was placed on the designated point parallel with the axis. The G51.1 code is modal and remains in effect until cancelled by a G50.1 command.

The axis word programmed with the G51.1 command is used to define the location mirroring will be about. The defined location intercepts the programmed axis at the programmed position. If only one axis is programmed, the mirroring plane is perpendicular to that axis. If more than one axis is programmed, the mirror plane passes through these points.

Important: The control only mirrors those axes that are programmed in the G51.1 block. Axes not programmed in the G51.1 block execute normally.

A G50.1 block cancels the mirror image function.

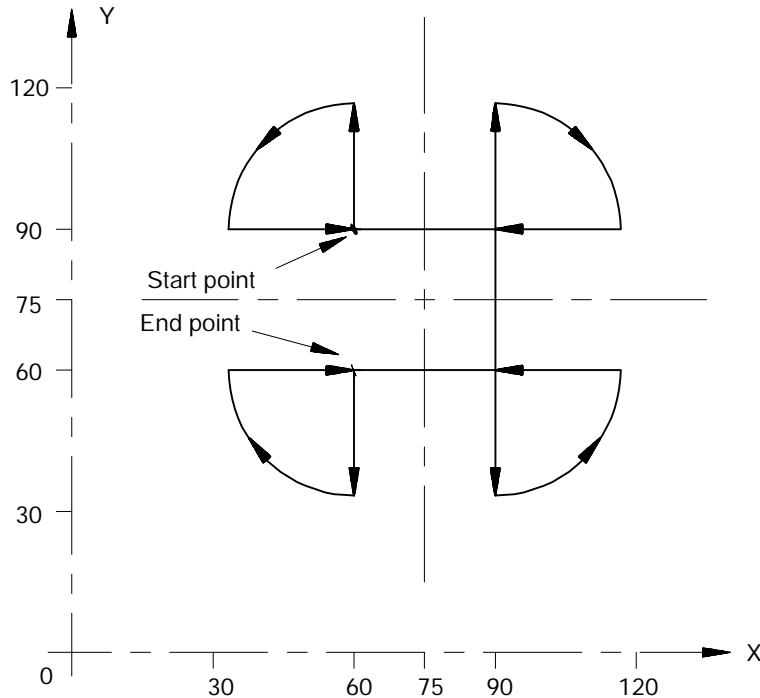
```
G50.1 X__Y__Z__ ;
```

The control only cancels the mirror feature for those axes that are programmed in the G50.1 block. Axes not programmed in the G50.1 block remain mirrored. There is no significance to the axis value programmed with the axis words in a G50.1 block. Axis values may or may not be required depending on how the way AMP was configured by your system installer. In either case, these values will be ignored by the control.

Example 14.14
Programmable Mirror Image

Main Program	Comment
(Mirror);	comment block, main program
G00G90;	rapid positioning, absolute mode
M98P8500;	call subprogram 8500
G51.1X75.;	mirror active on X
M98P8500;	call subprogram 8500
G51.1Y75.;	mirror active on Y (and X)
M98P8500;	call subprogram 8500
G50.1X0;	cancel mirror on X (active on Y only)
M98P8500;	call subprogram 8500
G50.1Y0;	cancel mirror on Y (no mirroring)
M30;	
Subprogram	Comment
O8500;	program number
G00G90X60. Y90.;	rapid to start point
G01Y120. F50;	move 1
G03X30. Y90. R30;	move 2
G01X60.;	move 3
M99;	return from subprogram

Figure 14.19
Results of Programmable Mirror Image Example



When the mirror image function is active on only one of a pair of axes used in circular interpolation or cutter compensation, the control:

- executes a reverse of programmed G02/G03 arcs. G02 becomes counterclockwise and G03 become clockwise.
- activates a reverse of programmed G41/G42 cutter compensation. G41 becomes tool right and G42 becomes tool left.

14.5.1 Manual Mirror Image

In addition to the programmable mirror image function, the control may also be equipped with an optional manual mirror image switch, installed by the system installer, that will activate the manual mirror image function. The manual mirror image function may also be activated by using the {FRONT PANEL} softkey.

The manual mirror image features differ from the programmable mirror image feature in that when using the manual mirror image feature the location of the mirrored plane is fixed along the selected axis in the current work coordinate system. This means that the mirror plane is parallel to the selected axis and passes through the zero point of the currently active work coordinate system.

The mirrored plane is fixed and cannot be moved from the selected axis. This mirrored plane is the equivalent of programming a programmable mirror image and using all zero values for the axis words.

The system installer may install a switch for each of the 4 available axes. What axes are mirrored with what switches is dependant on the PAL program in a particular system. It is also possible to mirror about more than one axis using more than one manual mirror image switch at the same time or one switch may control more than one axis. Refer to documentation prepared by the system installer for details.

Important: It is possible to use programmable mirror image at the same time as manual mirror image. When this is done, the programmable mirror image is done first followed by the manual mirror image. The same axis may be mirrored by programmable and manual mirror image at the same time.

14.6 Axis Clamp

This feature is used to disable the axis position display and allow an axis to be clamped into position. Typically an axis clamp is performed by the execution of an M-code in a part program or by a switch of some type controlled by the operator. The system installer determines how the axis clamp feature is enabled in PAL. Refer to the system installer's documentation for details.

When an axis is clamped, the control will freeze the axis position displays at their position. Any drift or movement generated by some external force will not generate any corrective response from the axis servo. This will prevent the servo from trying to move an axis back into position when it has been mechanically clamped to not move.

Any movement of the axis when it is clamped will be added to the current value of the following error and may be viewed on the screen displaying following error. Refer to your integration manual for more information. If the axis following error exceeds the allowable maximum following error (set in AMP), an error will be generated, and the control will go into E-STOP.

When the axis is unclamped, the control position display will be reactivated and the servo will return the axis to the necessary position for zero following error.

14.7
Feed to Hard Stop (G24)

The feed to hard stop feature is used to position the axis of a transfer line station or the transfer bar of the station against a mechanical stop and hold it against the stop. This mechanical stop physically halts axis travel. The system installer determines the position of this hard stop based on mechanical consideration of the machine and the process currently being performed by the axis or transfer bar.

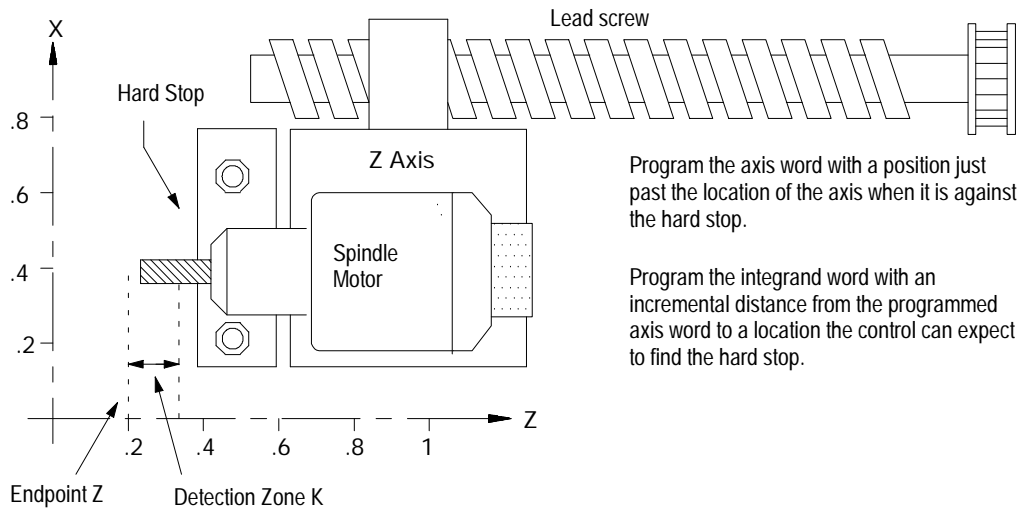
Program a feed to hard stop using a G24 code. Format for the G24 code is as follows:

$$G24 \begin{pmatrix} X_ \\ Y_ \\ Z_ \end{pmatrix} \begin{pmatrix} I_ \\ J_ \\ K_ \end{pmatrix} F_;$$

Where:	Programs:
X, Y, or Z	Hard stop axis. Use the axis word associated with the axis to be hard stopped. Program an endpoint for the axis that is past, but very close to, the actual hard stop location. The axis word can be programmed as either an absolute or incremental value (G90 or G91 mode).
I, J, or K	Detection zone. Use the axis integrand associated with the axis to be hard stopped. Program an incremental distance from the programmed endpoint for the control to start looking for a hard stop. The axis detection zone is an unsigned value.
F	Optional in the G24 block. F defines the active feedrate for the G24 feed to hard stop. If not programmed the currently active feedrate is used.

Example 14.1
Feed to Hard Stop

G90 G01 F20;
G24 Z.2 K.15;



Moving to the Hard Stop

The G24 code must be in a block that programs a position for one and only one axis. The G24 code is non-modal (G-code group 0).

The active cutting mode when the G24 code is executed must be G01 (linear interpolation). Other cutting modes and rapid traverse (modal group 01), are invalid during a G24 block.

Once the G24 code is executed the axis moves towards the programmed endpoint at the currently active feedrate. When the axis enters the detection zone set up with the integrand word, the control expects to encounter a hard stop. If a hard stop is detected before entering this detection zone, the control generates an error. An error is also generated if the control reaches the programmed endpoint without encountering a hard stop. Both of these error conditions force a cycle stop.

Detecting the Hard Stop

A hard stop is detected when the control's torque output reaches a Hard Stop Detection Torque that the system installer configured in AMP. This torque limit must be reached after entering the detection zone and before reaching the G24 blocks' programmed endpoint or an error is generated.

Once an axis is positioned against a hard stop, that axis remains in the hard stop state until it is programmed away from the hard stop. While against the hard stop, the control applies a set holding torque to the servo keeping the axis firmly pinned against the hard stop. This hard stop holding torque amount is determined by the system installer in AMP.

Only one axis can be positioned against a hard stop at any one time. Attempting to position more than one axis against a hard stop results in a programming error. Once an axis is positioned against a hard stop, it must be programmed away from the hard stop before another axis can use the hard stop feature. You can program an axis that is currently in the hard stop state to a different hard stop location with a G24 block provided the hard stop is in a direction opposite the current hard stop holding torque.

Special Considerations

Feature:	Consideration:
Control Reset	If a control reset operation is performed while the control is against a hard stop the holding torque is released and the axis is taken out of the hard stop state.
Block Reset	If a block reset is performed during a G24 block before the hard stop has been reached, the torque limits applied to that axis are removed and the G24 block is aborted.
Program Checking	Feed to hard stop blocks are executed as normal G01 moves while in QuickCheck. While in dry run the control will not execute G24 feed to hard stop blocks at rapid. Feed to hard stop blocks are executed at the programmed feedrate during dry run and all the axis torque restrictions are applied (dry run is ignored during the G24 block).
Deskew and Dual Axes	Only one servo can be fed to a hard stop. This restriction makes programming a G24 code for either a Deskew axis or Dual Axes invalid.
Probing	You can not use probing with the feed to hard stop feature because a probe G-code and a hard-stop G-code cannot be programmed in the same block. The probe must be unarmed when the G24 block is executed, and when the axis is removed from the hard stop. You can however perform probing while the axis is parked against the hard stop.
Block Retrace	You can not retrace any block that moves an axis off of a hard stop. This is because the start point of that move is determined by the hard stop which can only be safely reached by executing a G24 block. You can retrace a G24 block as long as the axis is still against the hard stop when the retrace operation begins. You cannot retrace through programmable acc/dec blocks (G47.x and G48.x). However, you can retrace through blocks where programmable acc/dec was already active.
Interrupt Programs	You can execute an interrupt program during a G24 block provided you are not performing a type 1 interrupt (type 1 interrupts are incompatible with the hard stop feature). You can not move an axis that is currently holding against a hard stop using an interrupt macro. You can not execute a G24 block within an interrupt macro.
Exact Stop Mode (G61)	The G61 (exact stop mode) does not function on G24 blocks. G61 mode is ignored when a G24 is executed.
Polar Programming (G16)	You can not program a G24 block if the axis you are programming against the hard stop is in the current plane and the control is in polar programming mode.

END OF CHAPTER

Using QuickPath Plus•

15.0 Chapter Overview

The QuickPath Plus (QPP) feature is offered as a convenient programming method to simplify programming. This method of programming can prove useful in simplifying the programming of a part directly from a part drawing. In this chapter we describe:

How to use:	On page:
Linear QuickPath Plus	15-3
Circular QuickPath Plus	15-7

15.1 Using QuickPath Plus

The most significant advantage of the QPP feature is that the programmer no longer needs to calculate the end-point of every block or every point of intersection. QPP will determine these points from angles and lengths.

QPP uses the following addresses:

, A	Angle - This word is always displayed as by the control even if the angle is named differently in AMP. If you have a 9/240 program that uses a different address than ,A and you want to run the program on a 9/230, 9/260 or 9/290 control the angles will work but the control names them ,A.
L	Length - This word is used to define the length of a linear tool path, the direction of which is determined by the angle (A). The length will always be interpreted as an incremental distance.

The control offers a variety of sample patterns with prompting to aid in the programming of QPP. These are found under the QuickView feature discussed in section 5.3.

Keep the following in mind when programming QPP:

- When programming using QPP, any axis words that are programmed must be in the current plane; and angles are measured from the first axis defining that plane. All examples in this section will assume the XY plane is active (angles are measured relative to the X axis).
- QPP always uses “,A” as the angle word. When creating new programs, you should always program the QPP angle with ,A. The system installer has the ability to define in AMP an additional letter that can also be used for the QPP angle. Refer to your system installer’s documentation. This additional QPP angle word is provided only for program compatibility with older systems.

- The angle word (*,A*) is always interpreted as an absolute angle regardless of the current mode (G90 or G91).
- The L-word is always interpreted as an incremental distance from the current position regardless of the current mode (G90 or G91). Radius or diameter mode (G08 - G09) has no effect on the *,A*- or L-word.
- If it is necessary to program more than one block to perform the QPP operation being used, and if an error is made in one of the program blocks, the control will always show the error as being in the first block of the two blocks regardless of whether the error is in the first or the second block. If executing in <SINGLE BLOCK> mode, the control will stop after executing the first block as normal.
- If it is necessary to program more than one block to perform the QPP operation being used, a maximum of 4 non-motion blocks may be programmed between these blocks. A non-motion block is any block that does not generate axis motion on one of the two axes in the current plane.
- The following G-codes will cause a syntax error if programmed in any QPP block:

All G-codes in G-code Group 0 (except G04, G09, and G60)

All G-codes in G-code Group 1 (except G00, G01, G02, and G03).

All G-codes in G-code group 4, 6, 9, 10, 11, and 16.

For a listing of G-codes and their group numbers, see the G-code table in appendix C in the back of this manual.

- If it is necessary to program more than one block to perform a QPP operation, it will cause an error if the current plane is changed between blocks (G17, G18, or G19).
- If an angle is programmed in a circular QPP block, an error will be generated.
- If an L word is programmed in a G13 or G13.1 block, an error is generated.

15.2
Linear QuickPath Plus

One-end coordinate

Many times part drawings will only give a programmer one-axis dimension for a tool path and require that the other axis dimension be calculated by the angle. The following QPP feature eliminates the need for this calculation. This must be a linear block (see section 15.3 for circular).

The format for this block is as follows:

```
, A__ {X__};  
      Y__
```

Where :	Is :
, A	Angle - This word is always displayed as by the control even if the angle is named differently in AMP. If you have a 9/240 program that uses a different address than ,A and you want to run the program on a 9/230, 9/260 or 9/290 control the angles will work but the control names them ,A.
X, Y	End Point - This word is used to program one of the coordinates of the end point of a linear path. The control will calculate the other end point automatically. This can be any axis word that is in the current plane.

Only one axes word from the current plane may be programmed in this block. Any axis word that is not in the current plane will be executed as a normal linear move to that coordinate and combined with the QPP generated tool path.

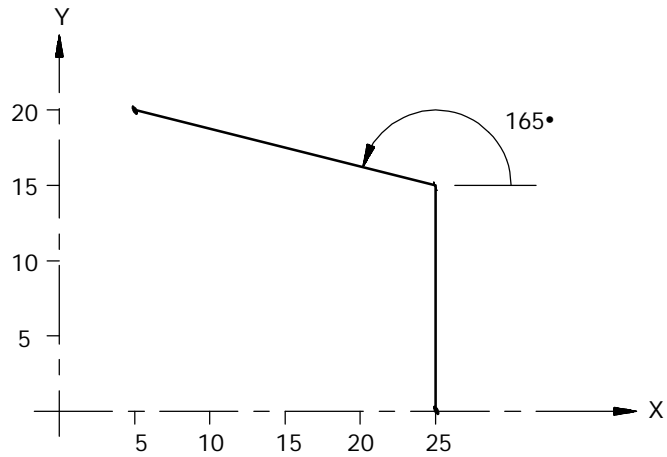
If both axis words from the current plane are entered in the block, the angle is ignored and the control moves to the coordinate position programmed with the axis words. All examples in this section will assume the XY plane is active.

Important: If the programmed tool path is going to be parallel to an axis in the current plane, the axis word for the end point in the block should be for the axis in the current plane that is parallel to the tool path. This means if the value of the angle (,A word) is 0 or 180 degrees the second axis in the plane must be programmed in the block. If the value of the angle is 90 or 270 degrees the first axis in the plane must be programmed in the block.

Example 15.1
Angle Designation

```
N10 G00 X25 Y0 F100. ;  
N20 G01 Y15 , A90;  
N30 X5. , A165;  
N40 M30;
```

Figure 15.1
Results of Angle Designation Example 15.1



Important: An arc may also use an angle (*A*) program block. This is discussed in chapter 16.

No end coordinate known (*L*)

This feature of QPP allows the programmer to define a tool path using only the start point angle and length of a tool path. This must be a linear block.

The format for this block is as follows:

, A__ L__;

Where :	Is :
, A	Angle - This word is always displayed as by the control even if the angle is named differently in AMP. If you have a 9/240 program that uses a different address than ,A and you want to run the program on a 9/260 or 9/290 control the angles will work but the control names them ,A.
L	Length - This word determines the length of the tool path. It is measured from the start point to the end point of the move along a linear path. No coordinate points are necessary.

Important: If any axis word from the current plane is designated in the block the L word is ignored and the control calculates the end point from the angle and the axis word. If an angle (,A) or a length (L) is programmed in a block that also contains both axis words in the current plane, then QPP is not performed and the control ignores the ,A and the L words in the block.

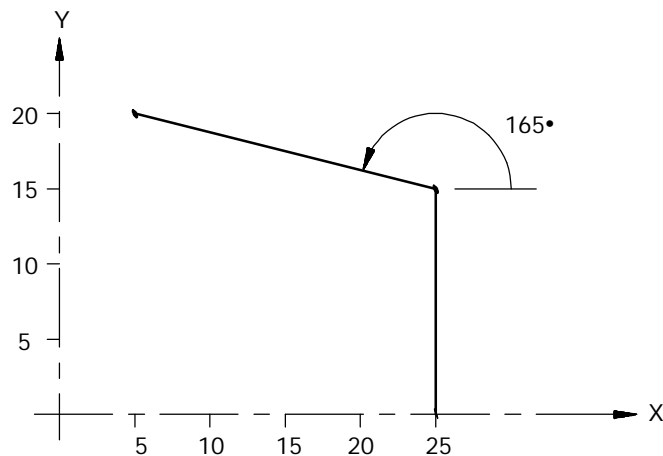
Example 15.2
Angle with Length Designation:

```

N10 G00 Y0. X25. F100;
N20 G01 ,A90 L15;
N30 ,A165 L20. 7;
N40 M30;

```

Figure 15.2
Results of Angle With Length Designation Example 15.2



No Intersection Known

This feature of QPP allows the programmer to define two intersecting, consecutive, linear tool paths without knowing the point that the actual intersection takes place at. Both of these blocks must be linear blocks and programmed in absolute mode. The angle of both of these lines must be known.

This is done with a sequence of two linear blocks (in the current plane) in which QPP is used to calculate the end point of the first block. The start point of the first block is the current tool position.

Important: The second block of these two blocks must be programmed in absolute mode. Any attempt to program the second block in incremental will generate an error.

The format for these blocks is as follows:

```
, A__ ;  
, A__ X__ Y__ ;
```

Where :	Is :
, A	Angle - This word is always displayed as by the control even if the angle is named differently in AMP. If you have a 9/240 program that uses a different address than ,A and you want to run the program on a 9/260 or 9/290 control the angles will work but the control names them ,A.
XY	End Point of second block - These are the actual coordinate location of the end point of the second block. They must be programmed as absolute values and must be axes in the current plane.

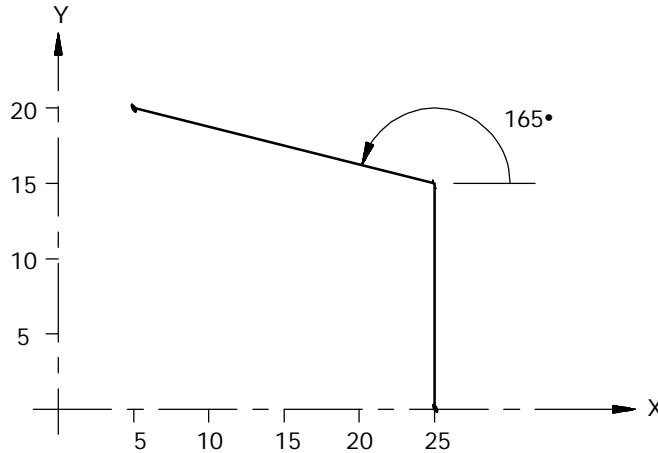
Important: There may be up to four program blocks between the two blocks in the above format, however, these blocks may not generate axis motion in the current plane.

Both of these blocks must be programmed in the same plane. If the current plane is changed between these two blocks execution (G17, G18, or G19 is programmed) the control will generate an error.

Example 15.3 QPP When An Intersection is Unknown

```
N10 G00 X25 Y0 F100;  
N20 G01 , A90;  
N30 , A165 X5 Y20. ;  
N40 M30;
```


Figure 15.3
Results of Unknown Intersection From Example 15.3



If the control cannot determine an intersection point for the two linear paths (for example if the paths are parallel) an error will occur.

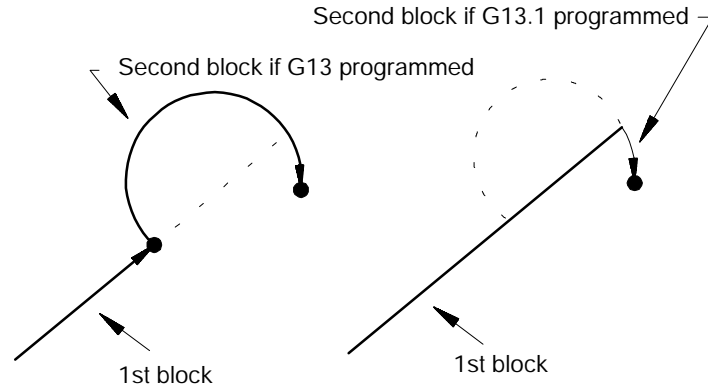
15.3
Circular QuickPath Plus
(G13, G13.1)

Circular QPP is used to help the programmer when a drawing does not call out the actual intersection of two consecutive tool paths and at least one of the tool paths is circular. This prevents the programmer from having to do any complex calculations to determine end points and start points when an arc is involved.

For most cases of circular QPP there may be two possible intersection points for the two defined blocks. Define which intersection is desired using either G13 or G13.1 in the first of the two blocks. Programming G13 will define the first intersection that occurs when the tool path of the first block intersects with the second block. Programming a G13.1 will define the second intersection that occurs when the tool path of the first block intersects with the second block.

G Code	Indicates
G13	First Intersection
G13.1	Second Intersection

Figure 15.4
G13 vs G13.1 Intersections



When programming Circular QPP, remember:

- When there is only one intersection involved with the tool paths, the G13 and G13.1 codes may be programmed interchangeably, however, one must be programmed.
- The G13 or G13.1 code must be programmed in the first of the two blocks defining the two tool paths.
- If the arc is programmed with an R word, the two tool paths must be tangent. The sign (+ or -) of the R word will determine the arc is less than or greater than 180 degrees as discussed in chapter 16.
- The angle word (.A) may not be programmed in a circular block.
- Both absolute coordinate values in the current plane must be programmed for the second block. Both must be programmed regardless of whether the final coordinates change or not.

Linear to Circular Blocks

When the coordinates of the intersection of a linear path into a circular path are not known, use the following format. Note that G13 or G13.1 must be programmed. These blocks must be programmed in absolute.

Format:

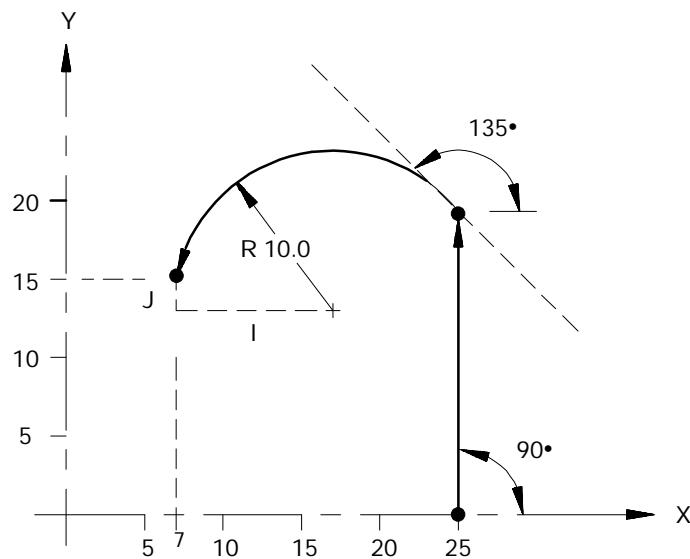
```
G13 G01 ,A__ ;           or   G13G01 ,A__ ;
G02 X__ Y__ I__ J__ ;     G02 X__ Y__ R__ ;
```

Important: If the second block is an arc and it is programmed using I, J, and K integrands, the values programmed with I, J, and K are not measured from the start point of the arc. This is because the start point of the arc is normally unknown when using this format. When using this format the integrands specify the distance from the end point of the arc to the center point.

Example 15.4
Line Into Arc Without Programming Intersection

```
G00X25. Y0. F500
G01G13. 1, A90;
G03X7. Y15. I9. 21J-2. ;
M30;
```

Figure 15.5
Results Of Example 15.4



Important: R may not be programmed to specify the arc radius for linear to circular block combinations unless the two tool paths are tangent.

Circular to Linear Blocks

When the coordinates of the intersection of a circular path into a linear path are not known, use the following format. Note that G13 or G13.1 must be programmed in the first of the two blocks. These blocks must be programmed in absolute.

Format:

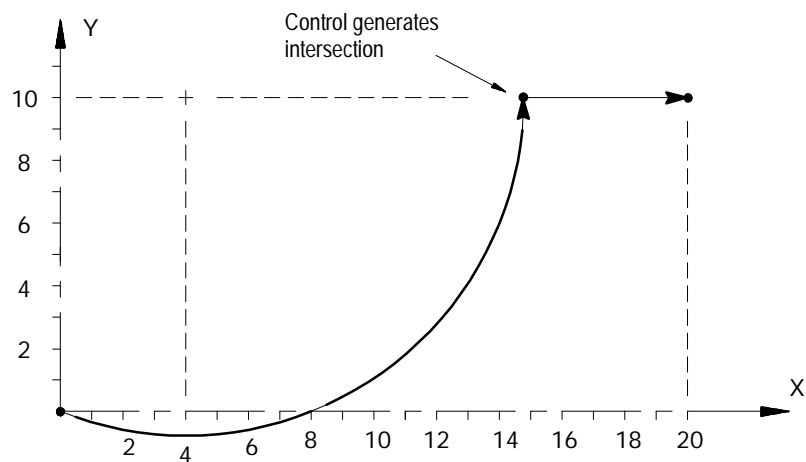
```
G13 G02 I__ J__ ;      or      G13 G02 R__ ;  
G01 ,A__ X__ Y__ ;      G01 A__ X__ Y__ ;
```

Important: J, and K values are the normal integrand values when using this format (measured from start point of arc to arc center). These are discussed in chapter 16.

Example 15.5
Arc Into Line Without Programming Intersection Point

```
G00X0. Y0;  
G13G03I4J10F100;  
G01, A0X20Y10;  
M30;
```

Figure 15.6
Results of Example 15.5



Important: R may not be programmed to specify the arc radius for linear to circular block combinations unless the two tool paths are tangent.

Circular to Circular Blocks

When the coordinates of the point of intersection of a circular path into a circular path are not known, use the following format. Note that G13 or G13.1 must be programmed. If using this format the R word may not be used to specify the radius of an arc in either of the circular blocks. These blocks must be programmed in absolute.

Format:

```
G13 G02 I__ J__ ;  
G02 X__ Y__ I__ J__ ;
```

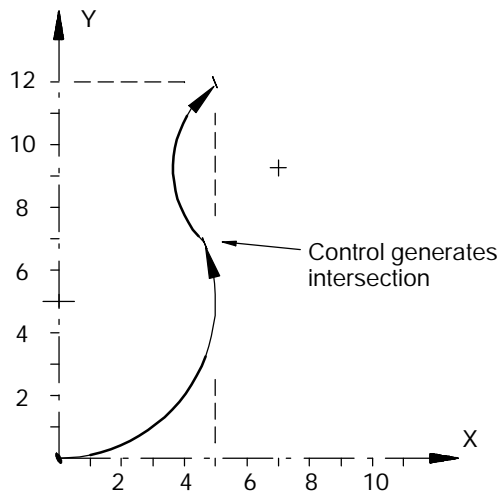
Important: The I, J, K integrand vectors are not necessarily the same values as discussed in section 15.1.3. The integrands of the first circular block specify the distance from the start-point to the center of the circle. The integrands of the second circular block specify the distance from the end-point to the center of the circle (this is the reverse of normal). At least one of these integrand words must be programmed in each of the two circular blocks.

Neither circular block may contain an angle word (,A) when using this format.

Example 15.6
Arc Into Arc Without Programming Intersection

```
G0X0Y. ;  
G13G03J5F100. ;  
G02Y12X5I 2J- 2. 75;  
M30;
```

Figure 15.7
Results of Example 15.6



END OF CHAPTER

Using Chamfers and Corner Radius

16.0 Chapter Overview

This describes how to use chamfer and corner radius to create corners. A chamfer is a linear transition between blocks. A corner radius is an arc transition between blocks.

16.1 Chamfers and Corner Radius

For cornering you can use either a chamfer or a corner radius between two motion blocks.

Both the chamfer and the corner radius features are generated between two motion blocks which must be programmed in the same plane. The motion block with the chamfer (*,C*) or the corner radius (*,R*) word is defined as the first cornering block. The next motion block in the cornering plane is defined as the second block.

<i>,C</i>	Chamfer size - This word is used to define a chamfer length that connects two intersecting tool paths. This word determines the distance that the chamfer will begin and end from the tool paths intersection.
<i>,R</i>	Corner radius - This word is used to define the radius of an arc that is tangent to two intersecting tool paths.

If more than one *,C* or *,R* word is programmed in the same block, only the right-most word is used, other is ignored. The second block can also have a chamfer or corner radius word in it. If it does, the second block will also be used as the first block of the next chamfer or corner radius.



CAUTION: If a programming error of some type is made in the block defining the second tool path in the chamfer or radius blocks, the control will not be able to cut the correct chamfer or radius. Instead the first block will be executed to its programmed end-point. This may cause damage to the part or cutting tool.

The control will generate an error if there are more than four non-motion blocks between the first and second motion blocks defining the corner transition. A non-motion block is any block that does not generate axis motion in the currently active plane.

You can use chamfers and corner radius with QuickPath Plus. They may be programmed in either absolute (G90) or incremental (G91) modes.

Using Chamfers

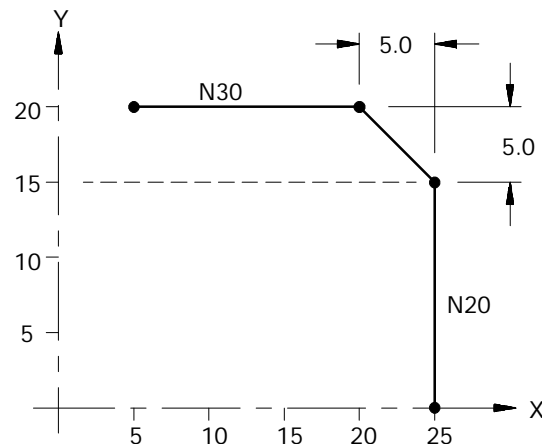
Program a chamfer size following the address ,C to cut a chamfer between consecutive tool paths. The chamfer word must follow a comma (,) and is programmed in the first of two paths connected by the chamfer. The value following the ,C address is the amount of tool path cut of each programmed tool path by the chamfer. The angle that the chamfer makes with the tool paths is dependant on the size of the chamfer.

The chamfer size is measured from the intersection of the two blocks. If the block is linear, then the distance programmed with the ,C word is measured from the intersection of the two tool paths along the linear path. If the block is circular, then the chamfer distance programmed is applied as a chord length on the arc measured from the intersection between the two blocks. This applies regardless of the combination of arcs and lines to be cut with a chamfer.

Example 16.1
Using the Basic Chamfer ,C

```
N10 G00 Y0 X25 F100;  
N20 G01 Y20. , C5. 0;  
N30 X5. 0;  
N40 M30;
```

Figure 16.1
Results From Chamfer Example 16.1



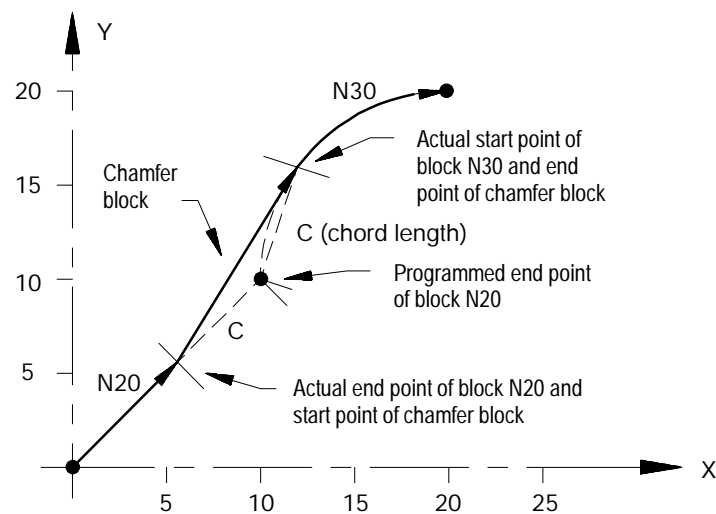
Example 16.2
Linear-to-Circular Motions with Chamfer

```

N10 G00 X0 Y0 F100;
N20 G01 X10. Y10. , C3;
N30 G02 X20. Y20. R10;
N40 M30;

```

Figure 16.2
Results From Chamfer Example 16.2



Corner Radius

Use the `,R` command to program a radius between two intersecting tool paths. The `R` command must be programmed after a comma (`,`). Program the `,R` followed by the radius size in the block where the first path is programmed. The control looks ahead to the block commanding the second path, and automatically inserts the circular rounding block to meet that path. This inserted, circular block is always tangent to both programmed tool paths. If the control cannot generate an arc that is tangent to both paths with the programmed `,R`, then the control will generate an error.

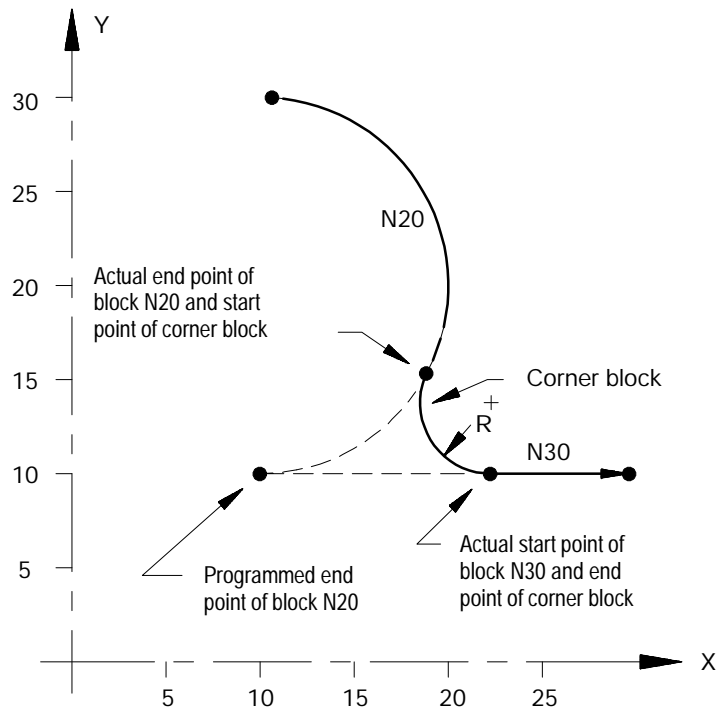
The first corner radius block always terminates at the point on the block where the rounding block is tangent to the first block. The rounding block terminates at the point where the generated rounding block is tangent to the second rounding block. The second rounding block starts from the end-point of the generated circular block and continues on to the programmed end-point of the second block.

Important: If the two motion blocks are tangent to each other, then any corner-rounding commands are ignored.

Example 16.3
Programming a Radius For a Circular Path into a Linear Path

```
N10 G00 X10. Y30;  
N20 X10. Y30 F100;  
N30 G02 X10. Y10 R10, R3;  
N40 G01 X30. Y10;  
N50 M30;
```

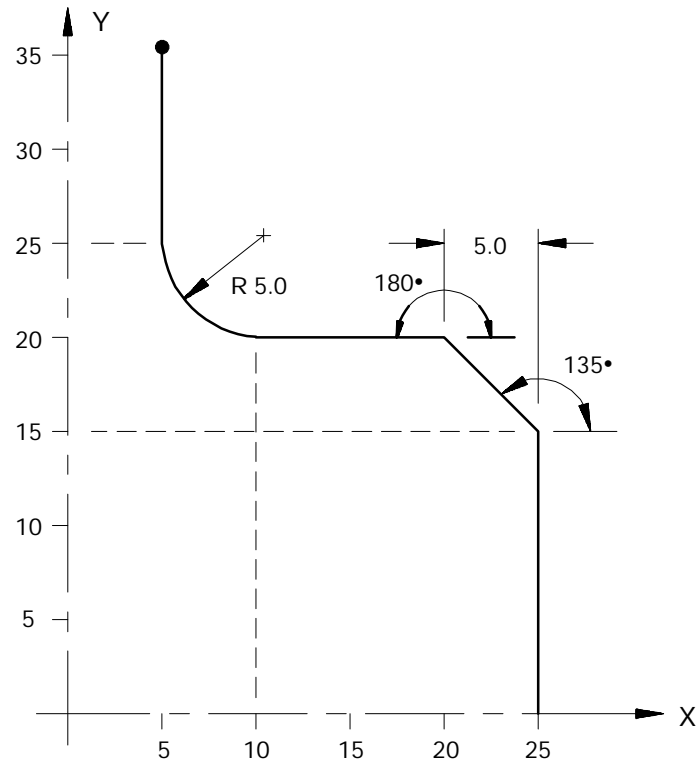
Figure 16.3
Results of Radius Example 16.3



Example 16.4
Radius and Chamfer with QuickPath Plus

```
N10 G00 X25. Y0. F100;  
N20 G01 , A90, C5. 0;  
N30 X5. Y20. , A180. , R5. 0;  
N40 Y35;  
N50 M30;
```

Figure 16.4
Results of Radius Example 16.4



Guidelines for Using Chamfers and Corner Radius

- If the control is executing in single block mode, the control will enter the cycle stop state after executing the first block and the adjacent chamfer or corner radius.
- If non-motion blocks are programmed separating the two intersecting blocks for the corner radius and chamfer features, the control will execute the chamfer or radius immediately after the first block. The non-motion blocks are executed after the control has executed the chamfer or radius.
- Any negative signs programmed with the ,C or ,R words are ignored. The absolute value of the word is used to cut the chamfer or radius. For example ,C-10 is used as ,C10.
- An error is generated if the length of a chamfer is larger than the programmed length of the first or second move, or for corner rounding if the programmed corner radius is so large that the tangent point on both of the two programmed blocks does not exist.

- An error is generated if an attempt is made to change planes between blocks that are chamfer or corner radius blocks.
- ,C and ,R must be programmed in blocks that contain axis motion in the current plane. If they are programmed in a block that does not contain axis motion in the currently active plane, the control will generate an error.
- ,C and ,R may not be programmed in a block that contains any of the following:
 - Any fixed cycle G codes
 - Any Dwell commands
 - Thread cutting blocks
 - Programmable zone G codes
- The system installer determines in AMP the resolution of the ,C and ,R words for both inch and metric programming. Refer to documentation prepared by the system installer for details.

END OF CHAPTER

Spindles

17.0 Chapter Overview

This chapter describes how to program spindles:

Information about:	On page:
Controlling Spindle	17-1
Spindle Orientation	17-3
Spindle Direction	17-5
Synchronized Spindles	17-6

17.1 Controlling Spindle (G12.1, G12.2, G12.3)

The G12 code is used to program the active controlling spindle for features and modes requiring spindle operation. The G12 code is modal. Only one spindle may be the controlling spindle. Only the 9/260-9/290 control uses more than one spindle. All other spindles are auxiliary spindles.

- **G12.1 – Spindle 1 Controlling**
- **G12.2 – Spindle 2 Controlling**
- **G12.3 – Spindle 3 Controlling**

Table 17.A shows the spindle capabilities of each control.

Table 17.A
Spindle Capabilities of the 9/Series Controls

Type of 9/Series Control	Number of Spindles	Spindle Type
9/230, 9/240	1	Primary
9/260	2	Primary, Auxiliary 2
9/290	3	Primary, Auxiliary 2, Auxiliary 3

Spindles 1, 2, and 3 must be configured in AMP, and the associated spindle parameters must be set properly to provide for the required spindle functions.

For systems with no spindle configured, simulated spindle feedback is provided for the controlling spindle. This allows all control features that require spindle feedback, i.e., IPR feedrate, threading, CSS, to simulate the feedback from a spindle even through the AMPed system configuration contained no spindle. The default is 4000 counts-per-rev device.

Important: On the 9/260 and 9/290 controls, if the auxiliary spindles are programmed but have not been configured as active through AMP, these errors are given as decode errors on any blocks that have the G12.2 or G12.3 code:

“SPINDLE 2 NOT CONFIGURED” and/or

“SPINDLE 3 NOT CONFIGURED”

Spindle Speed (S-word)

Use the S-word to program the spindle speed for all configured spindles. The common S-word can be applied per spindle by associating the S-word in the same block with the spindle directional M-codes. See the section on spindle direction for information about spindle directional M-codes. If no directional M-code is programmed in the block with the S-word, then the S-word is applied to the active controlling spindle. The S-word units represent revolutions per minute (RPM).



CAUTION: The displayed S-word always shows the controlling spindle's programmed spindle speed. The display of the S words for noncontrolling spindles must be handled by some other display means. See the system installer's documentation for display capabilities of the active spindle speed for noncontrolling spindles.

17.2 Spindle Orientation (M19)

For each spindle configured in a system, the control is equipped to perform a spindle orient operation. This operation is used to rotate the spindle to a given angle. Typically this may be used to orient the spindle for tool positioning for special machining operations, position a mechanical chuck for automatic chuck wrench operations, etc. This orient operation is not the same as using a spindle as an axis for positioning. An orient operation is performed separately from axis motions and can not be interpolated like normal axis motions.

There are two types of spindle orients available. They are:

- **Open loop orient** - The spindle does not use a feedback device for this type of orient. The final destination of the spindle when performing an open loop orient is determined by PAL. Typically there is some form of hardware switch used to determine the spindle is at the proper position. When the open loop orient is performed the spindle is turned at an AMP defined RPM and in an AMP defined direction.
- **Closed loop orient** - The spindle must be equipped with a feedback device. The final destination of the spindle when performing a closed loop orient may be determined in AMP, or entered in a program block requesting an orient. When the closed loop orient is performed the spindle is positioned at an AMP defined orient position. The direction of the orient is as follows:

if the spindle is turning the orient will complete in the same direction as the spindle is currently revolving. If the spindle is turning faster than the maximum orient speed (as programmed in AMP) it will first slow to the orient speed before performing the orient.

if the spindle is not turning the orient will be performed in whichever direction that will result in the spindle reaching the required position using the shortest angular distance.

Important: A spindle orient is also sometimes automatically requested by the control when performing some of the milling cycles. This milling cycle orient orients to either the AMP defined position if using a closed loop orient type or to the position defined as the open loop orient position by hardware and PAL interface.

Important: In systems allowing multiple spindles (9/260 and 9/290), only one M19 code can be in a block. If two or more M19 codes appear in one block, e.g., M19.2 M19#, this error message appears, "ONLY ONE M19 ALLOWED PER BLOCK."

Refer to the system installers documentation to determine which orient the system is equipped to perform. This manual assumes that a closed loop type orient is available. If an open loop orient is the only spindle orient available on a specific system refer to the system installers documentation for details on its operation as it is highly PAL dependant.

Both open- and closed-loop spindle orients can be requested either by programming the appropriate spindle orient code (M19, M19.2, M19.3) in a program block, or by requesting one through PAL. If closed-loop orient is requested through PAL, the orient angle is fixed at the default orient angle preset by the system installer in AMP.

If a closed-loop orient is requested by programming the appropriate spindle orient code (M19, M19.2, M19.3), the option exists to orient the spindle to the AMP-defined orient position or to a position programmed with an S parameter in the M19 block. The S parameter defines an angle at which the spindle is positioned relative to an angle of zero that is fixed for a specific machine. Refer to the documentation prepared by the system installer. This S parameter always programs an absolute angular position. The angle programmed is not affected by incremental or absolute programming mode (if open-loop orient is being used, the value programmed with the S parameter is ignored).

The M19 code is modal. To perform an orient to a programmed angle however an M19 with an S word must be re-programmed in each block to orient. Programming an S word alone will replace the current modal spindle speed used later when the M19 mode is canceled. Cancel the M19 code by programming one of the other spindle mode M codes.

To cancel spindle orient:	Program:	Meaning:
Spindle 1 code M19	M03 M04 M05	Spindle 1 clockwise Spindle 1 counterclockwise Spindle 1 stop
Spindle 2 code M19.2	M03.2 M04.2 M05.2	Spindle 2 clockwise Spindle 2 counterclockwise Spindle 2 stop
Spindle 3 code M19.3	M03.3 M04.3 M05.3	Spindle 3 clockwise Spindle 3 counterclockwise Spindle 3 stop

17.3 Spindle Direction (M03, M04, M05)

Use the spindle directional M-codes to program each configured spindle program controlled spindle rotation.

Table 17.B lists the spindle direction codes.

Table 17.B
Spindle Directional Codes

Spindle Type	Directional Code	This means:
Primary	M03 M04 M05	Spindle 1 clockwise Spindle 1 counterclockwise Spindle stop
Spindle 2	M03.2 M04.2 M05.2	Spindle 2 clockwise Spindle 2 counterclockwise Spindle 2 stop
Spindle 3	M03.3 M04.3 M05.3	Spindle 3 clockwise Spindle 3 counterclockwise Spindle 3 stop

Each spindle can have independent rotational control, and the rotational speed is programmed by using the S-word. If a directional spindle code is programmed in the same block as the S-word, then that S-word is applied to each of the block's associated spindles.

Example 17.1
9/290 Control with 3 Spindles Configured in AMP

N0001 M05	Spindle 1 stop
N0002 M05. 2 M05. 3	Spindles 2 & 3 stop
N0003 M03 M04. 2 S150	Spindle 1 clockwise 150 rpm Spindle 2 counterclockwise 150 rpm
N0004 M03. 2 M04. 3 S10	Spindle 2 clockwise 10 rpm Spindle 3 clockwise 10 rpm

Important: On the 9/260 and 9/290 controls, if the auxiliary spindle directional M-codes are programmed but the auxiliary spindles have not been configured as active through AMP, these errors are given as decode errors on any blocks that have directional M-codes of the associated spindle programmed:

“SPINDLE 2 NOT CONFIGURED” and/or

“SPINDLE 3 NOT CONFIGURED”

17.4 Synchronized Spindles

Use this feature to synchronize the position and/or velocity between two spindles with feedback using your 9/440, 9/260, or 9/290 control.

Two types of synchronization are available:

- **Velocity** – synchronizes the speed between two spindles only
- **Velocity and Position** – synchronizes the speed and angular position between two spindles

Prior to activation, you are responsible for selecting the proper gear ranges and ratios. The gear ratio between the feedback device and the spindle must be 1:1. Any other type, including nonunit ratios, will not allow repeatability of the orientation of your spindle and may cause positioning offset inaccuracies.

Spindle Configuration

Your system installer selects two spindles to make up the synchronization pair, which consists of the controlling and follower spindles. During synchronization, the controlling spindle initiates spindle motion while the follower spindle synchronizes with it. Your system installer determines the configuration of these spindles. Refer to your system installer's documentation for more information about spindle configuration.

Gear ranges are set separately for each spindle. If the controlling spindle speed is outside of the current follower spindle gear range when a seek is attempted, the controlling spindle will ramp to within the follower's limits set in AMP.

Selecting the Controlling Spindle

The synchronized spindle's controlling spindle, which is determined by your system installer, must be programmed as the part program's controlling spindle in your part program prior to synchronization. Use one of the G12 codes (G12.1, G12.2, and G12.3) to designate the active controlling spindle for spindle synchronization. Refer to page 17-1 for more information about the G12 codes and your system installer's documentation to identify your controlling spindle.

Important: Typically, the programmed speed of the controlling spindle dictates the speed of the follower spindle. For more information about valid gear ranges, refer to page 17-11.

17.4.1 Using the Spindle Synchronization Feature

Use these three G-codes to manipulate the spindle synchronization feature:

- Set spindle positional synchronization (G46)– sets the follower spindle speed/direction and relative position offset to match the controlling spindle.
- Set active spindle speed synchronization (G46.1)– sets the follower spindle speed/direction to match the controlling spindle.
- Deactive spindle synchronization (G45)– shuts off synchronization while maintaining the controlling and follower spindles' current speed and direction.

Activate Spindle Positional Synchronization (G46)

Use the “Activate Spindle Positional Synchronization” to synchronize speed and position. The position is based on a programmed S-word (degrees). If you do not program an S-word in the G46 block, it will automatically go to the relative positional offset, set by your system installer. Refer to your system installer's documentation for more information.

During a G46, the spindles attempt to match speeds. Once the speeds are matched, the spindles attempt to synchronize their relative positional offset. Once synchronization is achieved, the active spindle speed and mode (M03, M04, M05, or M19) programmed for the follower spindle is replaced by the current controlling spindle speed and mode.

Important: Changes in spindle speeds that would normally occur as a result of CSS or other programmed changes to spindle speeds, directions, and spindle speed override will not occur until synchronization is achieved.

The format for the G46 block is as follows:

G46S__;

Where:	Defines:
S	the angular offset between two spindles (degrees)*

*The range for the S-word is 0 to 360. Your system installer defines where 0__ is for each spindle. The value defined as the offset value is absolute and independent of spindle direction.

Important: No other program letters are allowed in the G46 block except auxiliary letters and system installer M-codes.

The following example assumes that the controlling and follower spindles were defined as spindle 2 and spindle 1, respectively, by your system installer.

Example 17.2
Spindle Synchronization

M03 S200;	Spindle 1 clockwise 200 rpm
M04. 2 S400;	Spindle 2 counterclockwise at 400 rpm
G12. 2;	Spindle 2 as controlling spindle
G46 S90;	Spindle 1 changes direction and accelerates to spindle 2's speed; spindle 1 synchronizes angular position with spindle 2 (offset 90 degrees)

Activate Spindle Speed Synchronization (G46.1)

Use the “Activate Spindle Speed Synchronization” to synchronize speed and direction only. Using G46.1 does not guarantee a consistent positional offset between the two spindles. During a G46.1, the follower spindle attempts to synchronize speeds with the controlling spindle. Once synchronization is achieved, the current spindle speed and mode (M03, M04, M05, or M19) programmed for the follower spindle is replaced by the current controlling spindle speed and mode programmed. The original follower spindle speed and direction is not retained.

Important: Changes that occur as a result of CSS or other programmed changes to spindle speeds, directions, and spindle speed override will not occur until synchronization is achieved.

The format for the G46.1 cycle is as follows:

G46. 1;

Important: No other program letters are allowed in the G46.1 block except auxiliary letters and system installer M-codes.

Deactivate Spindle Synchronization (G45)

Use G45 to deactivate the synchronized spindle feature. When synchronization is deactivated, the follower spindle will remain in the same state (M03, M04, M05, or M19) and at the last programmed speed for controlling spindle until you change the program settings or if your system installer writes PAL to recommend the spindle.

In a multiprocess system, if the spindles are synchronized, and an end-of-program (M02) is reached on the process that is synchronizing the spindles, both spindles will obey the end-of-program commands. Then synchronization will be disabled.

The format for the G45 cycle is as follows:

G45;

Important: No other program letters are allowed in the G45 block except auxiliary letters and system installer M-codes.

17.5 Special Considerations for Spindle Synchronization

When using the synchronized spindle feature, remember:

- you cannot retrace through a synchronization block (G45, G46, or G46.1). However, you can retrace through blocks where synchronization was already active.
- in dual-process systems, both spindles used for synchronization must be configured in the process that is programming spindle synchronization. Shared spindles are acceptable if they are already controlled by the process requesting synchronization.
- gear changes are not allowed during synchronization. If spindle speeds exceed the gear range of either spindle, the spindles will be limited to the more restrictive spindle's values.
- due to the servo switch from open- to closed-loop during synchronization, a slight hesitation in the spindles may be seen when this switch occurs. This small deceleration may be more apparent in systems with a smaller spindle motor or if synchronization is done at higher speeds.
- Program Restart, Mid-Start, and Interrupt Macros will be allowed. If synchronization is disabled during an interrupt macro, it will resynchronize upon return, in the event that all of the condition checks listed in this section allow it to, otherwise a decode error will result. Midstart and restart must also pass all conditions described in this section.

- you are responsible for selecting proper gear ranges prior to activating synchronization.

The following features cannot be used while synchronization is active:

- solid-tapping
- virtual/cylindrical programming

The following features cannot be used while synchronization is ramping:

- threading
- deep-hole peck drilling

Important: Virtual C and threading are available on synchronized spindles once synchronization is achieved.

- When synchronization is active, any part program commands destined for the follower spindle (i.e., M03, M03.2, M03.3...G12.1, G12.2, and G12.3) will cause an error. On a multiprocess configuration, this is true of either process.
- On a multiprocess 9/Series, the process controlling the controlling spindle also controls the follower spindle when spindle synchronization is active. If it is unable to obtain control, an error results: UNABLE TO SYNCH IN CURRENT MODE.

Important: Typically, the programmed speed of the controlling spindle dictates the speed of the follower spindle. In the event that the programmed speed exceeds the maximum or drops below the minimum allowable values for the synchronized pair, the spindle speed will be restricted to those allowable values, as shown on page 17-11.

- the example below shows what will happen when:
 - no overlap occurs between the controlling and follower spindles' gear ranges
 - the controlling spindle has a higher gear range than the follower spindle
 - the controlling spindle has a lower gear range than the follower spindle

Example 17.1
Valid Gear Ranges for Synchronized Spindles

Controlling Spindle Gear Range (RPM)	Follower Spindle Gear Range (RPM)	Requested Spindle Speed (RPM)	Valid Programmed Spindle Speeds (RPM)	Spindles will Synchronize at (RPM):
1000 to 3000	100 to 300	1500	None	N/A
1000 to 3000	800 to 1500	1800	1000 to 1500	1500
1000 to 3000	1800 to 3200	1500	1800 to 3000	1800

END OF CHAPTER

Programming Feedrates

18.0 Chapter Overview

This chapter describes how to program feedrates and acceleration/deceleration. Use this table to find the information in this chapter:

Information about:	On page:
Feedrates	18-1
Special AMP Assigned Feedrates	18-12
Automatic Acceleration/Deceleration	18-14

18.1 Feedrates

Feedrates are programmed by an F-word followed by a numeric value. Feedrates can be entered in a part program block or through MDI. They become effective in the block in which they are programmed. If the block requires rapid traverse motion (G00), the programmed feedrate will be ignored for that block, but will be stored in control memory as the active feedrate.

Feedrates are modal, meaning that they remain active in control memory unless replaced with a different feedrate programmed with an F-word.

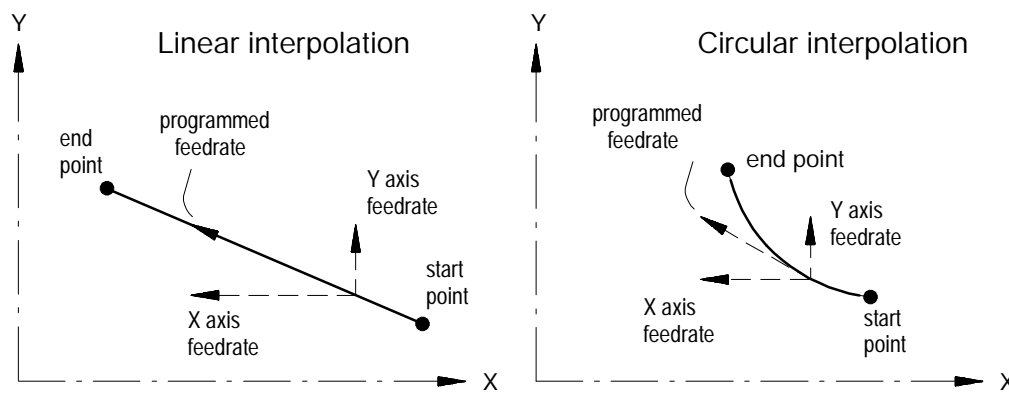
The feedrate programmed with the F-word applies to linear interpolation (G01), and circular interpolation (G02/G03) modes. For simplification this manual refers to the cutting tool moving relative to the part, even though most mills move the part across a stationary cutter.

Feedrate modes are either G93 (inverse time feed), G94 (feed per minute), or G95 (feed per spindle revolution). The following table shows the possible feedrate units depending on axis type.

Active G Code	Linear Axis Feed	Rotary Axis Feed
G20/G21 and G93	moves/min.	moves/min.
G02/G03 and G93	radians/min.	moves/min.
G21 and G94	millimeters/min.	degrees/min.
G20 and G94	inches/min.	degrees/min.
G21 and G95	millimeters/rev.	degrees/rev
G20 and G95	inches/rev.	degrees/rev.

Feedrates for linear and circular interpolation are “vector” feedrates. That is, all axes move simultaneously at independent feedrates so that the rate along the effective path is equal to the programmed feedrate (see Figure 18.1).

Figure 18.1
Feedrate Tangent To Tool Path



For example, if a feedrate is programmed as F100.0 millimeters per minute and a linear move is made from X0, Y0 to X10., Y10. the feedrate along that 45 degree angular path would be 100.0 mmpm. The actual feed rate of each axis will be approximately 70.7 millimeters per minute.

18.1.1 Feedrates Applied During Cutter Compensation

When the cutting tool is offset from a programmed path, (as in the case of cutter compensation) the programmed feedrate is applied to the center of the tool radius for all linear and outside arc paths. This discussion deals with the speed at which the outside surface of the tool passes across the workpiece surface disregarding tool rotation speed.

For linear paths, the result is not significant because the speed of the outside surface of the tool relative to the part surface remains the same as the programmed feedrate.

For outside arc paths, the resulting speed of the outside surface of the tool relative to the part surface is less than the programmed feedrate. This generally causes no problem and so the control does not take corrective action.

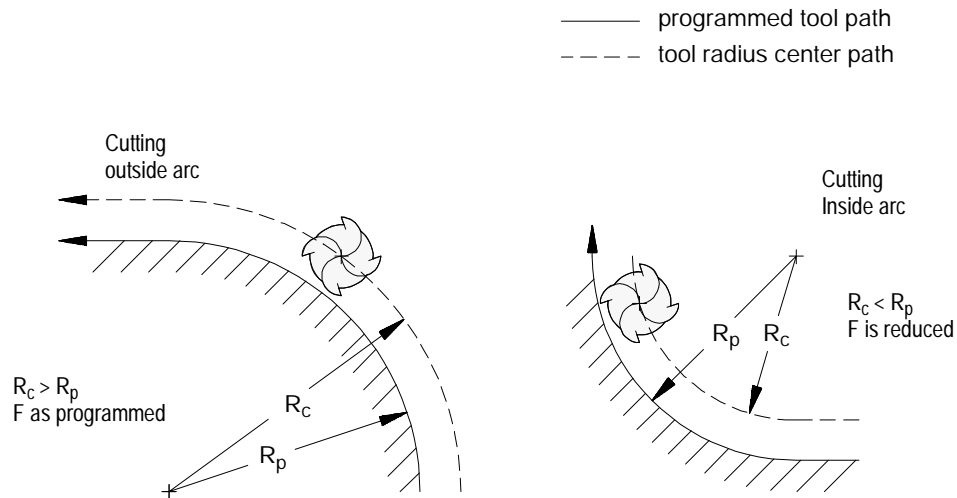
For inside arc paths, the resulting speed of the outside surface of the tool relative to the part surface would be greater than the programmed feedrate. Since this could cause excessive tool loading and poor cutting performance, the control automatically decreases feedrate.

For outside arc paths, the speed of the outside surface of the tool relative to the part surface can be determined using the following formula:

$$\text{Tool surface speed} = F \times \frac{R_p}{R_c}$$

Where :	Is :
F	The programmed feedrate
R _c	The radius of the arc measured to the center of the tool radius
R _p	The programmed radius of the arc

Figure 18.2
Inside and Outside Arc Feedrates with Cutter Compensation



For inside arc paths, the control will automatically maintain the programmed feedrate at the outside surface of the tool. The actual tool radius center feedrate will be reduced as needed through the arc path and then returned to the programmed feedrate after the arc is completed.

During inside arc paths, the control decreases the tool radius center feedrate by the ratio of R_c/R_p . If the R_c value is very small compared to R_p , as in the case of a small arc being cut with a large diameter tool, the value of R_c/R_p will be nearly zero, and the tool radius center feedrate will become excessively small.

To avoid this problem, the system installer must set a minimum feed reduction percentage (MFR) in AMP. This will set a minimum feedrate to be used whenever the value of R_c/R_p is very small. If R_c/R_p control will reduce the tool radius center feedrate no more than the MFR percentage.

18.1.2 Inverse Time Feed Mode (G93)

In G93 (inverse time feed) mode, the F-word represents the amount of programmed axis or axes motion that will be completed in a minute (moves per minute). For example, if a G93 block were programmed with a move to X20 (from X0) and a feedrate of 0.2, the X axis would move 0.2 of the move (to X4) in one minute.

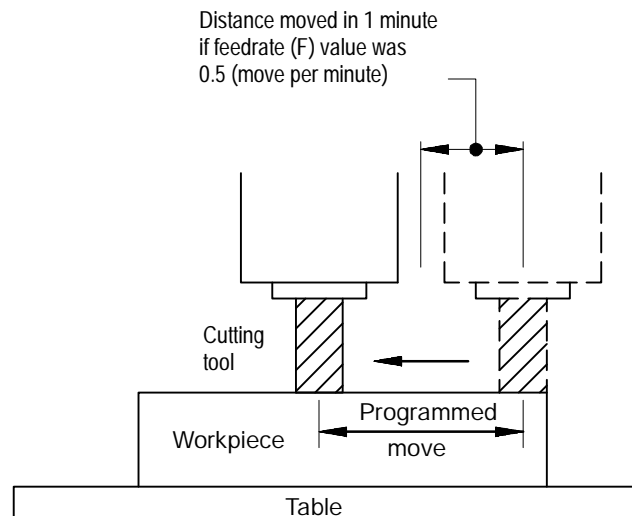
Another way of describing this would be to refer to the F-word as the inverse of the amount of time to complete a programmed move. If it were desired to complete the move described above in 2 minutes, then the F-word value would be $1/2$ (program 0.5).

If the programmed move is a circular move (G02 or G03 modes), then the F-word represents the number of radians of the arc or circle to be moved per minute (radians per minute).

In G93 mode, an F-word is effective only in the block it is programmed (F is not modal in G93). Therefore, all blocks that call for axis motion when G93 is active, must contain an F-word or an error occurs.

G93 is modal. It remains active until cancelled by a G94 (feed per minute) or G95 (feed per revolution).

Figure 18.3
Inverse Time Feed (G93)



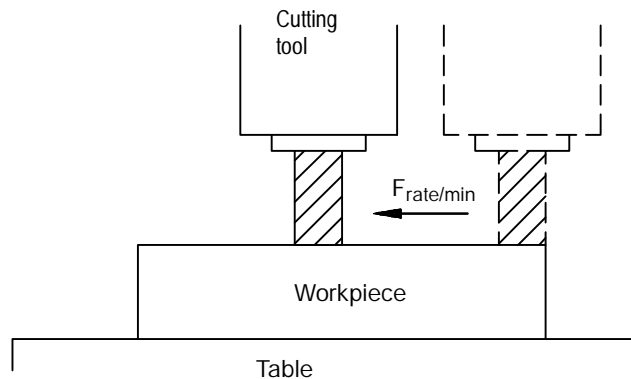
18.1.3 Feed-Per-Minute Mode (G94)

In the G94 mode (feed-per-minute), the numeric value following address F represents the distance the axis or axes move (in inches or millimeters) per minute. If the axis is a rotary axis, the F-word value represents the number of degrees the axis rotates per minute.

To request a feedrate of 35.5 mm of tool motion per minute, program:

```
G94 G21 F35.5;
```

Figure 18.4
Feed Per Minute Mode (G94)



When changing from G93 or G95 modes to G94 mode, an feedrate must be programmed in the initial G94 block.

Since the G94 code is modal, it remains active until canceled by the G93 or G95 mode. Any F-word programmed while G94 mode is active will be considered a feed per minute feedrate.

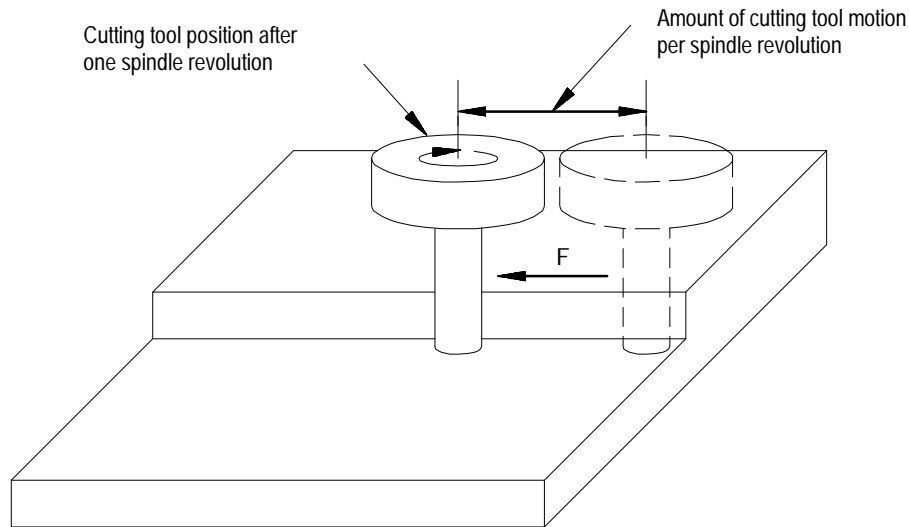
18.1.4 Feed-Per-Revolution Mode (G95)

In G95 (feed-per-revolution) mode, the numeric value following the F-address represents the distance the axis or axes move (in inches or millimeters) per spindle revolution. If the axis is a rotary axis, the F-word value represents the number of degrees the axis rotates per spindle revolution.

To request a feedrate of 1.5 mm per spindle revolution, program:

```
G95 G21 F1.5
```

Figure 18.5
Feed Per Revolution Mode (G95)



When changing from G93 or G94 modes to G95 mode, an F-word must be programmed in the initial G95 block.

Since the G95 code is modal, it remains active until canceled by the G94 mode. It is also temporarily cancelled during execution of a G93 block. Any F-word programmed while G95 mode is active will be considered a feed per minute feedrate.

18.1.5 Rapid Feedrate

Certain axis motions request from the control a rapid feedrate. For example, the G00 and some of the fixed cycles call for the rapid feedrate. The system installer specifies the rapid feedrate individually for each axis in AMP. When executing using a rapid feedrate the control drives the axes to maintain the fastest possible linear move while still remaining under (or at) each axis rapid feedrate.

When positioning mode is active, any programmed F-word executed by the control is stored as the currently active cutting feedrate. The rapid feedrate will not be affected.

18.1.6 Feedrate Overrides

Feedrate Override Switch

Feedrates programmed in any of the feedrate modes (G93/94/95) can be overridden using the feedrate override switch on the MTB panel. The feedrate override switch has a range of 0-150 percent of the active feedrate, and can alter the active feedrate in 10 percent increments.

The feedrate override switch operates on the feedrate that is active. The active feedrate may be less than the feedrate you programmed if:

- the control has limited the feedrate to the maximum cutting feedrate that is defined in AMP

or

- the Acc/Dec feedrate clamping has limited the active feedrate to a value that the axis can decelerate from in the current block. See the section on Short Block Acc/Dec on page 18-22.

The control checks whether the feedrate resulting from the feedrate override switch setting exceeds the maximum cutting feedrate set in AMP. If it does, the feedrate is restricted to the AMP maximum.

An M49 (overrides disabled) causes the override amounts that are set by the switches or buttons on the MTB panel to be ignored by the control. With M49 active, the override switches or buttons for feedrate, rapid feedrate, and spindle speed are all set to 100 percent. They can be enabled by programming an M48 (overrides enabled).

Rapid Feedrate Override

Rapid feedrate override on the MTB panel is used to override the rapid feedrate for G00 mode in four increments:

- F1 -- percent value set in AMP by system installer
- 25%
- 50%
- 100%.

Important: Normally this override is not active for any dry run motions unless otherwise specified in PAL.

Important: This override is also effective for jog moves that use the rapid feedrate (jogging using the {TRVS} button chapter 4).

Feedrate Override Switch Disable

An M49 causes the override amounts that are set by the switches on the MTB panel to be ignored by the control. With M49 active, the override switches for feedrate, rapid feedrate, and spindle speed are all set to 100 percent. They can be enabled by programming an M48 (overrides enabled).

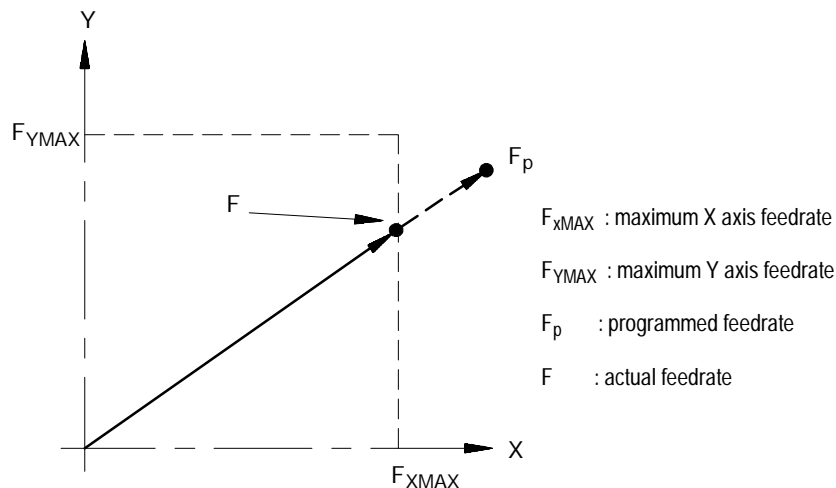
Feedhold

The system installer may have written PAL to allow the activation of a feedhold state through the use of a button or switch. When activated the control will decelerate and hold the current feedrate for all axes to zero until the feedhold state is deactivated. For details on using feedhold refer to documentation provided by the system installer.

18.1.7 Feedrate Limits (Clamp)

The system installer can set the allowed maximum cutting feedrate for each axis individually in AMP. If a feedrate is programmed that would cause an individual axis to exceed its limit, the control automatically reduces the feedrate for the tool to keep the limiting axis at its maximum cutting feedrate. This function is called feedrate clamping.

Figure 18.6
Feedrate Clamp



The programmed feedrate, F_p in Figure 18.6, would cause the X axis feedrate to exceed its maximum feedrate, F_{xmax} . consequently, the control uses F as the actual feedrate.

The maximum cutting feedrate limits the axis feedrate for any move controlled by a F-word. Feedrate override switch settings that cause the feedrate to exceed the maximum cutting feedrate will also be accordingly modified to keep the feedrate below or at the maximum cutting feedrate.

When the feedrate is “clamped” to a value below the programmed feedrate the control displays a flashing C next to the current axes feedrate. The displayed axis feedrate is the actual feedrate of the tool, not necessarily the programmed feedrate.

18.2 Feedrates to Control Torque Adaptive Feed (G25)

This feature will cause the servo to maintain a constant torque while cutting by varying the axis feedrate. The adaptive feed feature is helpful for machining processes where maintaining a constant feedrate is not as important as maintaining a constant cutting force. The adaptive feed feature also allows you to enter a feedrate range to prevent excessively low or high feedrates.

Your system installer must configure the adaptive feed feature for each servo using AMP.

When a move is programmed as an adaptive feed move, the control monitors the control’s torque output being sent to the servo drive. If the output torque is too low, the control increases the cutting feedrate until the output torque reaches the programmed torque. If the torque output is too high, the control decreases the cutting feedrate until the servo output drops to your programmed torque.

Programming G25 Adaptive Feed

Program a G25 block as follows:

$$G25 \left(\begin{array}{c} X_ \\ Y_ \\ Z_ \end{array} \right) Q_ F_ E_;$$

Where:	Programs:
X, Y, or Z	Axis endpoint. Program the endpoint of the axis that is to be positioned using the adaptive feed feature. This endpoint can be programmed as either an absolute or incremental value (G90 or G91 mode). You can only program one axis in an adaptive feed block. You can not program axes that are positioned by more than one servo (dual or deskew axes).
Q	Desired Torque percentage. Enter an integer percentage of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1% to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. Important: The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
F	Maximum Feedrate. Enter the maximum feedrate that the axis is allowed to reach during the adaptive feed move. This F-word is the same as the modal feedrate and is used in following blocks. If you do not program an F-word the control will use the previously active feedrate. Program the F-word in either G93, G94, or G95 mode.
E	Minimum Feedrate. Enter the minimum feedrate that the axis is allowed to reach during the adaptive feed move. Program the E-word in either G93, G94, or G95 mode.

You must program the G25 block in G01 mode. Programming G25 in other cutting modes (such as G00, G02, or G03) will generate an error. The G25 command is not modal and must be programmed in all blocks that are to use the adaptive feed feature.

When the adaptive feed move starts, the control issues a command that moves the axis with the desired torque percentage Q. The system installer determines in AMP if ACC/DEC is used for the start and stop of an adaptive feed move. If ACC/DEC is used the control will attempt to ramp the velocity command to minimize the initial start up and decel shock.

Adaptive Feed Maximum Feedrate

When cutting under low to no load the servo may not be able to reach the programmed torque without exceeding your programmed F-word. In these cases, once the maximum servo feedrate is reached, the control allows the torque to drop below your programmed torque so as to not exceed the maximum programmed axis feedrate (F). The error “Adaptive Feed Max Limit” is displayed on the CRT.

Adaptive Feed Minimum Feedrate

When cutting under high loads the servo output torque may not be enough to keep the axis from slowing down below your programmed E-word. In these cases, once the minimum programmed feedrate is reached, the control allows the torque to go above its programmed limit to maintain the programmed minimum feedrate (E). The error “Adaptive Feed Min Limit” is displayed on the CRT.

Feedrate Override

Because of the precise feedrate control necessary to maintain a constant torque the feedrate override switch is disabled for any move that is programmed using the Adaptive Feed feature. If PAL is written to do so, you can use this switch only to request a feed hold which will halt axis motion. Cycle stop will also halt axis motion.

Block Retrace

You can perform block retrace on an adaptive feed block. When an adaptive feed block is retraced adaptive feed control is made in the opposite direction of the originally programmed block. Typically since no cutting is performed on the block retrace the cutting tool will move at the programmed maximum feed rate (F).

Polar Programming

You can not program a G25 block if the axis being programmed is in the current plane and the control is in polar programming mode (G16).

18.3 Special AMP Assigned Feedrates

It is possible to select special feedrates that are assigned in AMP. This covers the feedrates assigned by AMP for the single digit F-word and the External feedrate switch. It does cover the feedrate for rapid moves or for dry run.

18.3.1 Single Digit F- words

Program a one-digit numeric value (1-9) following the F-code to select various preset feedrates. The digits 1-9 have corresponding numeric values for feedrate set in the F1-F9 table discussed in chapter 2.

Important: Single digit feedrates are always entered as per minute feedrates (IPM or MPPM) regardless of the control's current feedrate mode. When a single digit feedrate is programmed, the control automatically switches to the IPM or MPPM mode. The control automatically switches back to the previously active feedrate mode when the next feedrate is programmed that is not a single digit feedrate.

If there are no feedrates set in the tables that correspond to F1-F9, the control treats F1-F9 as commanded feedrates.

Single digit F-word feedrates are affected by the setting of the switch as regular feedrates. Blocks that are programmed to move at the rapid feedrate are still executed in the rapid mode.

Important: A programmed F0 corresponds to the maximum cutting feedrate (not normally the rapid feedrate) set in AMP by the system installer. F0 generates an error if the single digit feedrate feature is not used.

When calling a single digit F-word the F-word must be programmed with no decimal point and a single digit. For example, F1 calls an assigned feedrate correlating to the value set as feedrate 1. Designating F1 in a program assigns a feedrate of 1 in the current feedrate mode.

18.3.2 External Feedrate Switch

The system installer may install an optional external deceleration switch if desired. Typically, this is a mechanical switch mounted on the machine axes inside the hardware overtravel switches (refer to documentation prepared by the system installer for details on the application and location of this switch).

When this feature is active any axis moves that are to take place at a cutting feedrate (e.g., G01, G02, G03) use a special feedrate assigned in AMP. When this feature is active any axis moves that are to take place at a rapid feedrate (e.g., G00) also uses a special feedrate assigned in AMP. These feedrates are independent of each other and typically have different values. These feedrate changes take place immediately when the feature becomes active, even if this is in the middle of block execution.

Important: The feedrate set for the external deceleration feature for cutting moves can not exceed the maximum cutting feedrate.

If this feature is used simultaneously with the Dry Run feature the feedrates that are assigned to External decel feature are used. The feedrates for this feature are not related to the Dry Run feedrates although the operation of this feature is similar to Dry Run.

This feedrate is unaffected by the feedrate override switch, and the rapid override switch settings and operates as if the switches are set at 100 percent. Blocks that are programmed to move at the rapid feedrate are still executed in the rapid mode.

Typically this feature is used to protect the machine from harsh or sudden stops. If a very high feedrate is active at the time that a hardware overtravel occurs damage to the machine may result if a sudden stop is performed at the overtravel or the machine may coast past a safe range for axis motion. If the switch is installed before the overtravel area the feedrate of the move is reduced and the amount of coast into the overtravel area will be much less.

If the current feedrate is less then the feedrate set for the external deceleration feature, it will be accelerated to the external deceleration feedrate. This may cause problems with part finish or may damage the tool if this feedrate is higher than that which the part should be cut.



WARNING: It is possible for the system installer to write PAL to allow the operator to select the external deceleration feedrate at any time. This means that during normal automatic operation the operator can select external decel and replace all feedrates in the program with the external decel feedrates. This may result in damage to the machine, part, or injury to the operator.

18.4
Automatic
Acceleration/Deceleration
(Acc/Dec)

There are three types of axis acceleration/deceleration available:

- Exponential Acc/Dec
- Uniform or Linear Acc/Dec
- S-Curve Acc/Dec

These are used to produce smooth starting and stopping of the machine's axes and prevent damage to the machine resulting from harsh movements.

Your system installer determines the acc/dec parameter type (exponential or linear) for some manual motion types. To determine which motion types are configurable, refer to the following table. Refer to your system installer's documentation for more information about how your system is configured.

Refer to the table below to determine the type of acceleration/deceleration performed for manual motion and programmed moves.

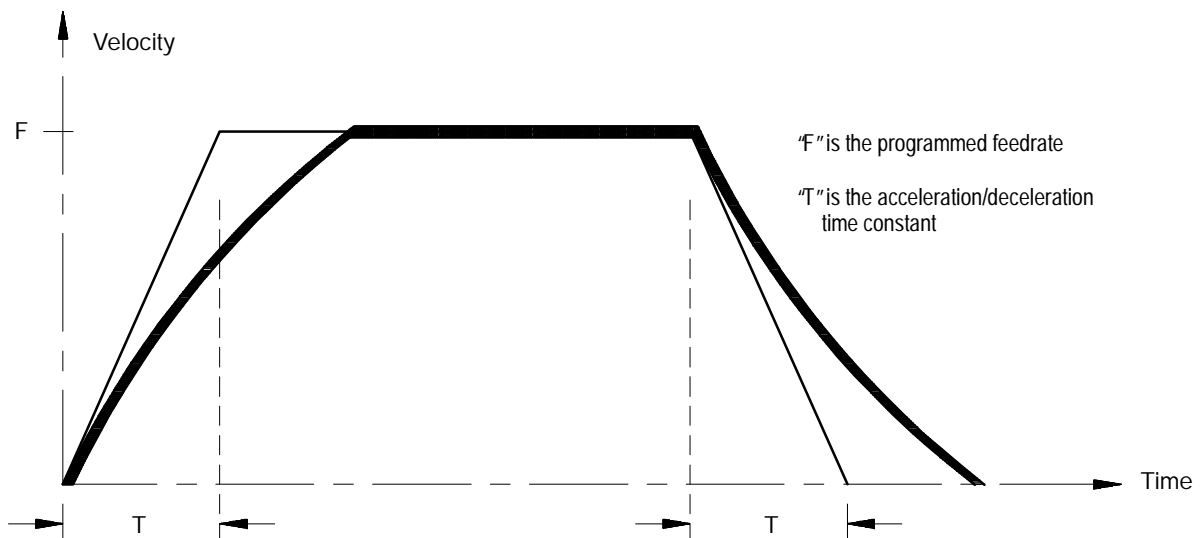
Table 18.A
Acc/Dec Type Performed with Manual Motion and Programmed Moves

Motion Type	Always Uses Exponential Acc/Dec	Configurable in AMP by System Installer via Manual Acc/Dec Mode	Always Uses Linear Acc/Dec	Linear or S-Curve Acc/Dec per G-code
Hand-pulse generator	n			
Arbitrary angle moves (i.e., hand-pulse generator and continuous and incremental motion)	n			
Homing	n			
All programmed moves except for G00 and exact stop			n	
Manual continuous motion		n		
Manual incremental motion		n		
PAL axis mover		n		
All moves programmed in G00 (positioning) mode				n

18.4.1 Exponential Acc/Dec

To begin and complete a smooth axis motion, the control uses an exponential function curve to automatically accelerate/decelerate an axis. The system installer sets the acceleration/deceleration time constant "T" for each axis in AMP. Figure 18.7 shows axis motion using exponential Acc/Dec.

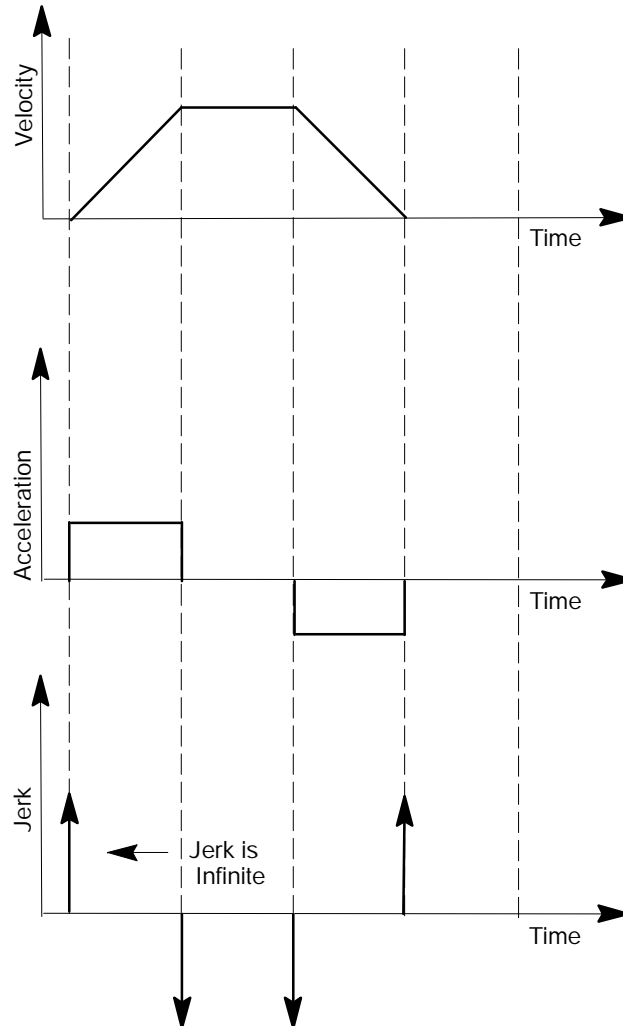
Figure 18.7
Exponential Acceleration/Deceleration



18.4.2
Linear Acc/Dec

Axis motion response lag can be minimized by using Linear Acc/Dec for the commanded feedrates. The system installer sets Linear Acc/Dec values for interpolation for each axis in AMP. Figure 18.8 shows axis motion using Linear Acc/Dec.

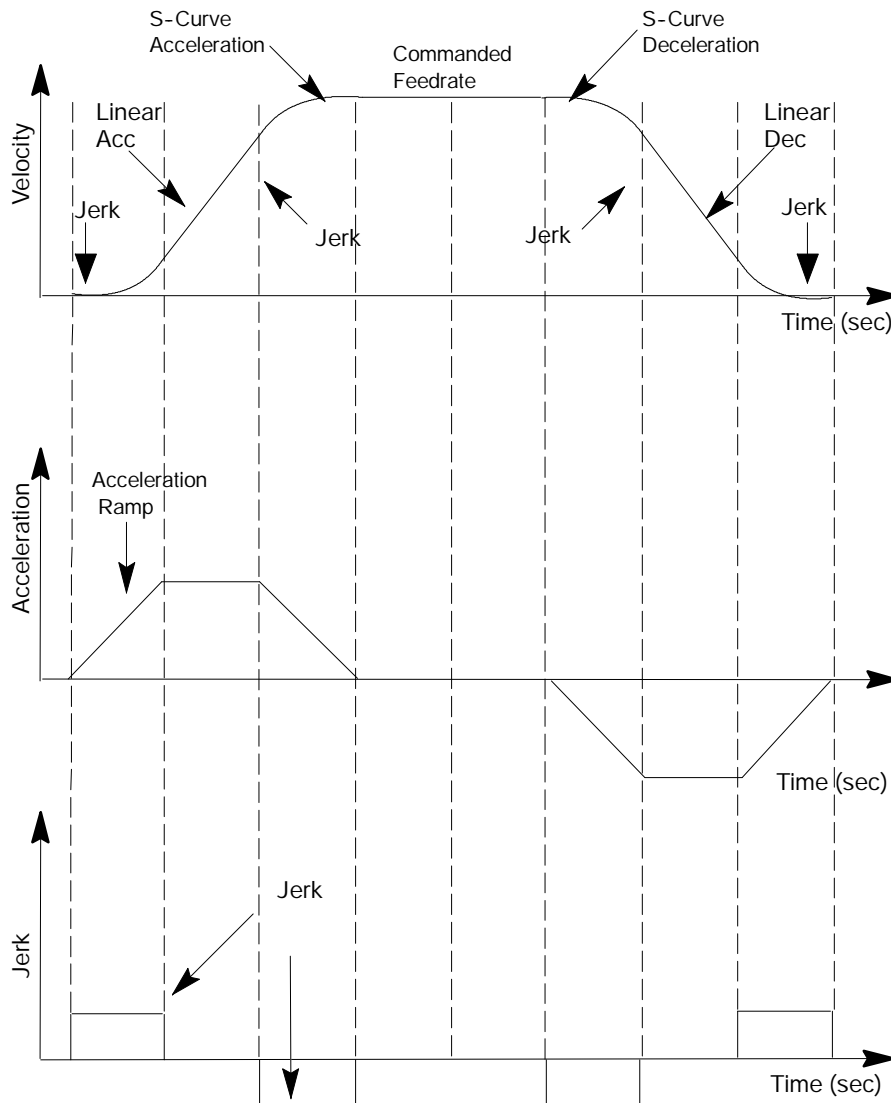
Figure 18.8
Linear Acc/Dec



18.4.3
S-Curve Acc/Dec

When S-Curve Acc/Dec is enabled, the control changes the velocity profile to have an S-Curve shape during acceleration and deceleration when in Positioning or Exact Stop mode. This feature reduces the machine's axis shock and vibration for the commanded feedrates. Figure 18.9 shows axis motion using S-Curve Acc/Dec.

Figure 18.9
S-Curve Acc/Dec



18.4.4 Programmable Acc/Dec

Programmable Acc/Dec allows you to change the Linear Acc/Dec modes and values within an active part program via G47.x and G48.x codes.

You cannot retrace through programmable acc/dec blocks (G47.x and G48.x). However, you can retrace through blocks where programmable acc/dec was already active.

Selecting Linear Acc/Dec Modes (G47.x -- modal)

Programming a G47.x in your part program allows you to switch Linear Acc/Dec modes in nonmotion blocks. If S-Curve Acc/Dec is active, all positioning moves within fixed cycles will use this mode.

- G47 - Linear Acc/Dec in All Modes
- G47.1 - S-Curve Acc/Dec for Positioning and Exact Stop Mode Only
- G47.9 - Infinite Acc/Dec (No Acc/Dec) (Enabled by your system installer in AMP)

Important: For optimum S-Curve Acc/Dec functionality, any block preceding a G47.1 block will decel to 0.

The table below shows you the interaction between contouring, positioning, exact stop moves, and acc/dec type (i.e., linear, exponential, S-Curve, and disabled).

Table 18.B
Interaction Between Contouring, Positioning, Exact Stop, and Acc/Dec Modes

Programming:	In this mode will result in:			
	G00	G01	G02	G03
G47	Linear/ Exponential ¹	Linear	Linear	Linear
G47 & G09/G61	Linear/ Exponential ¹	Linear	Linear	Linear
G47.1	S-Curve/ Exponential ²	Linear	Linear	Linear
G47.1 & G09/G61	S-Curve/ Exponential ²	S-Curve	Linear	Linear
G47.9	Disabled	Disabled	Disabled	Disabled
G47.9 & G09/G61	Disabled	Disabled	Disabled	Disabled

¹Linear/Exponential is a function of Positioning Acc/Dec. If Exponential is AMPed, this is the acc/dec type, otherwise, the type is Linear.

²S-Curve/Exponential is a function of Positioning Acc/Dec. If Exponential is AMPed, this is the acc/dec type, otherwise, the type is S-Curve.

Selecting Linear Acc/Dec Values (G48.n - - nonmodal)

Programming a G48.x in your part program allows you to switch Linear Acc/Dec values in nonmotion blocks. Axis values in G48.n blocks will always be treated as absolute, even if the control is in incremental mode.

Below is the format for calling G48 commands. Use this format with the axis names assigned by your system installer:

G48. n X_Y_Z_

Where :	In this mode :	Units of measure :	Sets up :	Macros :
XYZ	G48	N/A	acceleration, deceleration, and jerk ramps back to the AMPed values Important: All axis words in a G48 are ignored.	N/A
XYZ	G48.1	in./sec ² or mm/sec ²	acceleration ramps for Linear Acc/Dec mode	#5631 to 5642
	G48.2	in./sec ² or mm/sec ²	deceleration ramps for Linear Acc/Dec mode	#5651 to 5662
	G48.3	in./sec ² or mm/sec ²	acceleration ramps for S-Curve Acc/Dec mode	#5671 to 5682
	G48.4	in./sec ² or mm/sec ²	deceleration ramps for S-Curve Acc/Dec mode	#5691 to 5701
	G48.5	in./sec ³ or mm/sec ³	jerk limits	#5711 to 5722

Important: The allowable programmed range for the axis word depends on the configured format. If you exceed these allowable ranges set by your system installer, you may use paramacros to override this limit.

For example, if the allowable programmed range for the axis word is 3.4 (e.g., 999.9999 max input) and the desired jerk limit is 100,000 mm/sec³, you may set Paramacro #1 to 100,000 and program a G48.5 X#1 to set the jerk limit to 100,000. This method can be used for any of the G48 programming blocks.

Example 18.1
Allowable Programmed Range

#1 = 100000;

G48.5 X #1;

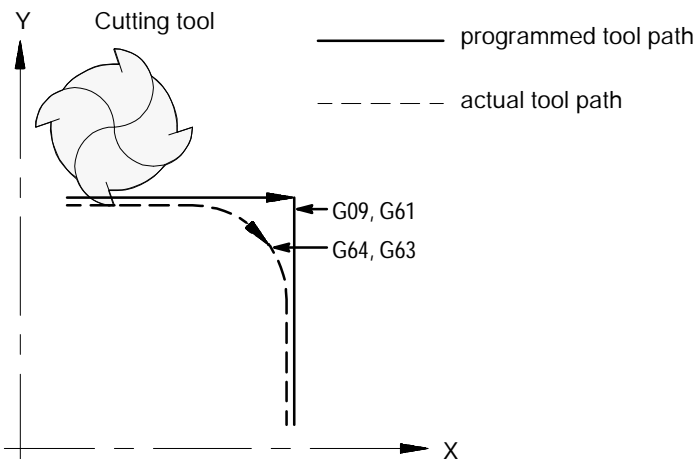
Important: The part program G48.n adjustments to Acc/Dec Ramps are not applied to jog moves. The AMPed Linear Acc/Dec mode rates are used when Manual Acc/Dec mode is linear.

18.4.5 Precautions on Corner Cutting

When Acc/Dec is active, the control automatically performs Acc/Dec to give a smooth acceleration/deceleration for cutting tool motion.

However, there are cases in which Acc/Dec can result in rounded corners on a part during cutting. In Figure 18.10 this problem is most obvious when the direction of cutting changes from the X axis to the Y axis. In this case, the X axis decelerates as it completes its move while the Y axis is at rest. As soon as the X axis reaches the AMP defined in-position band, the Y axis begins accelerating to make its commanded move. Since the Y axis begins motions before the X axis finishes, a slight rounding results.

Figure 18.10
Rounding of Corners



These two G codes can be used to eliminate corner rounding.

Exact Stop (G09 -- nonmodal)

If a programmed motion block includes a G09, the axis will move to the commanded position, decelerate, and come to a complete stop before the next axis motion block is executed. The G09 can be programmed in rapid (G00), feedrate (G01), or circular (G02/G03) motion blocks, but is active only for the block in which it is programmed.

Exact Stop Mode (G61 -- modal)

G61 establishes the exact stop mode. The axes move to the commanded position, decelerate and come to a complete stop before the next motion block is executed. Cancel this code by programming G61, G62 or G63.

Cutting Mode (G64 -- modal)

G64 establishes the cutting mode. This is the normal mode for axis motion and will generally be selected by the system installer as the default mode active on power up. Block completes when the axes reach the interpolated endpoint. Cancel this code by programming G61, G62, or G63.

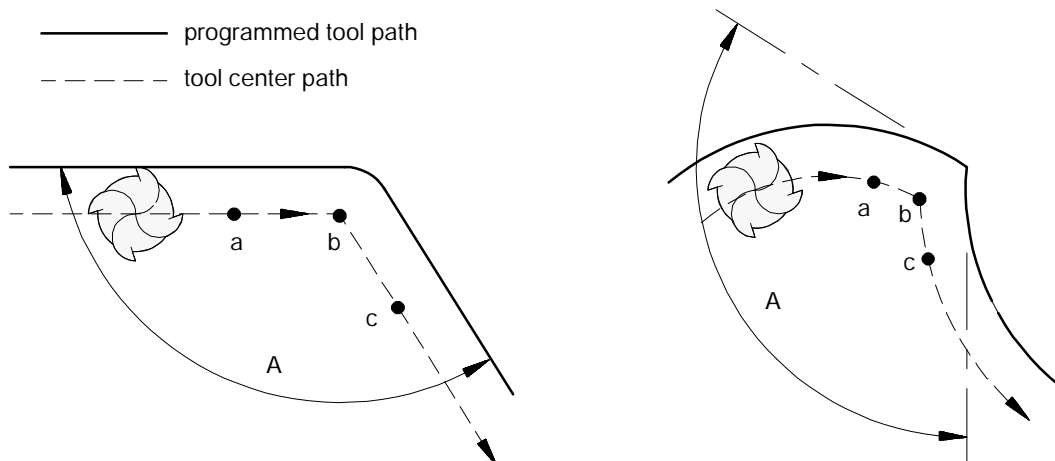
Tapping Mode (G63 -- modal)

In the G63 tapping mode, the feedrate override value is fixed at 100 percent, and a cycle stop is ignored. Axis motion commands are executed without deceleration before the end point. The program proceeds to the next block without checking in position status, similar to the operation of G64. Cancel this code by programming G61, G62, or G63.

Automatic Corner Override (G62 -- modal)

In cutter compensation mode (G41/G42), the load on the cutter increases while moving inside a corner. If the G62 automatic corner override mode is active, the control will automatically override the programmed feedrate to reduce the load on the cutter. Cancel this code by programming G61, G62, or G63.

Figure 18.11
Automatic Corner Override (G62)



When the corner angle, A , is smaller than angle A_p set in AMP, the programmed feedrate is overridden from point “a” to point “b”, and from point “b” to point “c”. The control compares angles A and A_p .

The system installer sets these values in AMP:

- angle A_p in AMP in 1 degree increments within a range of 1-90 degrees
- range in which the automatic corner override function is active -- essentially, the values of “a” and “c” in absolute distance measured along the tool path for “b”
- override value in 1-percent increments within a range of 1-100 percent.

To use an exact stop function while the automatic corner override mode (G62) is active, use the G09 instead of the G61. This is because G61 and G62 belong to the same G modal group and will cancel each other if programmed. Be aware that G09 is non-modal.

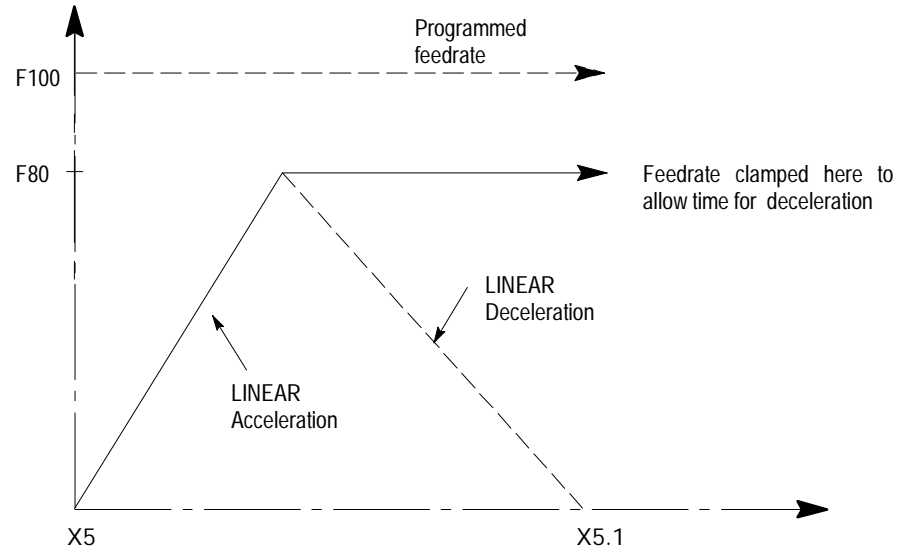
18.4.6 Spindle Acceleration (Ramp)

The system installer has the ability to change the rate in which a spindle is accelerated. AMP allows the option of setting RAMP to occur in 2 ms intervals during one system scan or as an immediate step. By writing the appropriate PAL, the system installer may also, in effect, generate a spindle “ramp” for even longer spindle acceleration ramps. Refer to documentation prepared by the system installer.

18.4.7 Short Block Acc/Dec G36, G36.1

In the control’s default mode (G36), the Acc/Dec feature in some cases limits the feedrate below your programmed feedrate. This occurs when the length of the block is relatively short relative to the length of time necessary to properly decelerate the axis to a stop. In the default mode (G36), the control limits the feedrate of any block to the maximum speed from which it can properly decelerate all the axes to a stop before that block ends. For example, consider the following velocity profile of a block moving from X5 to X5.1.

Figure 18.12
Feedrate Limited Below Programmed Feedrate to Allow Deceleration Time



For normal programming, this typically causes no problem. However, in cases where a series of very short blocks exist, the limitation to the feedrate may cause finish problems as well as increased cycle time.

To avoid the feedrate limitation, the short block Acc/Dec feature can be enabled by programming a G36.1. In this mode, the control will assume that no drastic deceleration is forthcoming and allow the control to raise the velocity of an axis higher than it could decelerate the axis before the block ends. This method of programming is only acceptable when:

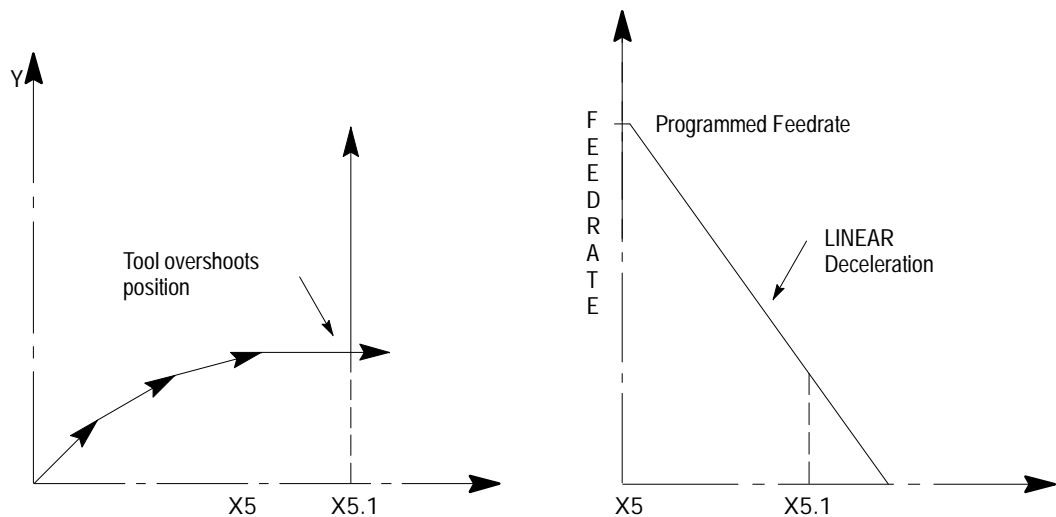
- no sudden changes in programmed feedrate within the short blocks exists (this includes requesting a feedhold or cycle stop)
- no drastic change in programmed direction is present within the short blocks exists

If any of the above considerations are not met during the G36.1 mode, the control will overshoot positions, since the axis will not have time to decelerate. For example, consider the following velocity curve if a drastic change in direction is requested after the move from X5 to X5.1 when in G36.1 mode. Note that the position X5.1 is overshoot, and the axis must reverse direction to reach proper position.

Figure 18.13
Drastic Change in Direction While in Short Block Mode

Tool overshoots end-point of move because of drastic change in block direction during short block mode.

Axis begins to decelerate at start of block but is traveling too fast to fully decelerate before end-of-move.



CAUTION: The programmer must take into consideration the direction of travel and change in feedrates of transitions from block-to-block in the short block mode (G36.1). If the transition exceeds the deceleration ramp of the axis, damage to the part or equipment can occur.

G36 and G36.1 are modal. The control should be in short block mode (G36.1) only when executing a series of fast short blocks that contains only slight changes in direction and changes in velocity. What constitutes a slight change in direction and velocity is dependent on the Acc/Dec ramp configured for your machine.

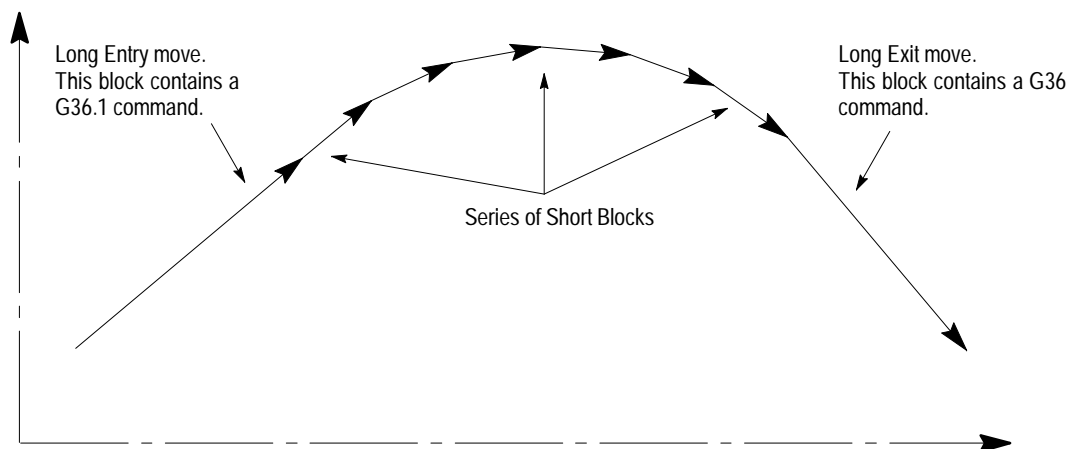
- G36 - Short Block Acc/Dec Disable
- G36.1 - Short Block Acc/Dec Enable

G36 is the default mode and is established at power up, E-STOP reset, and end-of-program (M02, M30, or M99). The recommended method of programming G36 and G36.1 is to program a relatively long entry and exit move into/out of the mode.

- The entry move should be a long move in the general direction of the first short block, and at the same feedrate as the first short block. This entry move should be long enough for the axes to reach programmed speed. Program the G36.1 code in this entry block.
- The exit move should be a long move in the general direction of the last short block, and at the same feedrate as the last short block. This exit move should be long enough for the axes to decelerate properly without overshooting its endpoint. Program the G36 code in this exit block.

The following example shows the recommended entry and exit moves for short block mode.

Figure 18.14
Entry and Exit Move to/from Short Block



END OF CHAPTER

Dual-axis Operation

19.0 Chapter Overview

This chapter describes how to program a dual axis. Use this table to locate specific information about dual axis operation:

Information about:	On this page
parking a dual axis	19-3
homing a dual axis	19-4
programming a dual axis	19-5
setting offsets for a dual axis	19-7

Important: This feature is not available on 9/230 CNCs.

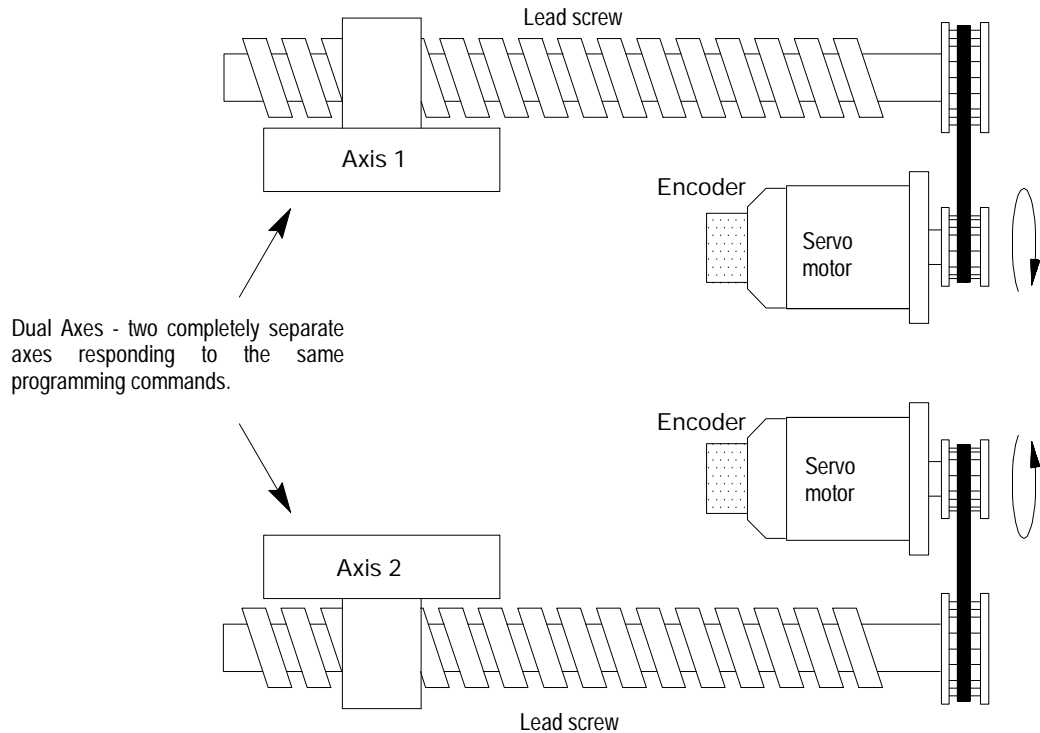
19.1 Dual-axis Operation

The Dual Axes feature allows the part programmer to simultaneously control multiple axes while programming commands for only one. It differs from the split axis feature of the control in that the split axis feature is used to control a single axis positioned by two servo motors.

The dual axes feature is especially useful for gantry mills with dual cutting heads and other machines running with parallel cutting tools. Figure 19.1 shows a typical configuration for dual axes.

Implementation of the dual axis feature can require significant PAL modification as well as proper AMP configuration. The dual axis feature is an option. Refer to your system installer's documentation to see if the dual axis option has been purchased for your machine.

Figure 19.1
Dual Axis Configuration



The control can support two dual axis groups. A dual axis group consists of two or more axes coupled through AMP and commanded by a master axis name. The master axis name is used by the part programmer or operator when commanding the dual axis group in part programs or for jog moves.

Each axis that makes up a dual group is controlled by a separate positioning command from the servo module. This dual group command is based on the move generated by the control when the master axis is commanded to a position.

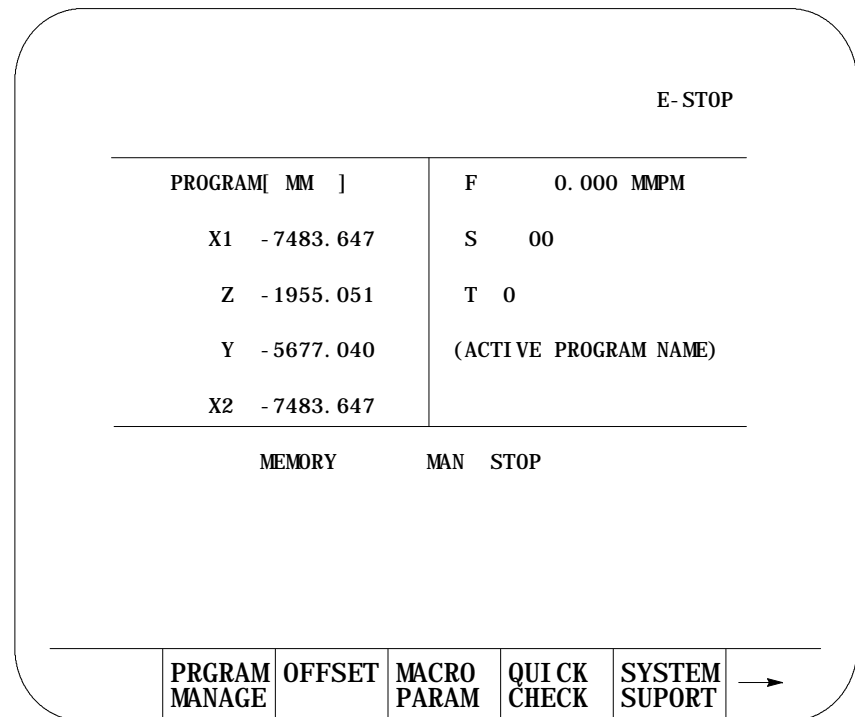
All axes that make up a dual group reach end-point at the same time. This requires that all axes that make up a dual axis group share the same feedrate parameters, acc/dec ramps, and other axes specific data for the group.

This section requires that you understand the following terms:

- Master Axis - A master axis is the name used to command the axes in a dual group.
- Dual Group - A dual group is a set of axes that are coupled together in AMP and commanded by a single master axis name.

Figure 19.2 shows the position display for a system that contains a dual axis group containing two axes with a master axis name of X. Whether or not all axes of a dual group show up on the position display is determined in PAL by the system installer.

Figure 19.2
Axis Position Display for Dual X Axis



19.1.1 Parking a Dual Axis

This feature allows the operator or programmer to disable selected axes of the dual group. Any axis that is a member of a dual axis group can be parked. Axes in the dual group may be parked simultaneously. If all axes in the group are parked, no motion can take place in the dual axis group.

Once parked, no motion is allowed on the parked axis. Programmed and jog commands (including any homing requests) made to the dual axis group are ignored by the parked axes.

Axes in the dual group may be parked or unparked only when the control is in cycle stop and end-of-block state. Also the control may not be in the process of completing any jog request or PAL axis mover request. If an attempt is made to park/unpark an axis and any one of the above requirements is not true, the control ignores the request to park/unpark the axis.



CAUTION: Care must be taken when an axis is unparked. When an axis is unparked, any incremental positioning requests made to the dual axis group are referenced from the current location of all axes in the dual group. This includes any manual jogging or any incremental part program moves. When an axis is unparked, we recommend the next command made to the dual axis group is an absolute command to re-align the axes in the dual group to the same position.

Parking an axis in a dual group is performed through PAL. Refer to the system installer's documentation for details on how axes are parked.

Important: Some systems may have special parking requirements when homing axes in a dual group.

19.1.2 Homing a Dual Axis

There are two different methods to home axes in a dual axis group. The system installer determines through PAL which method is available. The two methods are:

- home each axis in the dual group individually
- home all axes in the dual group simultaneously

Both of these homing methods can be available for automatic (G28) as well as manual homing operations.

The system installer can also define independent speeds and home positions for each axis in a dual group through AMP. This applies to both homing methods. Refer to the system installer's documentation for details on these speeds and locations.

Homing Axes Individually

This method requires that each axis be homed individually. When a manual home operation is performed, a home request must be made to each axis in the dual group individually. Refer to chapter 4 for details on how to request a manual home operation.

When using automatic homing (G28), the axes must be homed one at a time. This is accomplished by parking all other axes in the dual axis group except the axis that is to be homed and requesting the AMP assigned master axis name be homed in the G28 block. Once homed, that axis should be parked, the next axis to be homed should be unparked, and the homing procedure repeated. Refer to chapter 14 for details on how to request an automatic home operation (G28).

Homing Axes Simultaneously

This method allows a request for all axes in the dual group to be homed at the same time. This does not mean that all axes reach home at the same time. Keep in mind that the system installer may define different feedrates and different home positions for each axis in the dual group.

With proper PAL programming, the system installer can configure all axes in the dual axis group to home when the request is made using the master axis name. If this homing method is used, all unparked axes will home together. Refer to chapter 4 for details on how to request a manual home operation and chapter 14 on how to request an automatic home operation (G28).

19.1.3 Programming a Dual Axis

Axes in a dual axis group can be positioned using any of the normal programming or manual motion operations. To position a dual axis, only the master axis name can be requested. Requests to position a dual axis can be made in manual, automatic, or MDI mode.

For absolute and incremental moves, regardless of the start-point, each axis in the dual group reaches the requested position (or travel the requested distance) at the same time. For absolute moves, this means the individual axis feedrates may be modified, depending on the distance each axis must travel from start to end of the requested move.

The system installer can assign different maximum cutting, external decel, and rapid feedrate limitations for each axis in a dual axis group. The control uses the slowest feedrate for each of these features from any axis in the dual axis group.

Special consideration must be given when programming the following features:

Feature:	Consideration:
Mirror Imaging	Programmable mirror image is applied to all axes in the dual group. Manual mirror image, however, can be applied to each axis in the dual group individually. When manual mirroring is performed on selected axes in the dual group, positioning commands are in effect reversed from the programmed commands to the master axis. Manual mirror image is selected through PAL. Refer to the system installer's documentation and chapter 14 for details.
Scaling	When scaling, specify the scale factor for the master axis of the dual group. All other axes in the dual group are then scaled using the master axis scale factor. Refer to chapter 13 for details.
Digitizing	When generating program blocks using digitize, park all but one axis of the dual axis group. Use this unparked axis to generate the axis positions for the digitize feature. The blocks created are stored using the master axis name. Refer to chapter 5 for details.

Important: The PAL axis mover feature can be used if it is necessary to position dual axis group members separately without requiring any parking. Refer to your PAL reference manual for more information.

Invalid Operations on a Dual Axis

The following table contains a list of the features that are not compatible with dual axes. If it is necessary to execute one of these features on a dual axis, only the AMP master axis may be used. All other axes in the dual group must be parked. Refer to your system installer's documentation to determine which axis has been assigned in AMP as the master axis.

G code	Feature
G16	Polar Programming
G16.1	Cylindrical Interpolation
G31-G31.4	External Skip Functions
G37-G37.4	Automatic Tool Gauging Skip Functions
G38, G38.1	Probing Cycles
G76	Boring Cycle (with spindle shift)
G74.1, G84.1	Solid Tapping
G76	Boring Cycle w/Shift
G87	Back Boring Cycle
G88.1-G88.6	Pocket Milling Cycles
G89-G89.1	Irregular Pocket Milling

19.1.4 Offset Management for a Dual Axis

Consideration should be given to offsets used for a dual axis. In most cases, each axis can have independent offset values assigned to it. This section discusses the difference in operation of a dual axis when it concerns offsets. How to activate/deactivate and enter these offset values is not discussed here unless some change specific to a dual axis occurs. Refer to the section in chapter 4 of this manual discussing the offset you are using for implementation details.

Preset Work Coordinate Systems (G54-G59.3)

The operation of the work coordinate systems is functionally the same for a dual axis as any other axis. Each axis in the dual group can have its own independent value entered into the offset table. If you desire all axes in the dual group to have the same offset values, you must manually enter the same value for each axis in the dual group.

G52 Offsets

All axes in the dual group use the same value for the G52 offset regardless whether they are parked or not. When a G52 offset value is specified using the master axis name, each axis offsets its coordinate system incrementally by the G52 amount.

G92 Offsets

When a G92 offset value is specified using the master axis name, the current position of all axes in the dual group take on the location of the specified value. For example, if you have a dual axis named X and if it consists of two axes, X1 and X2, when programming the following:

```
G92X10;
```

the control causes the current positions of X1 and X2 to become 10 regardless of their current positions when the G92 offset is executed.

Different G92 offset values may be created for each axis if necessary. This is accomplished by performing a jog offset or by using the PAL axis mover to change the position of the dual axes relative to each other before the G92 block is executed.

Set Zero

A set zero operation may be performed on the axes in a dual group on an individual basis. For example, if you have a dual axis named X and it consists of two axes, X1 and X2, when the set zero operation is executed through PAL, you must specify which axis in the dual group to set zero. When the set zero operation is performed on an axis, the current axis location becomes the new zero point of the coordinate system.

Cutter Compensation

Only one tool diameter may be active at any one time. Any offset created by cutter compensation affects all axes in the dual group.

Tool Length Offsets

The system installer must select one of the dual axis group members as the tool length axis in AMP if separate tool length offsets are to be used for a dual group. Once one axis of the group has been assigned as the tool length axis, all axes in the group may have independent tool length values assigned to them.

Assuming tool length offsets are valid on your dual axis, their activation is the same as the tool length offset on non-dual axes. Refer to chapter 20 for details.

The offset function is selected with G43, G44, and G49. The offset value to activate is selected with an H word. This H word is the offset number and used for all axes in the dual group. The tool length offset values called by this H word, however, may be assigned individually. These values may be assigned either :

- manually through the tool wear and geometry tables as discussed in chapter 4.
- automatically through programming the correct G10 codes as discussed in section 11.2.1.

Assigning Tool Length Offsets Manually

For dual axes, extra tool length offset tables have been provided, one for each member of the dual axis group. By pressing the {NEXT SELECT} or {PREV SELECT} softkey, you can select which axis you are assigning length offset values in the dual axis group. Each member of the dual axis group is represented by the master axis name followed by a number indicating which axis in the group is active. Note that you can not activate the tool length offset using a softkey on a dual axis member.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE		MORE OFFSET	→
↑	MEAS- URE	INCH/ METRIC	PREV AXIS	NEXT AXIS	COPY OFFSET	→

If the same offset is desired for different dual axes, use the {COPY OFFSET} softkey to copy the offset values from one axis to another (refer to chapter 4 for details on using this feature).

Assigning Tool Length Offsets Through Programming G10

For dual axes, additional programming for the G10L10 and G10L11 codes is available when a dual axis is the tool length axis. In place of the normal G10 block which assigns tool length data using an R word, a dual axis used as a tool length axis must program length offset data using the following format:

```
G10L10P_A_B_C_;
G10L11P_A_B_C_;
```

where A, B, and C are the names the system installer assigned each axis in your dual group in AMP. A, B, and C are used to assign tool length data in place of the R word. Refer to the system installer's documentation for details.

END OF CHAPTER

Tool Control Functions

20.0 Chapter Overview

Tool control functions can be classified into 3 categories:

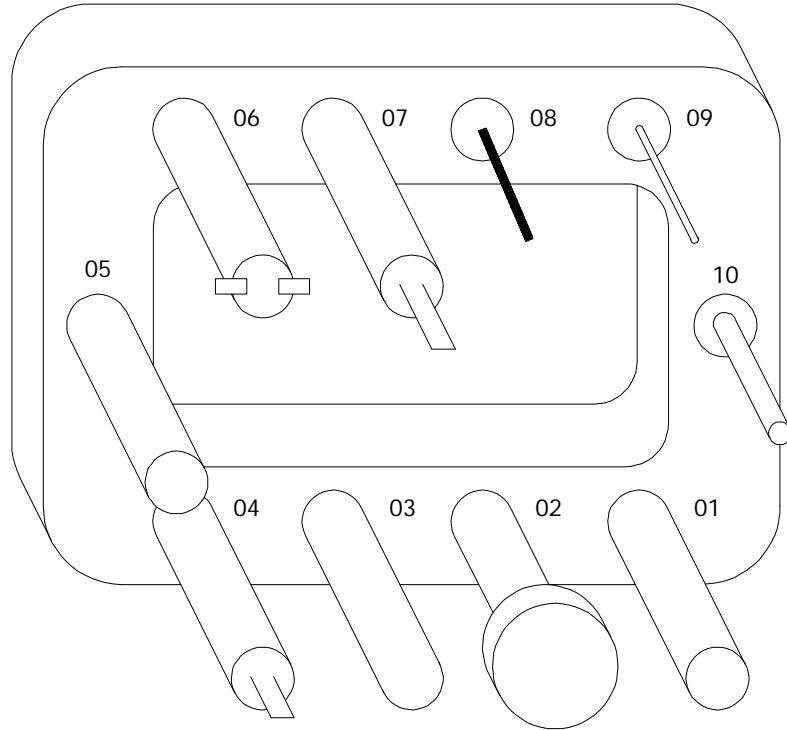
- Tool Selection- Programming a T-word and using random tool and tool life management to help select a tool
- Tool length offsets-compensate for the difference between the tool length assumed while programming, and the actual length of the tool used for cutting (see chapter 21 for details on tool diameter offsets using cutter compensation).
- Modify or enter data within the offset tables by programming the correct G10 code. For details see section 20.4.

Enter tool length offsets and cutter compensation data in to the control's offset table (see section 3.1). Once entered, call for a particular set of data by programming an H- (tool length data) or a D- (tool diameter data) word that corresponds to a offset number from the offset table.

20.1 Programming a T- word

A workpiece usually requires different kinds of cutting processes, and usually there are cutting tools that correspond to each process. The cutting tools are typically stored in a tool magazine and are assigned tool numbers (see Figure 20.1).

Figure 20.1
Typical Mill Tool Magazine



A T-address followed by a numeric value programs a tool selection (or tool group number - see section 20.5 on tool life management). The system installer determines in AMP how a tool change operation is programmed. There are four different options available. They are:

Return tool in M06 - When this method the T-word to activate is programmed in a block that does not contain an M06. The T-word is stored until some later block that contains an M06. When the M06 is executed the currently active tool is replaced with the last tool number programmed with a T-word. It is required with this method that the tool number (or group number) that is being replaced as the active tool is programmed in the block that contains the M06 command. If the M06 block does not contain the previously active tool number/group number, or if the wrong tool number/group number is programmed with the M06, the control will generate an error.

Next tool in T-word - This method is identical to the “Return tool in M06” method with the exception that the block containing an M06 can not contain a T-word. It is not necessary to program the previously active tool number/group number in the M06 block.

M06 Required - This method defines that a tool is only activated in an M06 block. A T-word that is programmed by itself becomes the next tool activated at an M06 block. Programming an M06 by itself activates the next tool. If a T-word is programmed in an M06 block that T-word is used as the active tool and any other unactivated T-word is discarded.

Activate Tool in T-word - For this method no M06 needs to be programmed to change tools. A tool change occurs immediately when the T-word is executed.

When the correct M06 block or T-word block that will execute a tool change is programmed the control outputs a tool selection signal to a tool changer. The tool changer should perform a sequence of operations to deliver the proper tool in response to the tool selection signal. For example, to select a cutting tool that is assigned tool number "03", write "T03" in the part program.

Since tool changers vary in style, size and function, the system installer is responsible for specific implementations through PAL. Refer to the PAL programmers manual and the manual supplied by the system installer for more details.

Important: When changing cutting tools it is usually necessary to change the tool offset at the same time. This is done with an H- or a D-word. For details see section 10.5.3.

Important: When the MISCELLANEOUS FUNCTION LOCK feature is activated, the control displays M-, B-, S-, and T-words in the part program with the exception of M00, M01, M02, M30, M98, and M99. This feature is activated through the Front Panel screen (as described in chapter 7) or through some optional switch installed by the system installer.

20.2 Tool Length Offset Function (G43, G44, G49)

To cut a workpiece using the bottom face of the cutting tool, it is more convenient to write the part program assuming that the gauge line of the tool holder equals the bottom face of the tool.

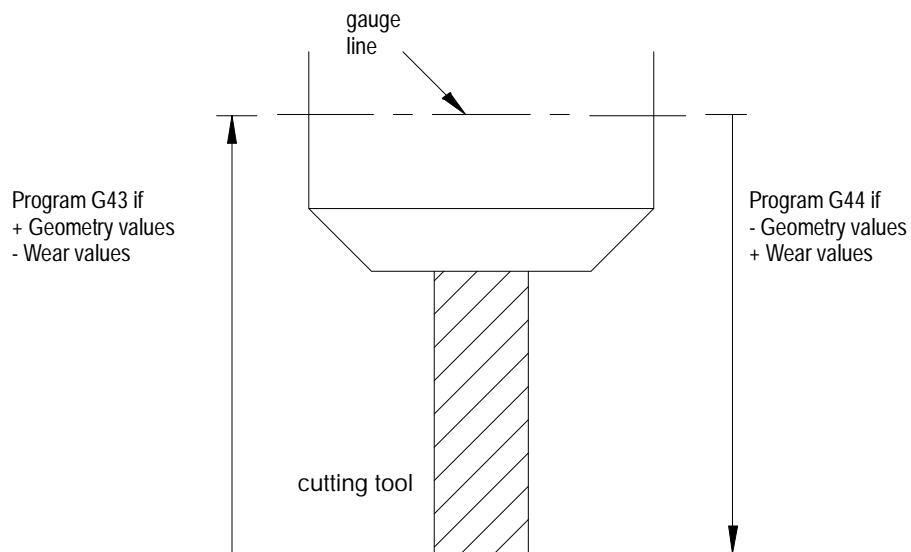
The term "gauge point" defines the precise point on the chuck or tool holder from which all programmed tool paths originate. Offsets refer to the distance from this gauge point to the edge of the tool that contacts the part being cut.

But when a cutting tool is set in the spindle, its bottom face is not at the gauge line. To cut the workpiece to the proper dimensions, offset the tool path by an amount that equals the difference between the gauge line and the bottom face of the cutting tool.

The control offers a function called tool length offset for offsetting tool paths. The tool length offset is usually equal to the difference between the bottom face of the tool and the gauge line. Put the tool length offset into memory in advance. This function lets the control use the same program to produce the same workpiece regardless of the length of the cutting tool.

Figure 20.2 illustrates the reference points used for deriving a tool length offset.

Figure 20.2
Tool Length Offset



There are three G codes, G43, G44 and G49, that are used when programming tool length offsets. To know when to use them, see below:

G43

If the sum of the tool geometry and the tool wear is a positive offset value, program G43.

For example:

If the values for tool offset no. 1 are:

Tool Geometry	+3.0000
Tool Wear	<u>-0.1000</u>
The tool offset is:	+2.9000

G44

If the sum of the tool geometry and the tool wear is a negative offset value, program G44.

For example:

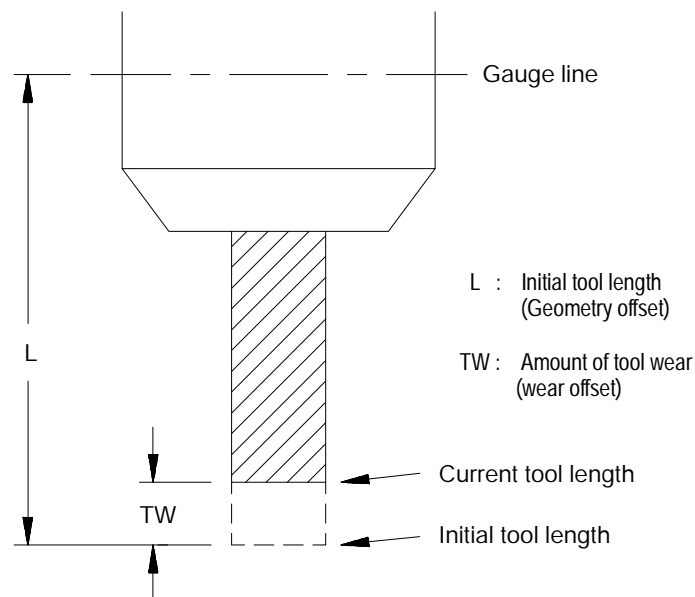
If the values for tool offset no. 1 are:

Tool Geometry	-3.0000
Tool Wear	+0.1000
The tool offset is:	-2.9000

G49

To cancel the tool length offset function, program G49.

Figure 20.3
Tool Offset Data



These G codes are modal, which means they are active from the program block that they have been entered to either the end of the program or until canceled. (They also belong to the same modal group.) However, the system installer must select one of these G codes in AMP. This G code would then be active during the entire program or until canceled. This manual assumes that G49 has been selected in AMP.

Use these formats for programming G43 or G44:

G43H__;

G44H__;

(“H” is the tool offset number.)

G43 or G44 does not have to be programmed with an H-word in the same block, or vice versa, in order for a tool offset to be made active. But the tool offset will only be activated at the time both a G-word and H-word are active.

Important: If using the tool life management feature, programming a H-word may not be necessary. (See section 20.5 for details on tool life management).

Depending on how the system installer has configured AMP, tool offsets may remain active after “end of program commands” (M02 or M30) are executed, a “control reset” is performed or E-STOP is reset.

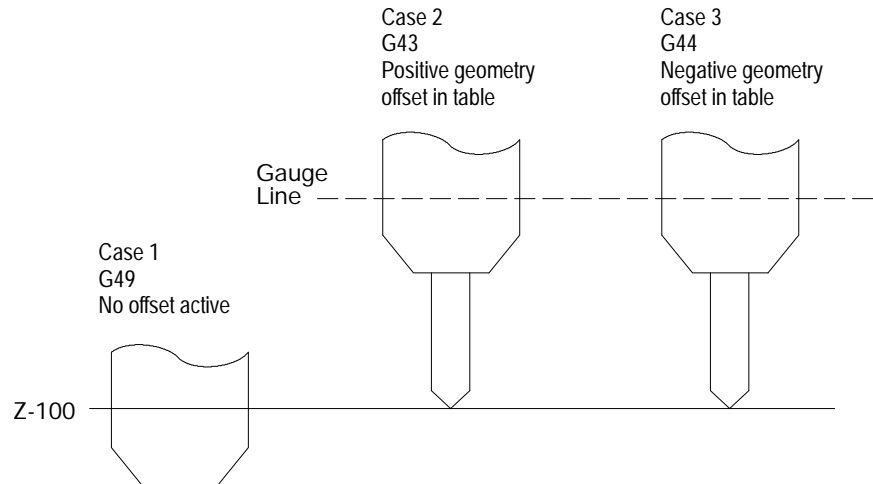
Example 20.1 and Figure 20.4 illustrate offset program blocks and how they affect tool position.

Example 20.1
Programming G43 or G44

Assume H01 offset data to be 15
Assume H02 offset data to be -15

G91G00Z-100. G43H00;	(Case 1)
G91G00Z-100. G43H01;	(Case 2)
G91G00Z-100. G44H02;	(Case 3)

Figure 20.4
Results of Example 20.1



Offset “H00” in the offset table is always equal to a value of zero, but does not cancel the tool offset mode like G49. H00 cancels H-words. Programming a G49 will not change the current H-word to H00. Example 20.2 illustrates this.

Example 20.2
Modal G43, G44 and Modal H-words

Program Block	Comment
N1G00G90;	
N2G43 H01;	G43 mode, H01 offset
N3 ;	G43 mode, H01 offset
N4H02;	G43 mode, H02 offset
N5;	G43 mode, H02 offset
N6G44	G44 mode, H02 offset
N7;	G44 mode, H02 offset
N8G49;	Offset mode cancelled, H02 offset
N9G43;	G43 mode, H02 offset
N10H0;	G43 mode, No offset value
N11G44;	G44 mode, No offset value

Important: Whenever a new tool length offset is initiated or cancelled, the block that changes the offset must be a linear block (G00 or G01). In the above example, blocks N2, N4, N6, and N8 must be linear blocks.

20.2.1 Activating Tool Length Offsets

The system installer has the option in AMP to determine exactly when the geometry and wear offsets will take effect and when the tool position will change to the new position. This manual makes the assumption that the system is configured to immediately shift the coordinate system by the geometry and wear amounts, and delay the move that will reposition the tool to the same location in the current work coordinate system. Refer to documentation prepared by the system installer to determine the application in a specific system.

Provided the system is configured as described above, the control activates a tool offset as described below.

1. The control reads a block that activates or deactivates a tool length offset. This may be a G43, G44, or G49 block or simply a block that contains an H-word (see Example 20.2).
2. The control immediately shifts the work coordinate system the amount of the tool geometry and tool wear amounts called by the H-word. The tool position display will change reflecting this shift. The absolute position display does not change. The offset is interpolated into the next move that generates axis motion on the offset axis, unless you are in incremental mode. If you are in incremental mode the offset is not interpolated into the next move of the axis. Example 20.3 shows how the move is generated in incremental and absolute modes when the tool offset programmed as 3.

Example 20.3
Immediate Shift/Delay Move in Incremental and Absolute Modes

Absolute Mode		Incremental Mode	
G00Z0	Rapid mode	G00Z0	Rapid mode
G90	Absolute Mode	G91	Incremental mode
T01	Activate tool 1. Program display changes Z position to -3.	T01	Activate tool 1. Program display changes Z position to -3.
Z1	Axis moves to +1	Z1	Axis moves to -2

If “immediate” is chosen for the move, the control generates a linear move that will reposition the cutting tool to its old coordinate position in the work coordinate system. This block is executed in the same block that calls for the offset. If axis words are present in the block that activates or deactivates a tool length offset, the control will add this generated move to the programmed move.

Important: Any block that activates or deactivates a tool length offset must be programmed in linear mode (G00 or G01) when executed. If a tool change is made in the circular mode, no axis motion may take place in the block changing the tool offset. The offset must be activated in a block with no axis words.

20.2.2 Tool Length Offset (TLO) Axis Selection (G43.1, G44.1)

When you program one of these TLO axis-select G-codes, the axis programmed in the block becomes the axis to which the tool length offset is applied; the forced axis name replaces the AMP-defined TLO axis. Otherwise, these G-codes have the same effect as the G43/G44 codes.

Important: The G43.1 and G44.1 blocks are motion blocks; therefore, the axis will move to the position (and offset) commanded in the block.

You must program a G49 before you can switch to a new TLO axis with G43.1 or G44.1. You must also remove the offset from any previously active TLO axis.

The axis that the control is currently using for tool length offset calculations is shown in reverse video on the tool offset screen. An axis shown in reverse video does not mean the offset is active, only that the axis is the current tool length axis. An asterisk indicates the currently active offset value.

The G10L10 and G10L11 offset table-modifier blocks now recognize axis names to let you modify TLO data for axes other than the currently active offset axis.

These conditions cause the AMP-defined TLO axis to become the active axis, replacing the axis that was selected with a G43.1 or G44.1:

- you perform a control reset
- the AMP parameter Cancel Tool Offsets on M02/M30 has been set to yes by the system installer, and an M02 or M30 block is executed

Switching planes with a G17/G18/G19 does not change the active TLO axis; the active TLO axis can be switched only by programming a G43.1/G44.1 with an axis name.

Copying Tool Length Offset Tables

Each selectable tool length axis has its own tool offset table and its own set of offset values. For example, if you select the X-axis as the tool length offset axis, you must enter an offset value for the X-axis for those tool offsets to be used on X.

To copy the offset values from one axis to another, follow these steps:

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {TOOL WEAR} or {TOOL GEOMET} softkey, choosing the table from which you want to copy.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

3. Choose the axis whose values you want to copy.

If you want to move:	Press this softkey:
forward	{NEXT AXIS}
backward	{PREV AXIS}

4. Press the {COPY OFFSET} softkey.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
↑	MEAS- URE	INCH/ METRIC	PREV AXIS	NEXT AXIS	COPY OFFSET	→

“COPY (SOURCE, DESTINATION): ” appears.

5. Enter the axis letter from which the data is coming, then a comma, and then the axis letter to which the data is going. For example,

COPY (SOURCE, DESTINATION): X, Z

copies the offset data from the X axis to the Z axis for all offset numbers.

20.3 Random Tool

The random tool feature is typically used to speed up production by saving cycle time when a tool is returned to the tool changing device. This is done by allowing the tool changer to randomly return the cutting tool to the most convenient pocket in the tool changing device. The control will remember what pocket the tool is returned to and be able to call the same tool from the new pocket at any time.

Important: This feature may be used with normal tool selection, or the tool life management feature if desired.

This feature has no effect on tool length offsets or cutter compensation. These features must still be activated correctly as described in their individual sections.

The random tool feature will automatically decide the pocket (or pockets) that contains the requested tool based on the information in the pocket assignment table. If the requested tool has not been assigned to a pocket the control will generate an error.

Based on the current pocket number, which is maintained by PAL, the control will tell PAL which pocket to move too and how far and in which direction to move. The control will also tell PAL where the tool currently in use can fit in the tool turret.

Important: This feature is very PAL dependant. Before using this feature make sure the system installer has written the PAL program to allow the use of Random Tool.

The control will automatically update the tool pocket assignment table when tool changes are made. The control will indicate to PAL the best location to return the tool to. PAL then decides based on this information where the tool gets placed in the tool holder. The pocket that is vacated by the new tool is marked as empty.

Manually Entering Random Tool Data

Data may be entered into the random tool table either manually, as described here, by programming, or by running a backup program of the tool data. These other methods are described later in this section.

To manually enter the random tool data, follow the steps described below:

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWIT CH LANG		→

2. Press the {RANDOM TOOL} softkey. The pocket assignment table screen is displayed as shown below. This screen shows the current tool to pocket assignments. The system installer assigns the number of tool pockets available on a system in AMP.

(softkey level 2)

↑	WORK CO- ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

Figure 20.5
Typical Random Tool Pocket Assignment Screen

POCKET ASSIGNMENT TABLE				PAGE 1 OF 2	
PKT	TOOL	PKT	TOOL	PKT	TOOL
001	0002	002		003	0001
004		005	0003	006	XXXX
007	0007	008	XXXX	009	
010		011		012	
013		014		015	
016		017		018	XXXX
019	0006	020	XXXX	021	
022		023		024	
025		026		027	
028		029		030	
031		032		033	
034		035		036	
037		038		039	

↑	REPLCE VALUE	CLEAR VALUE	CUSTOM	ACTI VE	BACKUP
---	-----------------	----------------	--------	---------	--------

The columns labeled PKT give the pocket numbers of the tool changer. The columns labeled TOOL give the tool number of the tool in the corresponding pocket. Pockets with no tools assigned to them show no information next to the pocket number. Pockets with tools shown as XXXX indicate that a custom tool (tool that requires more than one pocket) has been assigned to use that pocket.

3. Move the cursor to the pocket number for which an assignment or change is to be made. This is done by pressing the up, down, right, or left cursor keys on the operator panel. Move the cursor full pages by holding down the [SHIFT] key while pressing the up or down cursor keys. The selected pocket is shown in reverse video.

Important: If random tool is not to be used for a specific system, make sure that none of the tool pockets have tool numbers assigned to them.

4. To modify tool data there are three choices:
 - To remove a tool assigned to a pocket press the {CLEAR VALUE} softkey. The selected tool is deleted from the table.
 - To enter a tool number for the pocket, press the {REPLCE VALUE} softkey, key in the new tool number and press the [TRANSMIT] key. The old tool value will be replaced with the new value just keyed in.
 - To enter a custom tool (a tool that requires more then one tool pocket) enter the tool number of the custom tool in the pocket that is to be used as the “shaft pocket”. The shaft pocket is where the tool changer will be positioned when the particular custom tool is to be used. Enter the number of pockets needed (to a max of 9), a comma, followed by the position of the shaft pocket in this group of pockets. Pressing the [TRANSMIT] key enters the data into the table.

The screen will show XXXX for the tool number of any pockets that have been configured as part of a custom tool, and show the tool number in the pocket where PAL will be told to go in order to find the tool.

For example, in the screen shown in Figure 20.5, pocket number 19 is a shaft pocket for custom tool number 6. This custom tool requires 3 pockets, pockets 18, 19, and 20. When the {CUSTOM} softkey was pressed for pocket number 19, a value of 3,2 was entered.

Programming the Random Tool Data

This feature is available so that it is not necessary to always manually enter the data into the pocket assignment table. By programming the correct G10.1 blocks all information is entered into the tool pocket table. Note the control can automatically generate a G10.1 program by using the backup softkey as described later in this section.

Important: Note that G10.1 blocks may not be programmed when cutter compensation is active.

Programming of random tool data can only be done on a tool pocket if data has not already been configured for that pocket. If you need to make changes to a tool pocket that already has a tool assigned to it, you must either clear and re-load the entire random tool table as discussed below (you can not use a G10.1 to clear individual pocket data), or use the softkeys to manually access the random tool table and change the data using the keyboard.

The following block is used to set data for the random tool pocket assignment table:

G10.1 L20 P__ Q__ O__ R__;

Where :	Is :
G10.1 L20	This tells the control that the block will be setting data for the random tool pocket table. The G10.1 L20 is not modal, it must be programmed in every block that sets data for the random tool pocket assignment table.
P	The value following the P-word determines the pocket number that is being set.
Q	The value following the Q-word determines the tool number of the tool that is in the pocket determined with the P-word.
O	The value following the O-word enters the number of pockets that are needed for the tool. Normally a value of one is entered here however, for custom tools that require more than one pocket, program the number of pockets that are required.
R	The value following the R-word enters the pocket number of the shaft pocket for the tool. Normally a value of one is entered here however, for custom tools that require more than one pocket, program the location relative to the other pockets for that tool that the tool changer will go to to access that tool.

For example the following block:

G10.1L20P1Q2001R1;

tells the control that tool number 20 is in pocket number 1.

G10.1L20P3Q2304R2;

tells the control that tool number 23 has its shaft pocket as pocket number 3, four pockets are required for the custom tool and the second of these four pockets is the shaft pocket. This means that pockets 2, 3, 4, and 5 are used for the custom tool number 23.

Clearing Random Tool Data

The following block will clear all information in the random tool table:

G10.1 L20 P0 Q0 O0 R0;

Backup Random Tool Table

The control has a feature that will allow the information in the random tool table to be backed up (saved in the form of a program). This is done by the control generating a G10.1 program from the information already in the table. To do this follow these steps:

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
--	------------------	--------	----------------	-----------------	------------------	---

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→
--	----------------	-----------------	---------------	-----------------	--	---

2. Press the {RANDOM TOOL} softkey.

(softkey level 2)

↑	WORK CO- ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
---	-----------------	--------------	----------------	----------------	----------------	---

↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→
---	-----------------	------------------	--------	--	--	---

3. Press the {BACKUP} softkey. The control will prompt the operator for a program name. Key in the program name that is to contain the information from the random tool table and press the [TRANSMIT] key. This program name can not already exist in control memory.

This will allow the control to generate a program that will automatically load the necessary data into the random tool table. This program may be edited by the operator if desired, as changes to tool table are needed.

The control will automatically place this G10.1 program in control memory. If it is desired to send the program to a peripheral device it must be sent from control memory as discussed in chapter 9. The program can not be sent directly to a peripheral.

Starting a program with a tool already active

If desired, a part program may begin execution with a tool already active in the chuck. In order for random tool to be able to properly handle that tool, it is necessary to enter information about that tool in the random tool table.

Important: If random tool was used when the tool was loaded into the chuck, it is not necessary to enter any data since random tool will remember what tool is loaded even after power is turned off. This procedure is only necessary if a tool is loaded manually or if random tool was not used when the tool was loaded.

The control will need the following information to properly handle a tool that is already active in the chuck. Tool number, number of pockets the tool uses, and position of the shaft pocket relative to these other pockets (refer to the section on manual entry of data for details on shaft pocket and custom tool data). Do this in the following way:

1. Press the {ACTIVE} softkey. The control will prompt the operator for the tool number, the number of pockets, and the position of the shaft pocket relative to the these other pockets all separated by commas.

(softkey level 3)

↑	REPLCE VALUE	CLEAR VALUE	CUSTOM	ACTIVE	BACKUP
---	-----------------	----------------	--------	--------	--------

2. The control will display the configuration of the tool that it thinks is currently loaded into the chuck. If these values are incorrect, replace them using the correct tool information. Enter a value for tool number, number of pockets, and position of the shaft pocket all separated by commas on the input line. Data can be edited on the input line as discussed in section 2.10.
3. When the data for the tool that is currently in the chuck is correct, press the [TRANSMIT] key. The control will now be able to return the tool that is in the chuck to the best location in the tool changer at the proper time.

Important: You can also use the {ACTIVE} softkey to clear the currently active tool and specify no tool is currently in the spindle. To specify no tool is in the spindle press the {ACTIVE} softkey and delete any information that appears at the prompt. When the [TRANSMIT] key is pressed, the active tool is cleared and the random tool assumes no tool is in the spindle.

20.4
Programming Alterations of
the Offset Tables (G10L10 -
G10L13)

It is possible to alter or generate values in the tool offset tables (see section 3.1) by using the programming feature discussed in the following section.

It is possible to enter data in the offset tables by programming the correct G10 command. The following section describes the use of the G10 commands.

Important: Note that G10 blocks may not be programmed when cutter compensation is active.

A G10 code used to modify a tool offset table value will only change the value in the table. If that offset value is currently being used by the control, the active offset value will not be updated until the offset value is called again from the table using a D or an H-word.

Any values entered in an offset table using the G10 command while the control is in incremental mode (G91), are added to the currently existing offset data. Any values entered in an offset table using the G10 command while the control is in absolute mode (G90), are used to replace the currently existing offset data.

The following is a representation of the basic format for modifying the offset tables.

G10 L(10-13) P__ R__;

Where :	Is :
L(10-13)	Designates the definitions of the other parameter data being used in the G10 block. The L-word defines the other parameters as described in Table 21.A.
P	see Table 21.A
R*	see Table 21.A

* When setting tool length axis values, R can be replaced with the name of the axis (typically X, Y, or Z). Axis names are always used by the control when offset tables are backed up by the control. If you use an R to set a tool length value, it is applied to the axis currently selected as the tool length axis. See page 20-9 for details on selecting the active tool length axis.

Table 21.A
Parameters for Modifying the Tool Offset Tables

Value for the L Parameter	Parameter Definition P R, X, Y, Z	
	P	R, X, Y, Z
L10 Geometry table	Offset Number	tool length geometry value
L11 Wear table	Offset Number	tool length Wear table

Value for the L Parameter	Parameter Definition P R, X, Y, Z	
L12 Geometry table	Offset Number	Tool radius geometry value
L13 Wear table	Offset Number	Tool radius wear value

Example 20.4
Replacing the Tool Offset Tables Through Programming (G90)

Assume a Z axis geometry value (tool length) of 2 for offset number 4.

```
N00001 G90;
N00002 G10 L10 P4 Z3;    Offset number 4 has a new value of 3 for
                          tool length.
N00003 G10 L10 P4 Z1;    Offset number 4 has a new value of 1 for
                          tool length.
```

Example 20.5
Modifying the Tool Offset Tables Through Programming (G91)

Assume a Z axis geometry value (tool length) of 2 for offset number 5.

```
N00001 G91;
N00002 G10 L11 P5 Z1;    Offset number 5 has a new value of 3 for
                          tool length wear.
N00003 G10 L11 P5 Z3;    Offset number 5 has a new value of 6 for
                          tool length wear.
```

20.5 Automatic Tool Life Management

The automatic tool management feature is used to monitor the life of a tool, determine when the tool should be replaced, and provide a replacement tool when that tool is requested in a program.

This is done by assigning tools to selected groups. Then instead of calling a specific tool in a program the programmer calls a tool group. The control then selects the first tool assigned to that group. If that tool has exceeded its entered tool life then a replacement tool is selected from the next tool number assigned to that group. If that tool has exceeded its expected tool life then the next tool in the group is selected. This continues until no more tools are available in that tool group.

The correct tool length and tool radius offsets are assigned independently for each tool in the group.

20.5.1 Tool Directory Data

This section discusses how to set up the tool groups and the information that must be entered for each tool group. Note that this section discusses the manual method of entering this information. Section 20.5.3 discusses a method of entering all information into the tables by programming.

Assigning Tool Numbers to Groups

Normally tools that are assigned to the same group have similar characteristics (such as a boring tool or a drilling tool). If one tool in the group is worn the control should be allowed to select any tool in the same group and still be able to cut the same part using the same program.

The system installer determines in AMP the usable range of tool group numbers by determining a boundary (maximum number of groups allowed is 200). Any tool number that is programmed above this boundary is used as a tool group number (the value of the boundary is subtracted from the tool number programmed - see section 20.5.4 for details on calling a tool group during tool management). Any tool number that is programmed below this boundary is used as a normal tool number.

Different tool length offset numbers, and radius offset numbers are entered into the tool management table with the tool numbers in each group. When a tool is selected from a group by the control the tool length and radius offset numbers are activated with them. See section 3.1 on entering tool data for details.

Tool Life Measurement Type

The control is capable of measuring the life of a tool using one of three possible methods:

0. Time - This is selected by choosing 0 as the type of tool life measurement. Time measures tool life as the length of time that a cutting tool is operated at a cutting feedrate. The value for the expected tool life is entered in units of minutes.
1. Number of times used- This is selected by choosing 1 as the type of tool life measurement. Number of times used measures tool life as the number of times that the tool is selected as the active tool. The value for the expected tool life is entered as the number of times the tool may be used to cut parts. This number is the number of part programs. Regardless of the number of times that a tool is selected as active in a specific program it will only count as one use each time the program is executed.

- Distance - This is selected by choosing 2 as the type of tool life measurement. Distance measures tool life as the distance that the tool has been moved using a cutting feedrate. The value for the expected tool life is entered in units of inches or millimeters depending on the mode that the control is operating in at the time. For multi-axis moves, the vectorial distance traveled by the tool is the distance used for tool life measurement.

The tool life type (selected as either 0, 1, or 2) is selected on a per-group basis. Different groups may use different tool life types however each tool in the group uses the same tool life type.

Tool life Threshold Percentage

A threshold level may also be assigned to a tool group. The threshold level is assigned as a percentage of the total expected life of the tool. When a tool reaches this threshold level it is classified as old for that tool group. A tool is classified as old only to allow the operator to see that a tool is close to expiration. If the tool is being used when it reaches the threshold level it will continue to be used as normal until the tool reaches the “expired state” (100% of the expected tool life). If the tool life has reached the “expired state” when cutting a part, the control will continue to use the tool until the operator executes a tool change operation.

The tool life threshold percentage is selected on a per-group basis. Different groups may use different threshold percentages however each tool in the group uses the same threshold percentage.

Entering Tool Group Data

To enter tool group data it is first necessary to create the tool groups. This is done automatically when the group is selected to edit. To enter tools into groups and enter other tool group data follow these steps:

- Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWITC H LANG		→

2. Press the {TOOL MANAGE} softkey.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
---	----------------	--------------	----------------	----------------	----------------	---

↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→
---	-----------------	------------------	--------	--	--	---

3. Press the {TOOL DIR} softkey. The control will display the current tool directory screen showing all of the current tools and the groups that they have been assigned to (see the following figure). The control will display the prompt "EDIT GROUP:".

(softkey level 3)

↑	TOOL DIR	TOOL DATA	BACKUP DATA			→
---	-------------	--------------	----------------	--	--	---

Figure 20.6
Typical Tool Group Directory Screen

TOOL GROUP DIRECTORY				PAGE 1 OF 1	
(FILE NAME)					
<u>GROUP</u>	<u>TOOL NUMBER</u>				
1	1	2	44	88	
2	55	63	90	99	
↑	EDIT GROUP	DELETE GROUP	DELETE ALL		→

At this point if it is desired to delete any or all tool groups that already exist for some reason follow these steps:

- To delete a select tool group press the {DELETE GROUP} softkey. Key in the desired group number to delete and press the [TRANSMIT] key. This will delete all information in the tool group including the tool offset numbers, threshold rate, tool numbers, etc....
 - To delete all of the tool groups press the {DELETE ALL} softkey. The control will prompt “DELETE ALL TOOL MANAGEMENT DATA? (Y/N):”. Entering “Y” will delete all tool management data that has been entered into the management tables (this will not delete any G10 programs that are backups or used to set the tool management tables). Entering “N” will abort the delete operation.
4. Key in the group number to edit and press the {EDIT GROUP} key. The following screen showing all the information for that tool group will be displayed.

Figure 20.7
Typical Tool Group Data Screen

ENTER VALUE=

EDIT TOOL GROUP (ACTIVE PROGRAM NAME)	PAGE 1 OF 1 THRESHOLD RATE=80%
--	-----------------------------------

ENTRY	TOOL NUMBER	LIFE TYPE = TIME
1	1	
2	2	
3	4	
4	0	
5	0	

↑

CHANGE TOOL	INSERT TOOL	DELETE ALL	CHANGE TYPE	CHANGE T RATE
----------------	----------------	---------------	----------------	------------------

→

5. From this screen it is possible to perform the following operations. The application of these operations was discussed in detail earlier in this section.
 - **Change Tools** - Alter one of the tool numbers that has already been entered in the group. Move the cursor to the tool number to be changed by pressing the up or down cursor keys (move the cursor full pages by holding down the shift key while pressing a cursor key). Press the {CHANGE TOOL} softkey. Key in the new tool number and press the [TRANSMIT] key.
 - **Insert Tool** - Insert a new tool number for that group. Move the cursor to the location to insert a new tool number at by pressing the up or down cursor keys (move the cursor full pages by holding down the shift key while pressing a cursor key). Press the {INSERT TOOL} softkey. Key in the new tool number and press the [TRANSMIT] key. The actual range of allowable tool numbers is 1 to 9999.
 - **Delete Tool** - Remove a tool number from that group. Move the cursor to the tool number to be removed by pressing the up or down cursor keys (move the cursor full pages by holding down the shift key while pressing a cursor key). Press the {DELETE TOOL} softkey.
 - **Change life type** - Alter the method that the control uses to record and measure tool life for that group. Press the {CHANGE TYPE} softkey. The prompt "LIFE TYPE..." is displayed on line 2 of the CRT. Note that the same life type is assigned to all tools in any one tool group. Key in the number of the desired tool life measurement type for that group and press the [TRANSMIT] key. 0 for "time", 1 for "number of tool uses", and 2 for "distance".

Important: When a change is made to the tool life type, any accumulated life and the expected life for all of the tools is reset to zero.

- **Change life threshold rate** - Alter the percentage of a tools expected life that a tool will be labeled as old. This percentage applies to all tools in the selected group. To alter the threshold percentage (the percentage of expected tool life that has been used before a tool will be classified as old) press the {CHANGE T RATE} softkey. Key in the percentage of the expected life and press the [TRANSMIT] key. Tool life threshold rate is explained earlier in this section.

All of this information may be altered for the tool groups using the programming method discussed in section 20.5.3.

20.5.2 Assigning Detailed Tool Data

This section assumes that tools have already been assigned to their specific groups as discussed in section 20.5.1. This section discusses specific information that is to be entered into the tool life management tables for the individual tools. This information may also be entered into the tool management tables using the programming method discussed in section 20.5.3. This information includes:

- Tool length offset number
- Tool diameter/radius offset number
- Expected life of a tool
- Renew a tool life

Tool length and diameter/radius offset number

This feature of tool life management is used so the programmer does not need to know what tool has been called by tool life management and still have the correct tool offsets and cutter compensation activated. Using this method the programmer no longer needs to program an H or a D-word to activate tool offsets or cutter compensation numbers, tool life management will activate the offset numbers automatically.

Important: The control will only automatically enter the tool length and cutter compensation offset numbers. This may or may not activate the tool length offset (G43, G44) or cutter compensation features (G41, G42). These features must still be activated as normal with the exception of the fact that it is not necessary to program the H or D-word since it is automatically entered by tool management.

Expected Tool Life

This feature of tool life management is used to set the expected life of a tool. The type of tool measurement used is assigned to the tool group as discussed in section 20.5.1. This tool measurement type determines the units that are used for the expected tool life.

As a tool is used the amount of usage is recorded and displayed as the accumulated tool life (the amount of the expected tool life that has been used). This is displayed individually for each tool on the tool data display screen. The accumulated tool life may be reset to zero by pressing the {RENEW TOOL} softkey.

The following is a discussion of the units that should be entered for the different tool life measurement types:

0. **Time** - If tool life is measured in units of time (0 is selected as tool life type), then the units for the expected tool life is minutes. Enter the minutes of operation that the tool is expected to operate and still be within the tolerance required for the part being cut. The accumulated life of a tool is only measured when that tool is the active tool, and it is performing a cutting operation. Moves that are rapid, or blocks that do not produce axis motion are not added to the accumulated tool life.
1. **Number of times used** - If tool life is measured by the number of uses (1 is selected as tool life type), then the units for the expected tool is the number of programs that the tool may be selected as an active tool in. The accumulated life of a tool is increased by one if that tool is selected in a program as the active tool. Remember that the same tool may be activated more than once in a program, however its accumulated life will only increment by one. Enter the total number of program executions that may use the tool before the tool will no longer meet the required tolerance for the part being cut.
2. **Distance** - If tool life is measured in units of distance (2 is selected as tool life type), then the units for the expected tool life is either inches or millimeters (depending on the current operating mode of the tool). Enter the distance of travel that the tool is expected to cut and still be within the tolerance required for the part being cut. The accumulated life of a tool is only measured when that tool is the active tool, and it is performing a cutting operation. Moves that are rapid, or blocks that do not produce axis motion are not added to the accumulated tool life. For multi-axis moves (including arcs and helices) the distance added to the accumulated life is the vectorial distance, not necessarily the distance traveled on each axis.

Entering Specific Tool Data

The following steps describe in detail the method of entering specific tool data for tool management. This includes tool offset numbers, and expected tool life:

Important: This section assumes that the steps required to assign tools to specific groups has been performed as described in section 20.5.2.

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {TOOL MANAGE} softkey.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

3. Press the {TOOL DATA} softkey. The control will display the prompt "EDIT GROUP:".

(softkey level 3)

↑	TOOL DIR	TOOL DATA	BACK DATA			
---	-------------	--------------	--------------	--	--	--

4. Key in the group number to edit using the keys on the operator panel and press the [TRANSMIT] key. The following screen showing all the information for that tool group will be displayed.

Figure 20.8
Typical Tool Data Screen

GROUP 1 DATA TYPE=TIME PAGE 1 OF 1 (FILE NAME)					
THRESHOLD RATE = 80%					
TOOL NO	T. LEN OFF NO	CUTTER CMP NO	EXPECT LI FE	ACCUM LI FE	TOOL STATUS
1	2	3	100	100	EXPI RED
12	20	57	100	95	OLD
23	40	95	100	0	

↑	EDT LN OFF #	EDT CT OFF #	EDIT LI FE	RENEW TOOL	SCROL COLOR	→
---	-----------------	-----------------	---------------	---------------	----------------	---

5. From this screen it is possible to perform the following operations. The application of these operations was discussed in detail earlier in this section.
 - Enter or alter the tool length offset number. - To enter or alter a value for the tool length offset number, move the cursor to the tool number of the tool to alter and press the {EDIT LN OFF #} softkey. Key in the new offset number that will call the correct data from the offset tables for that tool and press the [TRANSMIT] key. The old value for tool length offset number (if any) is discarded and the new value replaces it.
 - Enter or alter the tool radius or diameter offset number - To enter or alter a value for the tool radius or diameter offset number, move the cursor to the tool number of the tool to alter and press the {EDIT CT CMP #} softkey. Key in the new offset number that will call the correct data from the offset tables for that tool and press the [TRANSMIT] key. The old value for radius or diameter offset numbers (if any) is discarded and the new value replaces it.

- Enter or alter the expected life of a tool - To enter or alter a value for the expected life of a tool, move the cursor to the tool number of the tool to alter and press the {EDIT LIFE} softkey. Key in the new expected life of the tool (in units as determined by the tool life type) and press the [TRANSMIT] key. The old value for expected life (if any) is discarded and the new value replaces it.
- Reset the accumulated tool life to zero - To reset the accumulated tool life to zero, move the cursor to the tool number of the tool to alter and press the {RENEW TOOL} softkey. The old accumulated life of the tool is discarded and a value of zero is entered as the new accumulated tool life. This is normally performed after an old or expired tool has been replaced with a new tool. This will update the status of the tool and remove any "OLD", or "EXPIRED" status.
- Change color used to graph the tool selected on the CRT for color systems. Each tool group may be assigned its own tool color. To change the color selected for a tool on the CRT press the {SCROL COLOR} softkey. This will alter tool color for the current tool when tool paths are displayed in either check run graphics, or active tool path graphics.

(softkey level 4)

↑	EDT LN OF #	EDT CT COMP #	EDIT LI FE	RENEW TOOL	SCROL COLOR	→
---	----------------	------------------	---------------	---------------	----------------	---

20.5.3 Programming Data and Backing Up Tool Management Tables (G10L3, G11)

This feature allows the rapid loading of information into the tool management tables. This is done by allowing a program to be executed that will automatically load the tool management tables. This program may also be generated automatically when the tool management tables are backed up as discussed later in this section.

Data is sent to the tool management tables when the control executes the following G10 block:

G10L3;

This block indicates to the control that any information following this block is to be used to set the tool management tables.



CAUTION: Any time that a G10L3 block is executed the control will automatically clear all information that is in the management tables for all tools and tool groups.

Any time after the G10L3 command, parameters may be programmed to enter what tool group is being entered, the type of tool life measurement that is being used, and the tool life threshold percentage. Details on these features are discussed in section 20.5.1. The format for this block is:

P_I_Q_;

Where :	Is :
P	The value entered with the P-word is used to program what tool group number is being edited. The following blocks will assign tools to that tool group.
I	The value entered with the I-word is used to program the type of tool life measurement that is to be used for all the tools in that group. I0 sets a type of time, I1 sets a type of number of uses, and I2 sets a type of distance (see section 20.5.1 for details). If more then one I-word is programmed for a tool group the control uses the last programmed I-word for that group. If no I-word is programmed for a group the control will use I0 (time) as a default value.
Q	The value entered with the Q-word is used to program the threshold percentage for that tool group. Enter the percentage of the total expected tool life that will cause the tools in the group to be classified as old. See section 20.5.1 for details on threshold percentage. If the Q-word is not programmed in a block the control uses a default value of 80%.

The following program blocks assign tools to groups, length and cutter compensation offset numbers, and expected tool life to specific tools. This information is assigned to the last group number programmed in a block using the P-word. The format for these blocks are:

T_H_D_L_;

Where :	Is :
T	The value entered with the T-word is the tool number of the tool to be assigned to that group
H	The value entered with the H-word is the tool length offset number from the tool geometry and wear tables that is to be assigned to this tool. The H-word is only valid if programmed in the same block as a T-word.
D	The value entered with the D-word is the tool radius offset number from the tool geometry and wear tables that is to be assigned to this tool. The D-word is only valid if programmed in the same block as a T-word.
L	The value entered with the L-word is used to program the value of the expected tool life for that tool and the following tools in the same tool group. The controls interpretation of this is dependant on the value set with the I-word in this program block. The value programmed with the L-word remains active for all following tools in that group until replaced with a different L-word, or a new tool group is programmed with a P-word.

All of the tools should then be programmed for that group in individual blocks. When all of the tools for that group have been entered change groups by programming a different P-word in a block.

When all of the tools for all of the different groups have been entered, end the execution of editing the tool life management table by programming either a M02 or M30 end of program blocks or by entering the following block:

G11;

This cancels the G10 data setting mode for tool management.

Important: Any information that was previously entered for any of the tool groups is lost when the control executes the G10L3 block.

Example 20.6
Programming Tool Life Management Data

Program Block	Description
G10L3;	Starts loading tables.
P1I1Q60;	Begins loading data for tool group 1. Type 1 (number of uses) measurement. Threshold 60%.
T1H5D7L25;	Places tool 1 in group 1 with length offset number of 5, cutter radius offset number 7, and expected life of 25 uses.
T2H2;	Places tool 2 in group 1 with length offset number of 2, no cutter radius offset number and expected life of 25 uses.
T15H7;	Places tool 15 in group 1 with length offset number of 7, no cutter radius offset number and expected life of 25 uses.
P2;	Begins loading data for tool group 2. Type 0 measurement (default). Threshold at 80% (default).
T12H3D6L40;	Places tool 12 in group 2 with length offset number of 3, cutter radius offset number of 6, and expected life of 40 minutes.
T13;	Places tool 13 in group 2 with length and radius offset numbers of 0 and expected life of 40 minutes.
P4I0Q90;	Begins loading data for tool group 4. Type 0 (time) measurement. Threshold at 90%.
T20H3D6;	Places tool 20 in group 4 with length offset number of 3, cutter radius offset number of 6, and expected life of 0 minutes.
Q50;	Resets the threshold at 50% for group 4.
G11;	Ends the loading operation.
M02;	

Backing up tool management tables

This feature causes the control to automatically generate a G10L3 program that will store all of the information that it finds in the current tool management table. Any time that this G10 program is executed it will clear any information that is currently in the management tables and replace it with the information that is in the G10 program.

To generate the G10L3 backup program of the tool management tables follow these steps:

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {TOOL MANAGE} softkey.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

3. Press the {BACKUP DATA} softkey. The prompt "BACKUP FILENAME:" will be displayed on the input line.

(softkey level 3)

↑	TOOL DIR	TOOL DATA	BACK DATA			
---	-------------	--------------	--------------	--	--	--

4. Key in any legal program name and press the [TRANSMIT] key. This program name will be used as the program that will store all of the backed up tool management data. The control will generate the tool management G10 program.

20.5.4 Programming Using Tool Management

The following section discusses how to activate a tool using tool life management. Here are some considerations to keep in mind when using tool life management.

- The system installer sets up a boundary for T-words used with tool life management in AMP. Any T-word that is programmed less than or equal to this number will be used as a normal tool number. Any T-word that is programmed greater than this number is used as a tool group number for tool management.
- When a T-word is programmed using tool life management the group that is called with the T-word is equal to the programmed T-word, minus the value of the boundary set in AMP by the system installer. For example if the system installer sets the boundary as 100, programming T101 would call a tool from group 1. T102 would call a tool from group 2, programming a T-word of 100 or less would call the specific tool number that is programmed. This manual assumes that a boundary of 100 is set in AMP. Refer to the system installers documentation for details on the boundary value for a specific system.
- The system installer determines in AMP when the control activates a tool after a T-word is programmed. This manual assumes that an M06 block is required to activate the tool change process, and any T-word that is programmed in the M06 block must be the tool number or group number of the tool that is being replaced. This may not be the case in a specific system (see section 20.1 for the four different types). Refer to the system installers documentation for details on activating a T-word.
- Any time after a tool from a tool group has been activated and a D- or an H-word is programmed, the newly programmed D or H value will take priority. The values in the tool management table for the tool length offset number and tool radius offset number are ignored until a different tool is selected or that same tool is reselected.

Example 20.7
Programming Tool Changes Using Tool Life Management

The following example assumes that the system installer has configured in AMP, both, the boundary for tool life management at 100, and an M06 to perform a tool change. It also is assumed that the tool changer is located at the secondary machine home point called by a G30, this is not necessarily true for different machine applications.

Program Block	Description
G49G30X10Z10F100;	Return to secondary home position.
T101;	Next tool change will be a tool from group 1.
M06;	Change to a group 1 tool.
G43;	Activate tool length offset using the offset number for the tool as assigned in the tool management table.
G29;	Return from secondary home position
G42;	Activate cutter compensation right using the offset number for the tool as assigned in the tool management table.
T102;	Next tool change will be a tool from group 2.
G01X13Y1F200;	Cutting with a group 1 tool.
G30;	Return to secondary machine home.
M06T101;	Replaces the group 1 tool with a group 2 tool. Note the T-word is optional in this block.
G29;	Return from secondary home position. New tool length offset values and new tool radius offset values take effect.
G01X2Y2F100;	Cutting with a group 2 tool.
G41D2;	Changes the current tool radius number that was activated with this tool and replaces it with the new D2 offset values. Note that the tool management table does not get changed. Also changes to cutter compensation left.
M30;	

END OF CHAPTER

Cutter Diameter Compensation (G40, G41, G42)

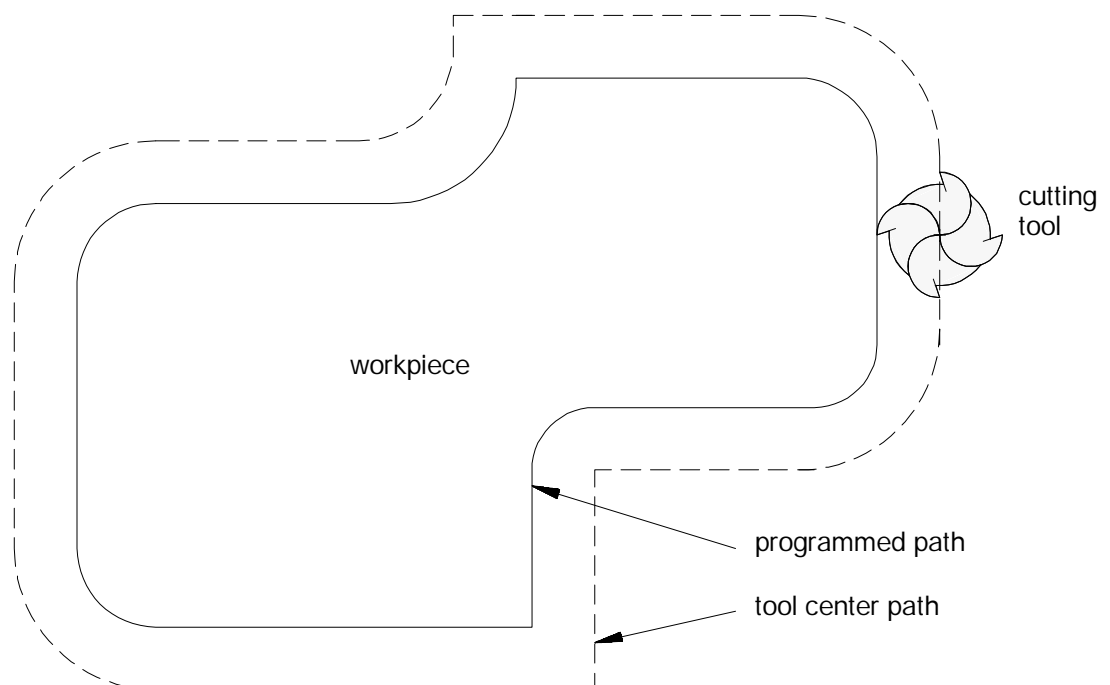
21.0 Chapter Overview

To cut a workpiece using the side face of the cutting tool, it is more convenient to write the part program so that the center of the tool moves along the shape of the workpiece.

Since all cutting tools have a diameter, a program written for moving the center of the tool will not cut the workpiece to the proper size. To produce a workpiece that has the correct size, offset the path of the tool center from the shape of the workpiece by an amount that equals the radius of the tool.

The control offers a function, called cutter compensation, for offsetting tool paths by the amount of a tool radius. Put the radius of the tool into the offset tables in advance (see section 3.1 or 20.5). This function lets the control use the same program to produce the same workpiece regardless of the radius of the tool that does the cutting.

Figure 21.1
Tool Radius Diameter Compensation



We use these terms in this section:

- **inside** -- An angle between two intersecting programmed tool paths is referred to as inside if, in the direction of travel, the angle measured clockwise from the second tool path into the first is less than or equal to 180 degrees. If one or both of the moves are circular, the angle is measured from a line tangent to the tool path at their point of intersection.
- **outside** -- An angle between two intersecting programmed tool paths is referred to as outside if, in the direction of travel, the angle measured clockwise from the second tool path into the first is greater than 180 degrees. If one or both of the moves are circular, the angle is measured from a line tangent to the tool path at their point of intersection.
- **r** -- cutter radius
- **CR** -- cross-point between two programmed paths after the cutter compensation is activated

Two types of cutter compensation are available on the control:

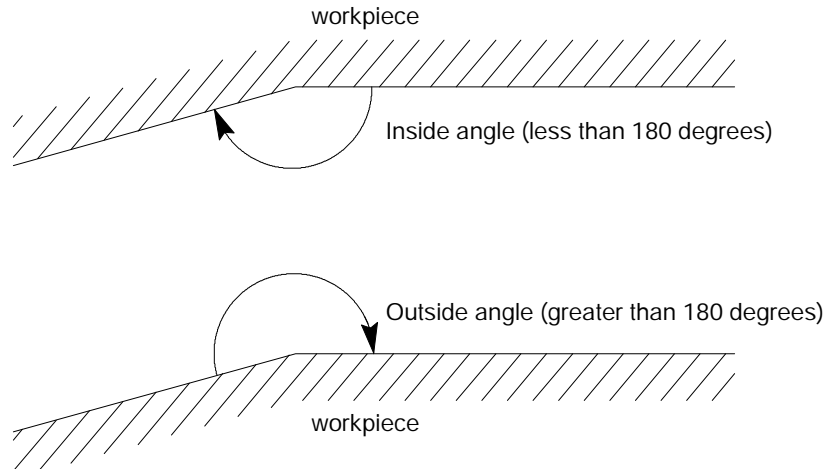
- **type A** (as described in the section 21.3)
- **type B** (as described in section 21.4)

This table highlights the differences between the two types:

Type of Move	Type A	Type B
Entry Move Into Cutter Compensation	The tool takes the shortest possible path to its offset position.	<ul style="list-style-type: none"> • The tool stays at least one radius away from the start point of the next block at all times. • Extra motion blocks can be generated to attempt to prevent gouging of the part as may occur in Type A.
Tool Path	Same	
Exit Move From Cutter Compensation	The tool takes the shortest path to the end-point of the exit move for both inside and outside corners.	<ul style="list-style-type: none"> • The tool takes the shortest path to the end-point of exit move for inside corners only. • For outside corners, the tool stays at least one radius away from the end-point.

The system installer determines whether type A or type B is used by a control in AMP.

Figure 21.2
Definition of Inside and Outside



21.1 Active Cutter Compensation

Use these G-codes for cutter compensation:

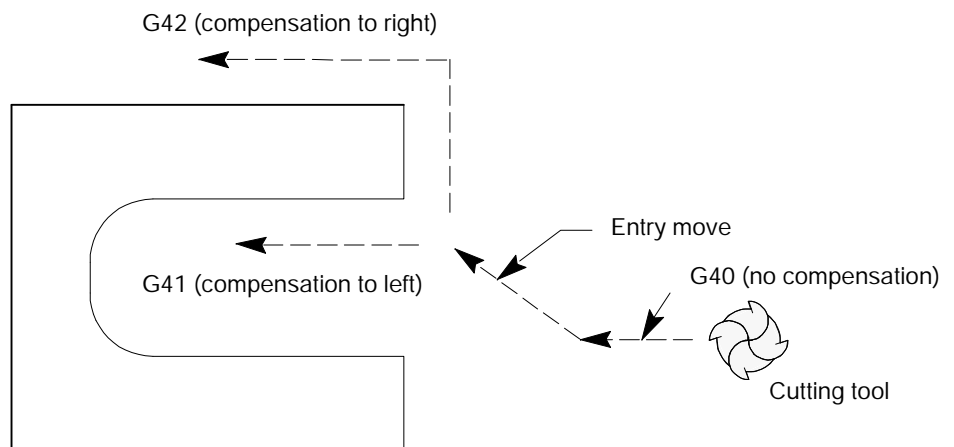
- G41 -- cutter compensation, left
- G42 -- cutter compensation, right
- G40 -- cutter compensation, cancel

Left or right is defined as offsetting the tool to the left or right of the programmed cutting path when facing the direction of cutter motion.

Important: If a negative value is set in the offset tables as the tool radius, compensation direction (tool left or right) is reversed for G41 and G42. G41 and G42 are also reversed during the mirroring operation (see chapter 14).

All of these G-codes are modal and belong to the same modal group.

Figure 21.3
Cutter Compensation Direction



Program the cutter compensation function with the following format:

G41 (or G42) X ___ Y ___ Z ___ D ___ ;

Where :	Is :
G41 (or G42)	cutter compensation direction, G41=left, G42=right
X, Y, Z	End-point of entry move into cutter compensation. Program an entry move on axes only in the currently active plane. Axis motion must take place in order for cutter compensation to be active on an axis.
D	Designates the offset numbers and pulls data: 1) from the wear and geometry tables for the tool radius, and 2) from the geometry table for tool orientation (see section 20.2.1 for information on programming a D word). The D word is optional in the G41 or G42 blocks. The D word may also be designated in any previous or following program block.

Cutter compensation can be programmed in various ways. Following are examples of 1-, 2-, and 3-block programs activating cutter compensation with entry moves.

Example 21.1
Initializing Cutter Compensation

Assume: G17 (XY Plane Selection)

Program Block	Comment
One Block G42 D1 X1 Y1;	Sets compensation right, selects tool radius offset number, and activates move to X1 Y1
Two Blocks D1; G42 X1 Y1;	Selects tool radius offset number Sets compensation right and activates move to X1 Y1
Three Blocks D1; G42; X1 Y1;	Selects tool radius offset number Sets compensation right Activates move to X1 Y1

Important: Any entry move (see section 21.3.1 or 21.4.1) into cutter compensation must be a linear move. Initial activation of cutter compensation by programming of either the G41 or G42 commands in a circular cutting mode (G02 or G03) is not allowed. However, if cutter compensation is already active, the G41 or G42 commands may be programmed in a circular block to change cutter compensation direction either left (G41) or right (G42).

Example 21.2
Cutter Compensation Sample Paths

All of the following blocks result in the same tool path. Assume the selected plane is the XY plane.

```
N1D1X0Y0;  
N2G41X1Y1;  
N3X2;  
M30;  
or  
N1X0Y0F500;  
N2G41X1Y1D1;  
N3X2;  
M30;  
or  
N1X0Y0F500;  
N2G41;  
N3X1Y1D1;  
N4X2  
M30;
```

Important: The cutter compensation feature is not available for any motion blocks that are programmed in MDI mode (see section 21.6.5). The cutter compensation mode may be altered by programming either G41, G42, or G40; or the tool radius may be changed in an MDI program. However, none of the tool paths executed in MDI is compensated. Any changes made to cutter compensation are applied until the next block executed in automatic mode.

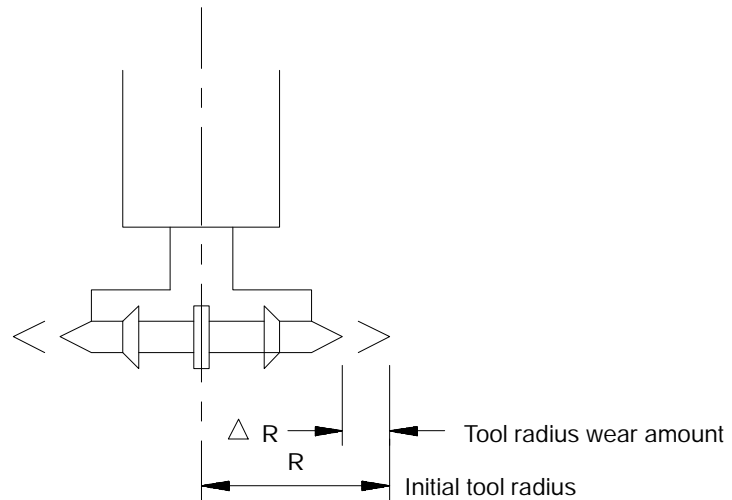
The D-word calls the following data from the offset tables:

- initial cutter radius data
- cutter radius wear offset data

The sum of these two types of offset data is used by the control as the data for the cutter compensation function.

Unless Cutter Compensation is active, when a program recover is performed, the control automatically returns the program to the beginning of the block that was interrupted. In the case of power failure, the control will even reselect the program that was active prior to the interruption.

Figure 21.4
Tool Radius Wear



G40 (cutter compensation cancel) is active when power is turned on, E-STOP is reset, when the control is reset, or when an M02 or M30 end-of-program block is executed.

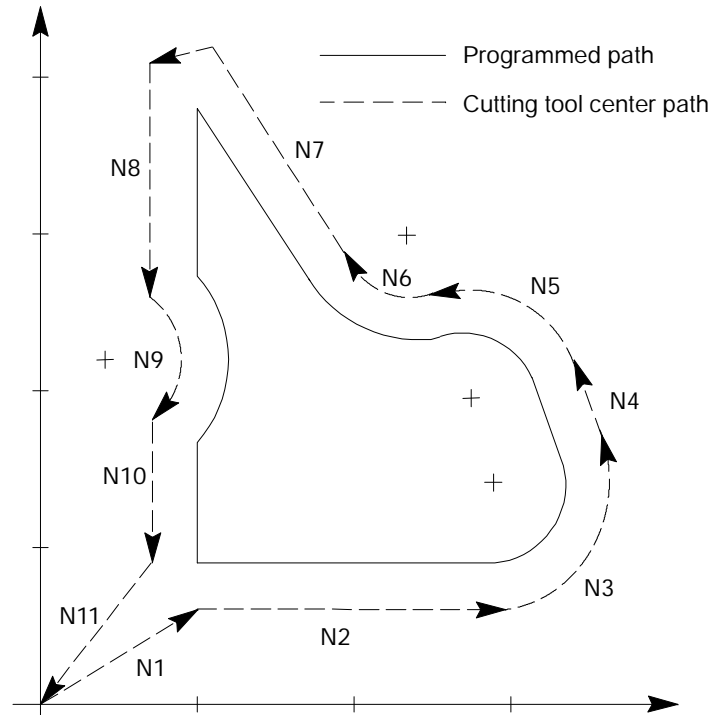
Example 21.3
Cutter Compensation Sample Path

Assume: D01 = 5mm

Program Block	Comment
G00X0Y0;	Establish current position as X0Y0
N1G00G42X20. Y20. D01;	Rapid to start and set compensated right
N2G01X70. F1000. ;	Feed move 1
N3G03X82. 99Y42. 5R15. ;	Move 2
N4G01X72. 99Y62. 5;	Move 3
N5G03X59. 33Y66. 16R15;	Move 4
N6G02X38. 521Y69. 797R16;	Move 5
N7G01X20. Y95. ;	Move 6
N8Y71. 18;	Move 7, creates a generated block
N9G02Y48. 82R15. ;	Move 8
N10G01 Y20. ;	Move 9, cutter moves away from path at end

N11G00G40X0Y0D00; Rapid to start point and cancel compensation
N12M30; End of Program

Figure 21.5
Results of Cutter Compensation Program Example



21.2
Cutter Compensation
Generated Blocks
G39, G39.1

In certain instances, cutter compensation creates a non-programmed move called a generated block. These blocks improve cycle time and corner-cutting quality.

Cutter Compensation generates blocks for type A or B moves as follows:

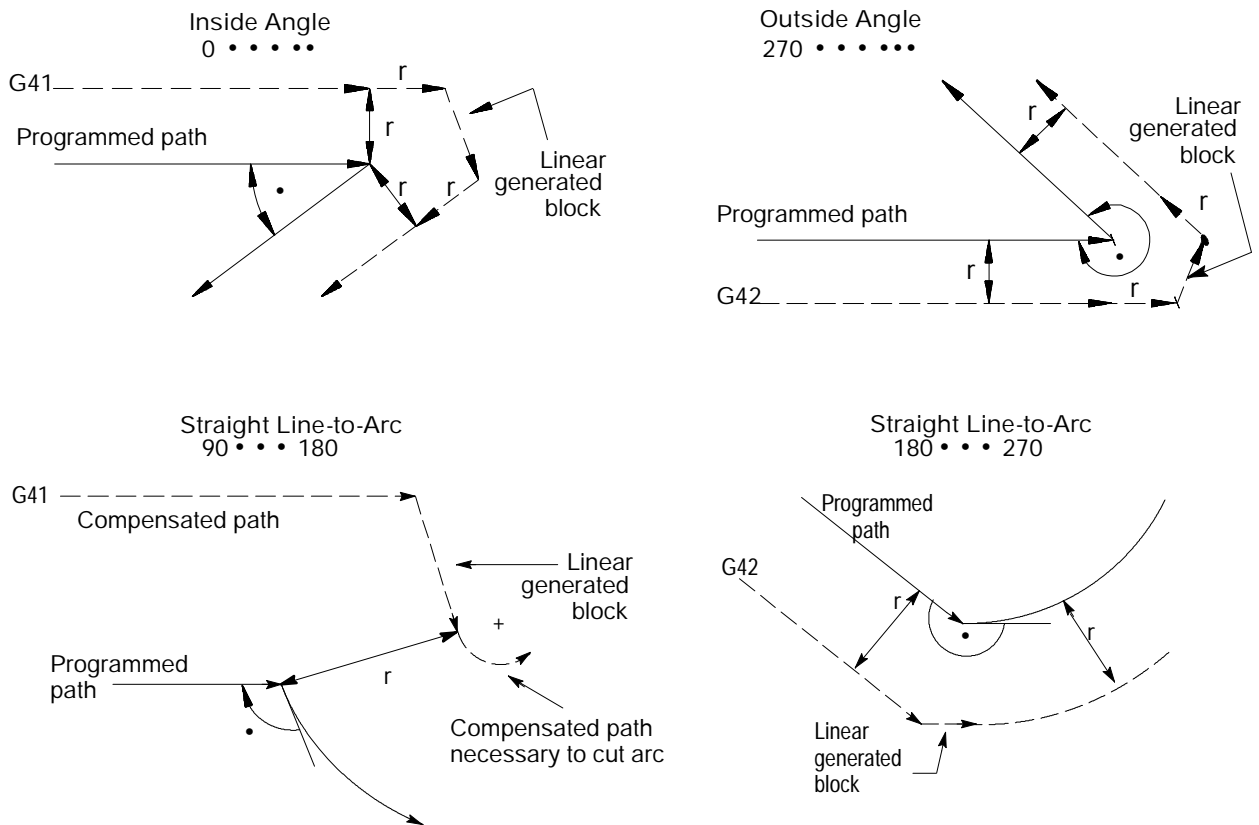
Type of Move	Type A	Type B
Entry Move	No block is generated	Block is generated
Tool Path	Block is generated	Block is generated
Exit Move	No block is generated	Block is generated

Important: Cutter compensation generated blocks, as shown in Figure 21.6, are created only under these conditions:

When	is active and is cutting:	which is:
G41	an inside angle	less than 90 degrees
G42	an outside angle	greater than 270

When	is active and is cutting:	which is:
G41	straight line to arc (or arc to straight line)	greater than 90 degrees but less than 180 degrees
G42	straight line to arc (or arc to straight line)	greater than 180 degrees but less than 270 degrees

Figure 21.6
Cutter Compensation Examples

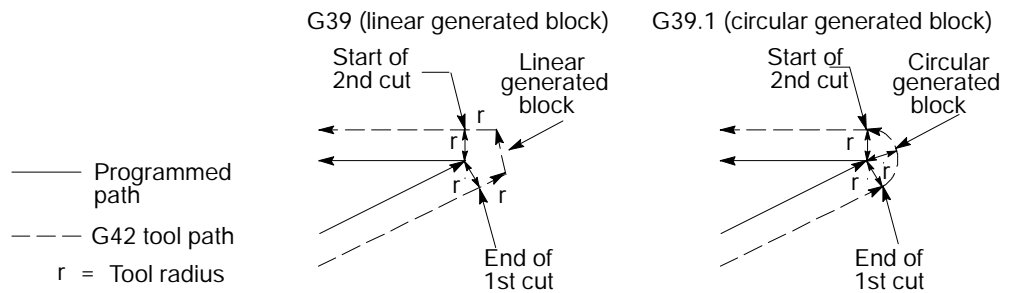


Besides choosing between types A and B (selected in AMP), cutter compensation generated blocks can also be controlled by programming a G39 or G39.1. These G-codes determine whether the generated block will be linear (G39) or circular (G39.1) as shown in Figure 21.7.

G39(or G39.1);

Where :	Causes:
G39	linear generated blocks. If neither G39 nor G39.1 is programmed, G39 is the default. This command is modal.
G39.1	circular generated blocks. When cutting straight line-to-arc or arc-to-straight line moves, the generated block will <u>always</u> be linear and the G39.1 will be ignored. This command is modal.

Figure 21.7
Cutter Compensation Generation Blocks (G39 vs G39.1)



G39 or G39.1 can be programmed in any block. However, they must be programmed in or before the block that causes a cutter compensation generated block.

Important: For linear generated blocks, the system installer can define a minimum block length in AMP. If the generated move length is less than the system-defined minimum block length, no generated block is created. The tool path proceeds to the intersection of the two compensated paths. If the generated move length is equal to or greater than the system-defined minimum block length, a generated block is created.

Throughout this chapter, we show drawings where a generated block is created. Both G39 and G39.1 are shown in these drawings where applicable.

21.3 Cutter Compensation (Type A)

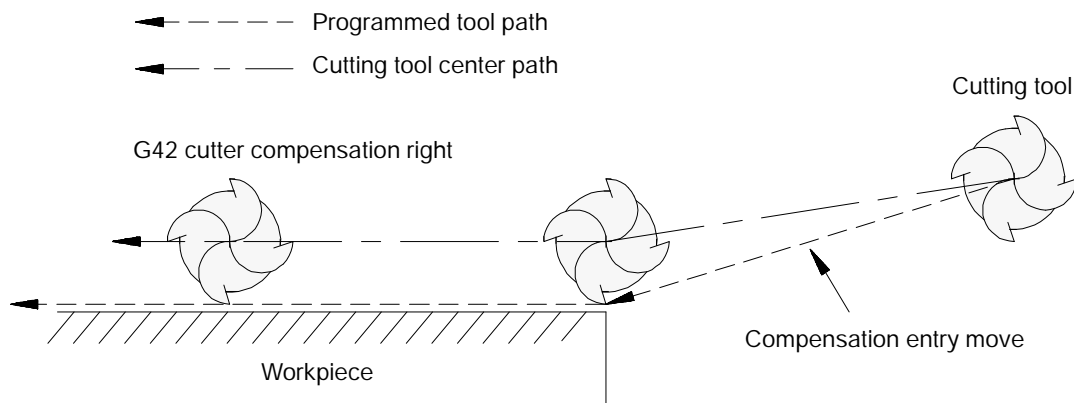
The easiest way to demonstrate the actual tool paths taken by the cutting tool when using cutter compensation type A is by pictorial representation. The following subsections give a brief description of the cutter path, along with a figure to clarify the description.

21.3.1 Cutter Compensation Type A Entry Moves

An entry move is defined as the path that the cutting tool takes when the cutter compensation function first becomes activated in a program. Figure 21.8 gives an example of a typical entry move.

Important: Any entry move into cutter compensation must be a linear move. Initial activation of cutter compensation by designation of either the G41 or G42 commands in a circular cutting mode (G02 or G03) is not allowed. The G41 or G42 commands may be designated in a circular block to change cutter compensation direction, as long as cutter compensation is already active.

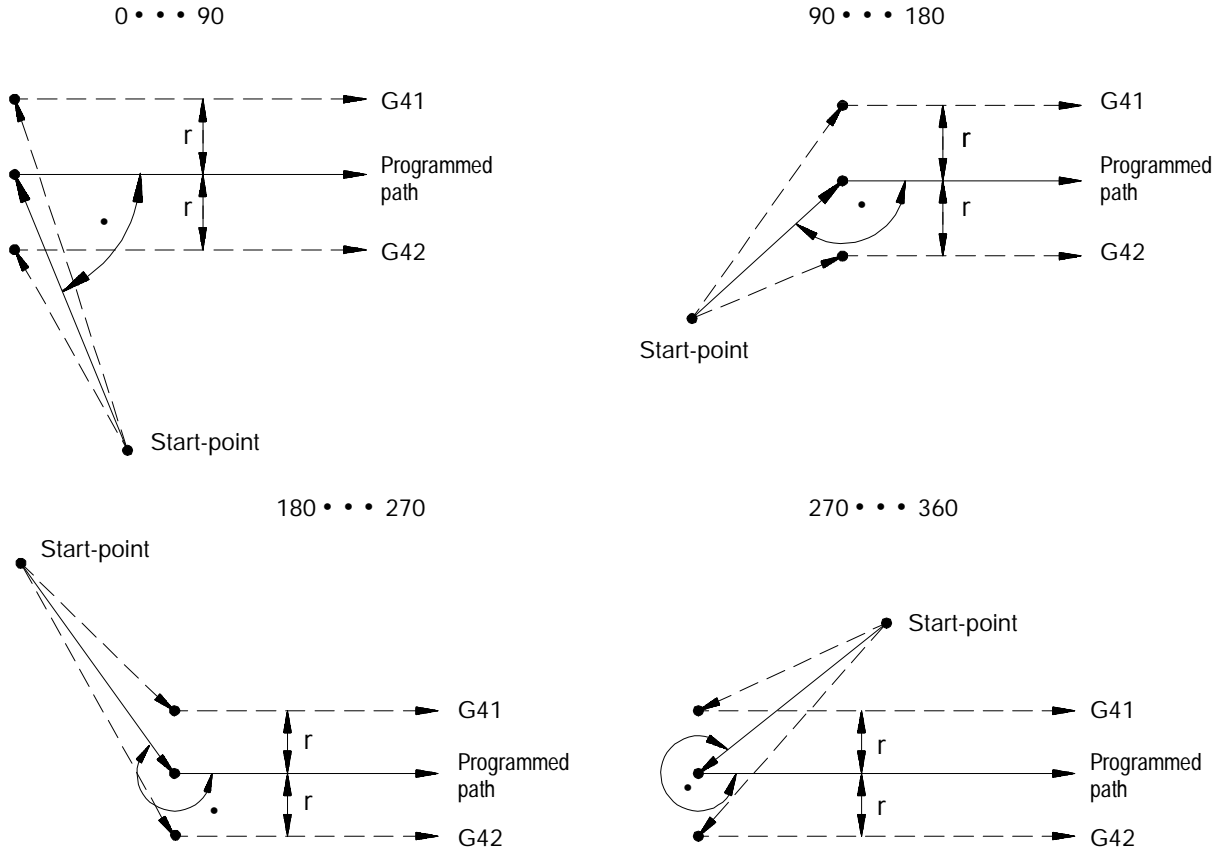
Figure 21.8
Cutter Compensation Entry Move



The entry move of the cutting tool for type A cutter compensation takes the shortest possible path to its offset position. This position is at right angles to and on the left or right side of the next programmed move in the currently defined plane.

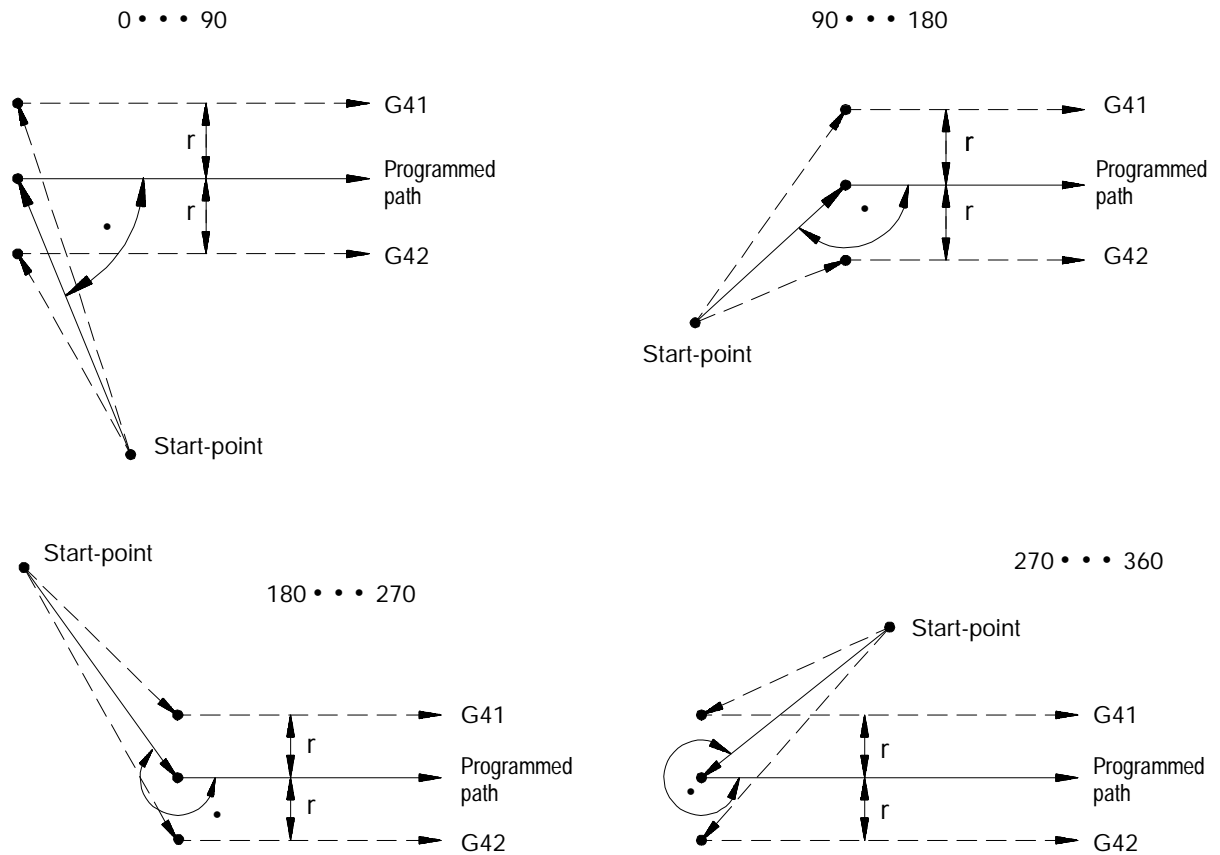
Figure 21.9 through Figure 21.11 show examples of typical entry moves using type A cutter compensation:

Figure 21.9
Tool Path for Entry Move Straight Line-to-Straight Line



If the next programmed move is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the start-point of that circular move.

Figure 21.10
Tool Path for Entry Move Straight Line-to-Arc



There is no limit to the number of blocks that may follow the programming of G41 or G42 before an entry move takes place. The entry move is always the same regardless of the number of blocks that do not program motion in the current plane for compensation.

Example 21.4
Sample Entry Move After Non-Motion Blocks

Assume current compensation plane is the XY plane.

N1X0Y0F500;	
N2G41D1;	This block commands compensation left.
N3Z1;	This is not the entry block since no axis motion takes place in the current plane.
N4...;	No axis motion in current plane.
N5...;	No axis motion in current plane.
N6...;	No axis motion in current plane.
"	"
"	"
"	"
N999X1Y1;	This is the entry move for the previously programmed G41.
N1000M30;	

The system installer selects in AMP the maximum number of non-motion blocks to be allowed during cutter compensation before the entry move must be re-initialized (see section 21.6.2).

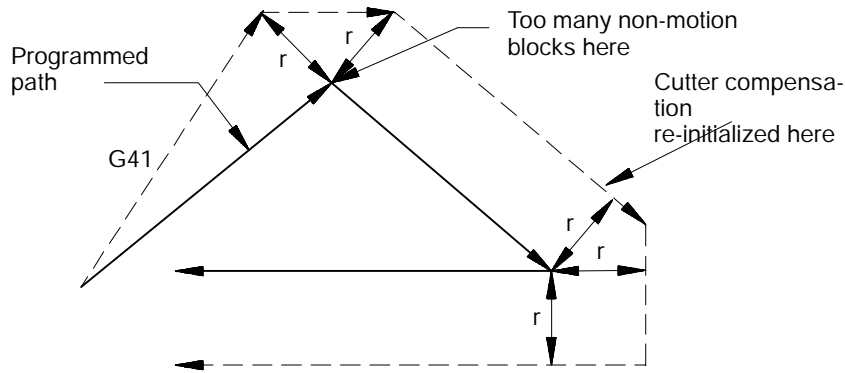
For example, assume that the system installer has designated that only 2 non-motion blocks may be performed. If more than two blocks following the entry move do not contain axis motion in the current plane, then the entry move is re-performed at the next block containing axis motion in the current plane.

Example 21.5
Too Many Non-Motion Blocks After Entry Block

Assume current plane to be the XY plane and the system installer has designated that only 2 non-motion blocks may be performed before cutter compensation is re-initialized.

N1X0Y0F500;	
N2G41D1X1Y1;	Entry move cutter compensation left.
N3Z2;	No axis motion in current plane.
N4...;	No axis motion in current plane.
N5X4Y-4;	New entry move cutter compensation left.
N6M30;	

Figure 21.11
Entry Move Followed by Too Many Non-Motion Blocks



21.3.2 Cutter Compensation Type A Exit Moves

The cutter compensation feature is cancelled by programming G40. The path that is taken when the tool leaves cutter compensation is referred to as the exit move. The path that the tool follows during an exit move is dependant on:

- The direction of compensation (G41 or G42)
- The angle between the last motion made in cutter compensation (in the current compensation plane) and the motion of the of the exit move

Designating a tool offset number D00 in a program does not cancel cutter compensation and does not generate an exit move. Cutter compensation simply continues on as if a tool radius had been changed to a radius of zero (see section 21.6.4 on changing cutter radius). The exit move, if D00 is the active tool radius, is then equal to the programmed tool path.

Important: An exit move cannot be a circular move (G02 or G03). Any exit move must be programmed on a linear path. Any attempt to generate an exit move using a circular path generates a block-format error.

Example 21.6 gives some sample exit move program blocks.

Example 21.6
Type A Sample Exit Moves

Assume the current plane to be the XY plane and cutter compensation is already active before the execution of block N100 in the following program segments.

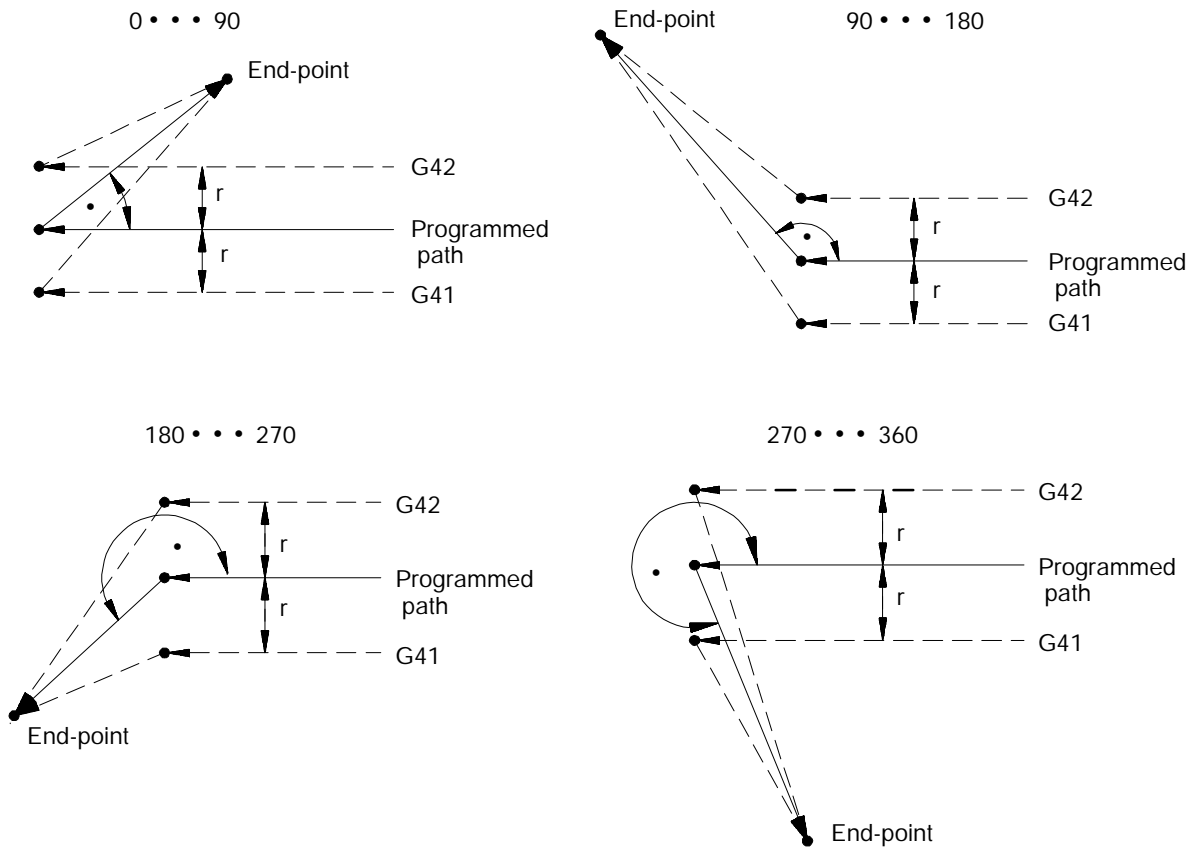
N100X1Y1;	
N110X3Y3G40;	Exit move.
N100X1Y1;	
N110G40;	Exit move.
N120X3Y3;	
N100X1Y1;	
N110G40;	
N120Z1;	No axis motion in the current plane.
N130...;	No axis motion in the current plane.
N140...;	No axis motion in the current plane.
"	"
"	"
N200X3Y3;	Exit move.
N100X1Y1;	
N110Z1;	No axis motion in the current plane.
N120...;	No axis motion in the current plane.
N130...;	No axis motion in the current plane.
"	"
"	"
N200G40X3Y3;	Exit move.

All of the program blocks in Example 21.6 produce the same exit move provided that the number of non-motion blocks in the compensation mode has not exceeded a value selected by the system installer in AMP.

The exit of the cutting tool for type A cutter compensation takes the shortest possible path to the end-point of the exit move. This path starts at right angles to the left or right of the end-point (depending on G41 or G42) of the last move in the currently defined plane (it is possible to redefine this start-point using an I, J, and/or K word as described later in this section). The end-point of the exit move is no longer offset to the left or right.

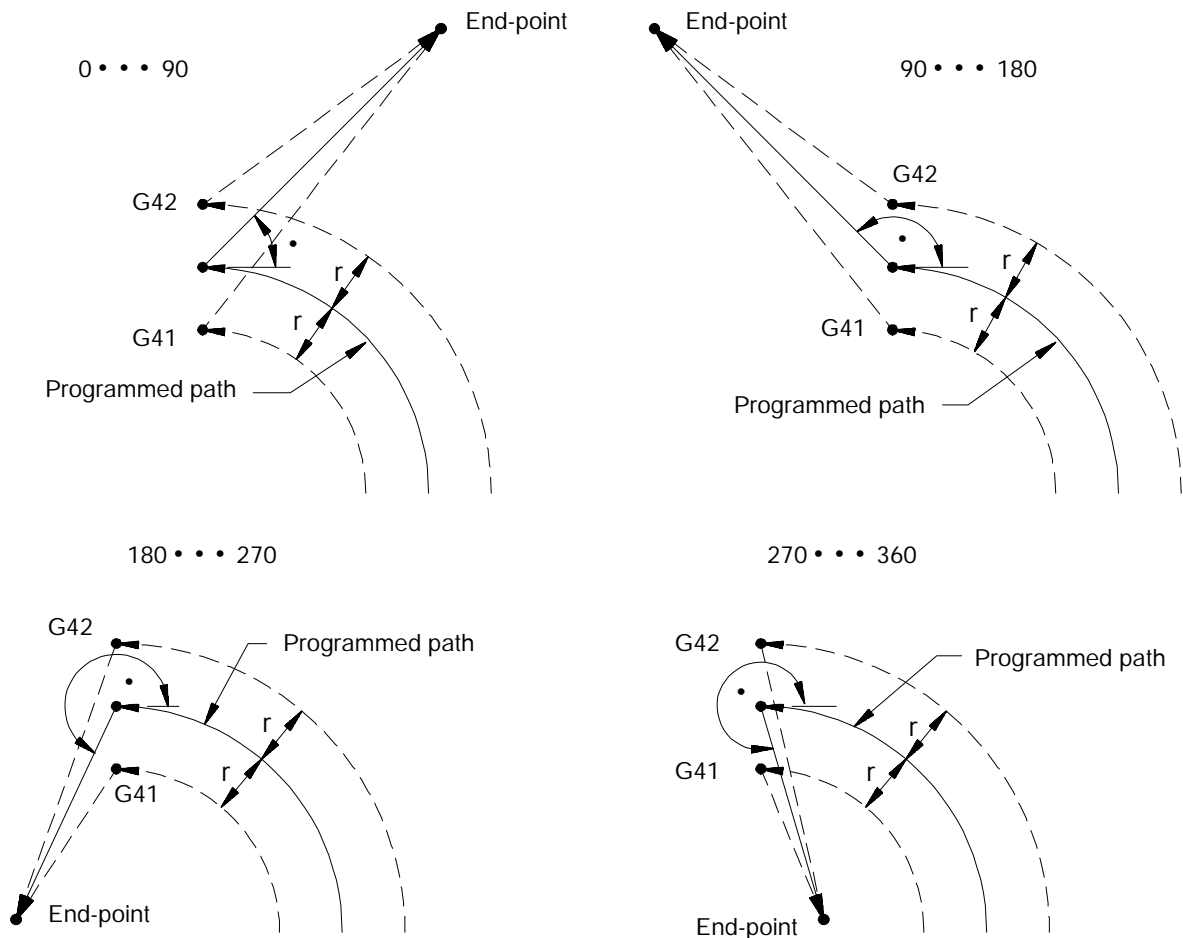
Figure 21.12 through Figure 21.16 show examples of typical exit moves using type A cutter compensation. All examples assume that the number of non-motion blocks before the designation of the G40 command have not exceeded the number allowed as determined by the system installer in AMP.

Figure 21.12
 Tool Path for Exit Move Straight Line-to-Straight Line



If the last programmed move prior to the exit move (which must be linear) is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the end-point of that circular move.

Figure 21.13
Tool Path for Exit Move Arc-to-Straight Line



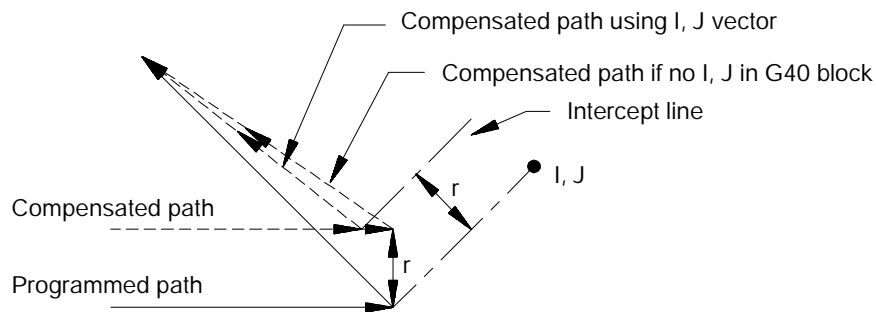
The above examples in Figure 21.12 and Figure 21.13 assume that the number of blocks that do not contain axis motion in the current plane, following the programming of G40 before an exit move takes place, does not exceed an amount selected in AMP by the system installer. If the number of non-motion blocks following G40 exceeds the limit, the control generates its own exit move. This may often cause over-cutting of the part.

It is possible to modify the path that the tool takes for an exit move by including an I, J, and/or K word in the exit move. Only the I, J, or K words that represent values in the current plane are programmed in the block containing the exit move. I, J, and K correspond to the X, Y, and Z axes respectively.

The I, J, and K words in the exit move block define a vector that is used by the control to redefine the end-point of the previous compensated move. I, J, and K words are always programmed as incremental values regardless of the current mode (G90 or G91).

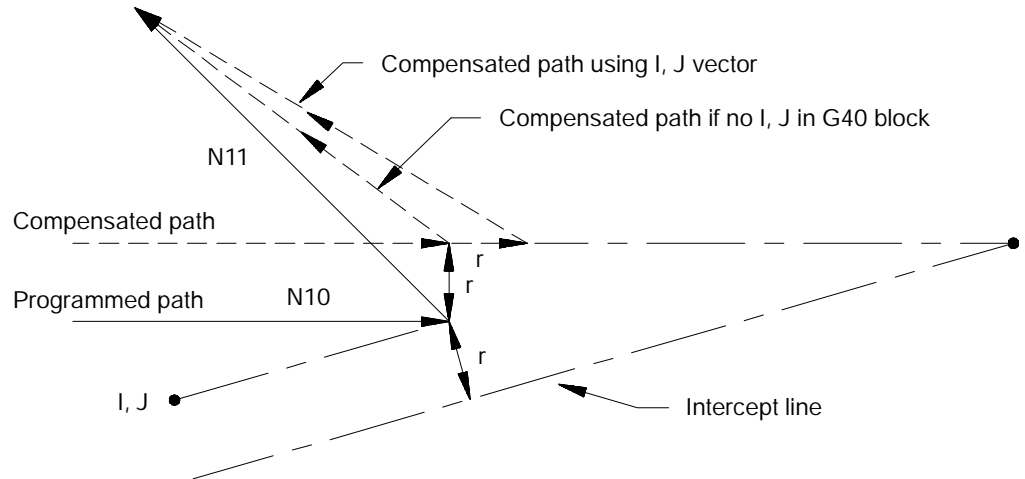
The vector defined by the I, J, and/or K words is along a line drawn from the end-point of the programmed path to a point referenced from the end-point of the programmed path a distance along the axes in the current plane an amount as designated with the I, J, and/or K words. A new vector is then defined parallel to the vector defined by the I, J, and/or K word and offset from this vector in the direction and amount of the currently active offset (G41 or G42). The intersection of this new vector with the current compensated tool path defines a point which is the new end-point of the last programmed compensation move.

Figure 21.14
Exit Move Defined By An I, J, K Vector



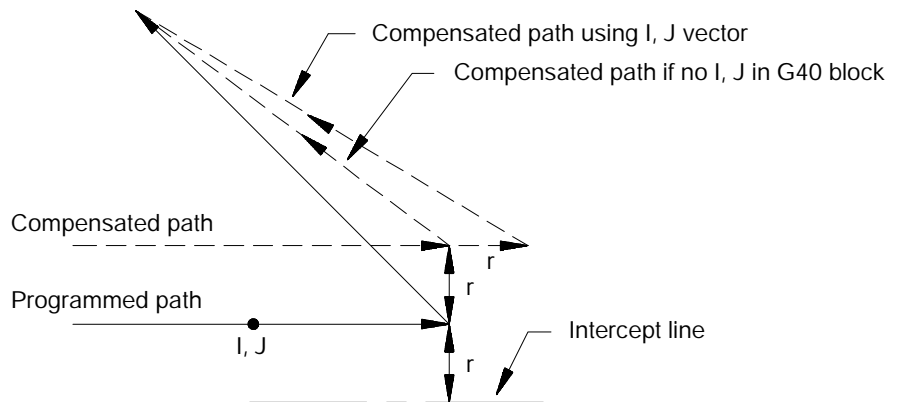
Exception is taken to the above figure when the change in length of the compensated path is more than one radius of the tool. In this special case, this offset is limited to one radius of the tool. The direction of the offset is towards the point of intersection of the I, J, K vector with the current compensated tool path.

Figure 21.15
Exit Move Defined By An I, J, K Vector But Limited To Radius



If the vector defined by I, J, and/or K is parallel to the programmed tool path, the resulting exit move are offset in the opposite direction of the I, J, K vector by one radius of the tool.

Figure 21.16
Exit Move When I, J, K Vector Parallel to Programmed Tool Path



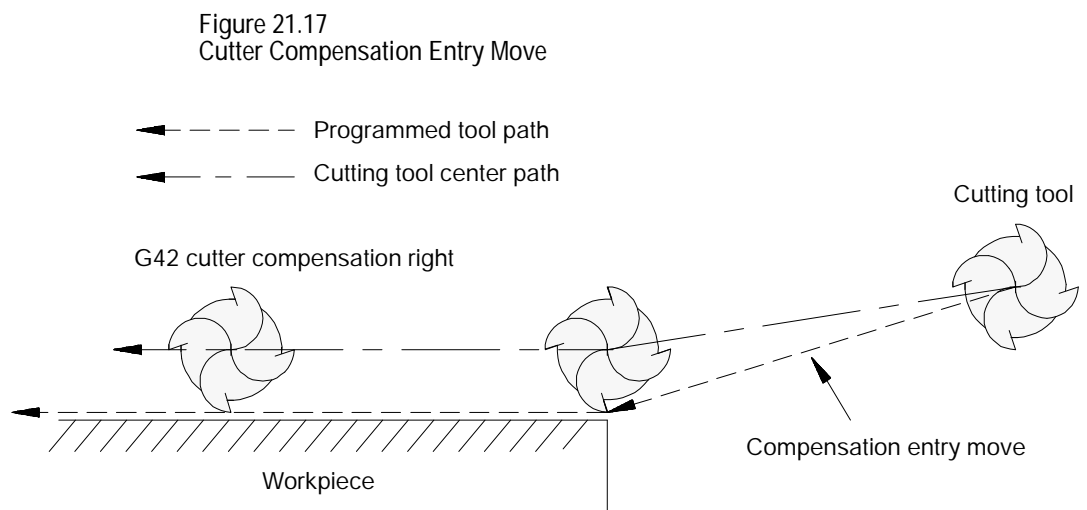
Important: If one I, J, and/or K value is programmed without the second one, the value of the second I, J, and/or K word defaults to 0.

21.4 Cutter Compensation (Type B)

The easiest way to demonstrate the actual tool paths taken by the cutting tool when using cutter compensation type B is by pictorial representation. The following subsections give a brief description of the cutter path along with a figure to clarify the description.

21.4.1 Cutter Compensation Type B Entry Moves

An entry move is defined as the path that the cutting tool takes when the cutter compensation function first becomes activated in a program. The following figure gives an example of a typical entry move:

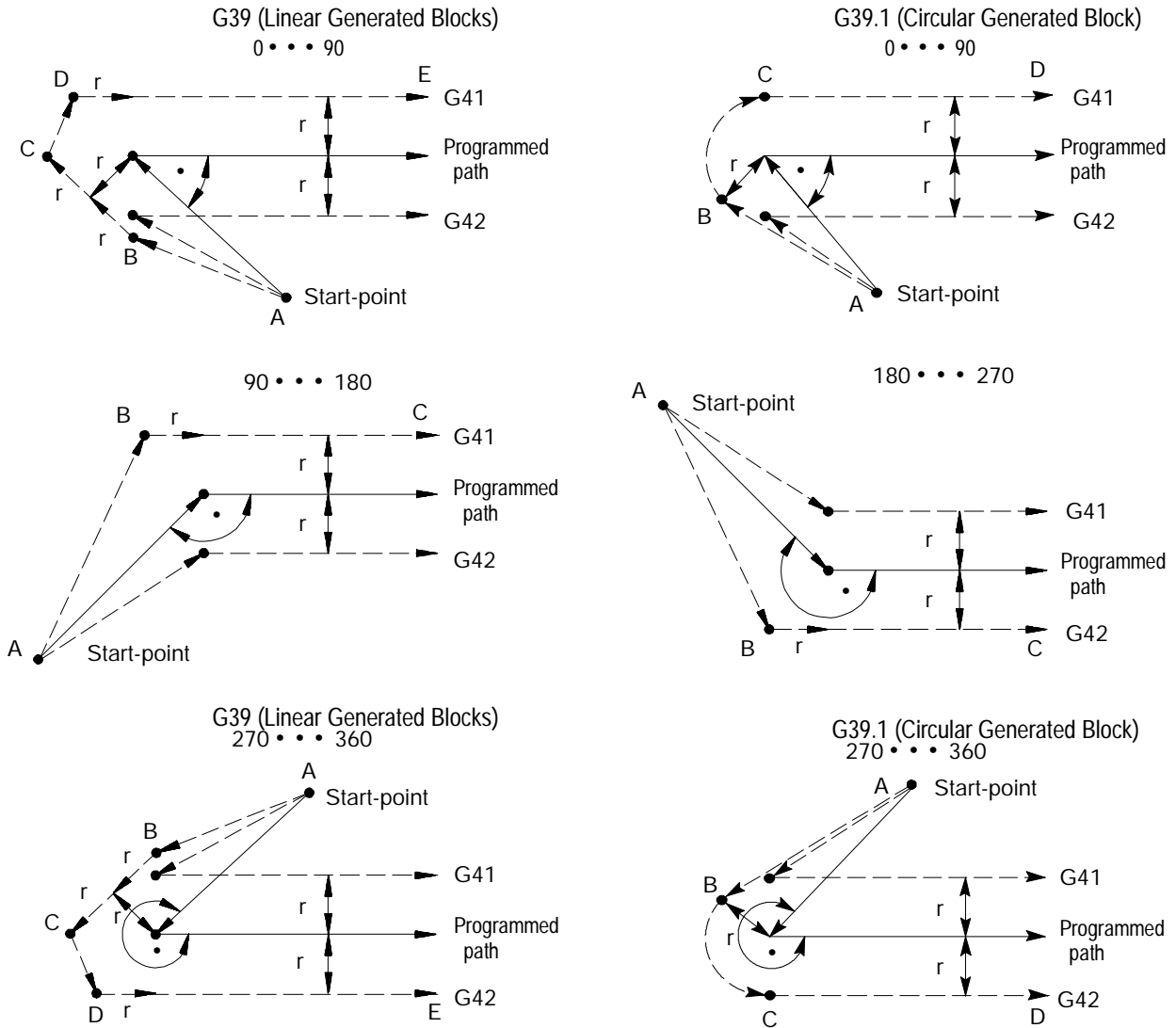


Important: Any entry move into cutter compensation must be a linear move. Initial activation of cutter compensation by designation of either the G41 or G42 commands in a circular cutting mode (G02 or G03) is not allowed. The G41 or G42 commands may be designated in a circular block to change cutter compensation direction, as long as cutter compensation is already active.

The entry move of the cutting tool for type B cutter compensation can generate extra motion blocks to attempt to prevent gouging of the part as may sometimes occur using compensation type A. Type B cutter compensation keeps the cutting tool at least one radius away from the start-point of the next block at all times during an entry move. The final end-point of the entry move is a position at right angles to and on the left or right side of the next programmed move in the currently defined plane.

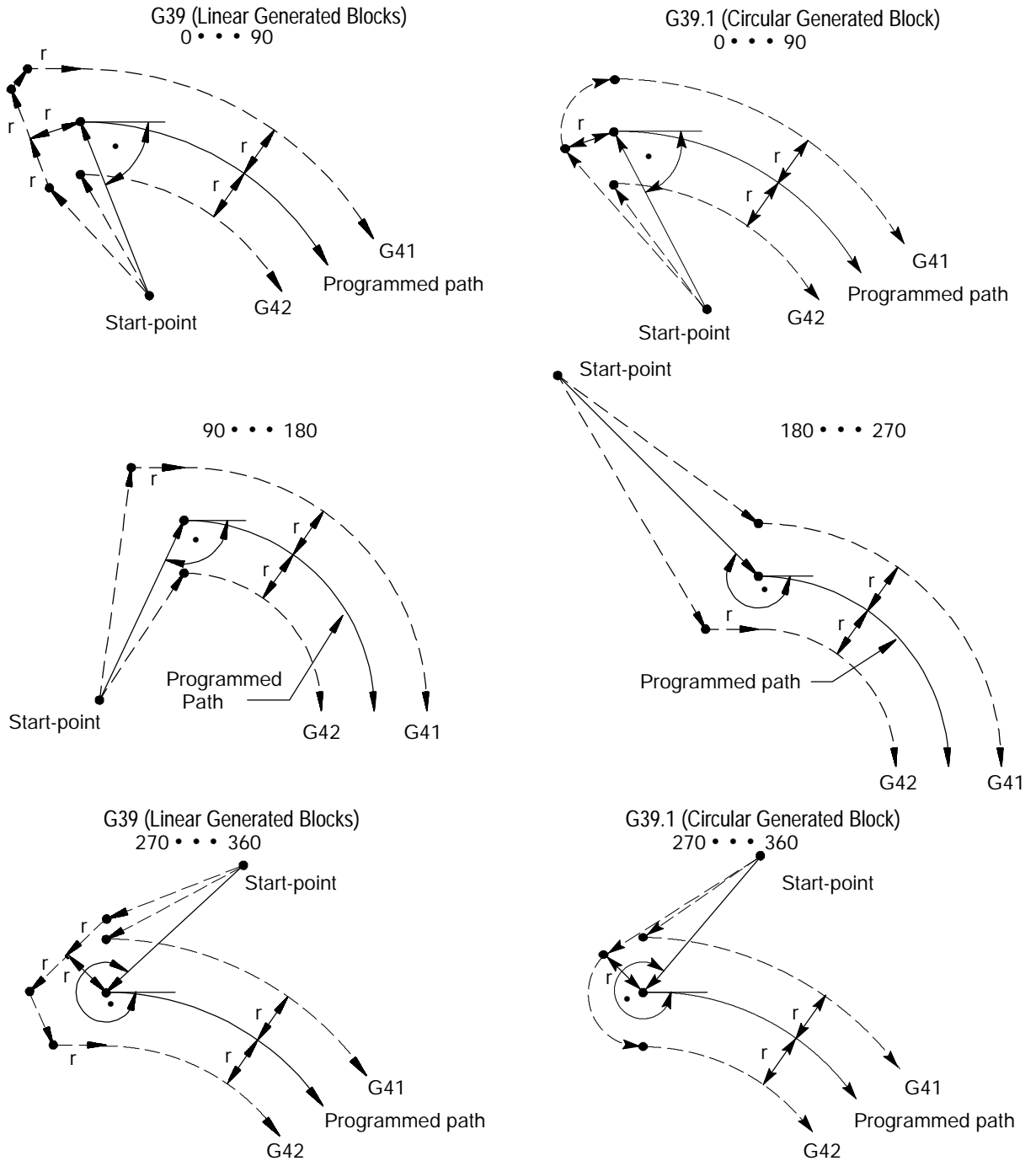
Figure 21.18 and Figure 21.19 show examples of typical entry moves using type B cutter compensation:

Figure 21.18
Tool Path for Entry Move Straight Line-to-Straight Line



If the next programmed move is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the start-point of that circular move.

Figure 21.19
 Tool Path For Entry Move Straight Line-to-Arc



There is no limit to the number of blocks that may follow the programming of G41 or G42 before an entry move takes place. The entry move are always the same regardless of the number of blocks that do not program motion in the current plane for compensation.

Example 21.7
Sample Entry Move After Non-Motion Blocks

Assume current compensation plane is the XY plane.

```

N01X0Y0F500;

N2G41D1;      This block commands compensation left.
N3Z1;         This is not the entry block since no axis
              motion takes place in the current plane.
N4...;        No axis motion in current plane.
N5...;        No axis motion in current plane.
N6...;        No axis motion in current plane.
"            "
"            "
"            "
N999X1Y1;     This is the entry move for the previously
              programmed G41.
M30;

```

The system installer selects in AMP the maximum number of non-motion blocks that are to be allowed during cutter compensation before the entry move must be re-initialized (see section 21.6.2).

For example, assume that the system installer has designated that only two non-motion blocks may be performed. If more than two blocks following the entry move do not contain axis motion in the current plane, then the entry move are re-performed at the next block containing axis motion in the current plane.

Example 21.8
Too Many Non-Motion Blocks After Entry Block

Assume current plane to be the XY plane and the system installer has designated that only two non-motion blocks may be performed before cutter compensation is re-initialized.

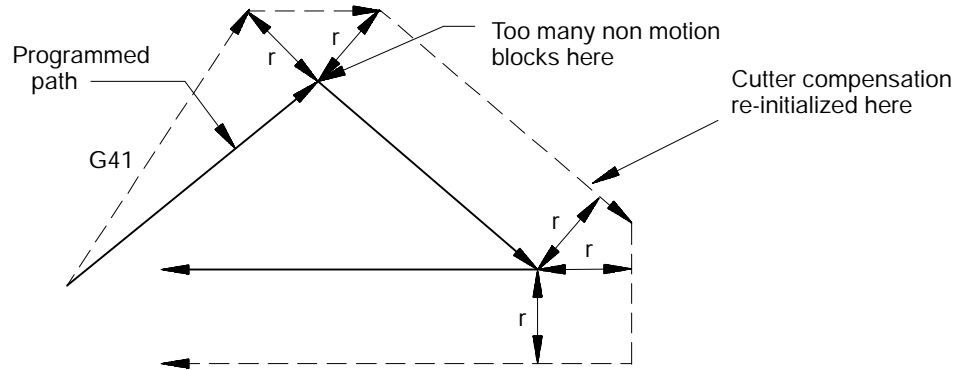
```

N1X0Y0F500;

N2G41D1X1Y1;  Entry move cutter compensation left.
N3Z2;         No axis motion in current plane.
N4...;        No axis motion in current plane.
N5X4Y-4;      New entry move cutter compensation left.
M30;

```

Figure 21.20
Entry Move Followed By Too Many Non-Motion Blocks



21.4.2 Cutter Compensation Type B Exit Moves

The cutter compensation feature is cancelled by programming G40. The path that is taken when the tool leaves cutter compensation is referred to as the exit move. The path that the tool follows during an exit move is dependant on:

- The direction of compensation (G41 or G42).
- The angle between the last motion made in cutter compensation (in the current compensation plane) and the motion of the of the exit move.

Designating a tool offset number D00 in a program does not cancel cutter compensation and does not generate an exit move. Cutter compensation simply continues on as if a tool radius had been changed to a radius of zero (see section 21.6.4 on changing cutter radius). The exit move, if D00 is the active tool radius, is then equal to the programmed tool path.

Important: An exit move cannot be a circular move (G02 or G03). Any exit move must be programmed on a linear path. Any attempt to generate an exit move using a circular path generates a block format error.

Example 21.9 gives some sample exit move program blocks:

Example 21.9
Examples of Exit Move Blocks

Assume the current plane to be the XY plane.

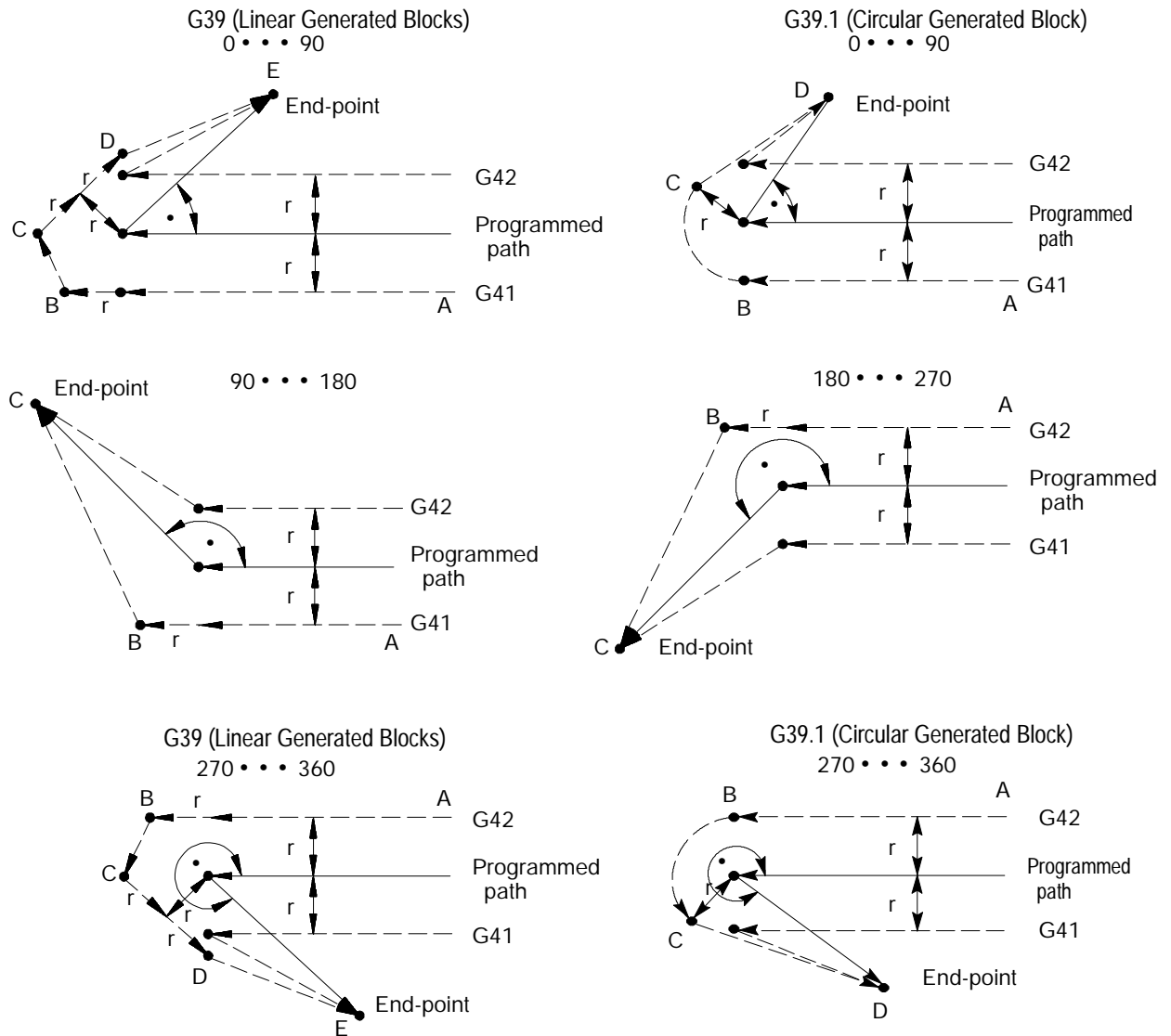
```
N100X1Y1;
N110X3Y3G40;  Exit move.
N100X1Y1;
N110G40;
N120X3Y3;      Exit move.
N100X1Y1;
N110G40;
N120Z1;         No axis motion in the current plane.
N130...;        No axis motion in the current plane.
N140...;        No axis motion in the current plane.
"               "
"               "
N200X3Y3;      Exit move.
N100X1Y1;
N110Z1;         No axis motion in the current plane.
N120...;        No axis motion in the current plane.
N130...;        No axis motion in the current plane.
"               "
"               "
N200G40X3Y3;   Exit move.
```

All of the program blocks in Example 21.9 produce the same exit move provided that the number of non-motion blocks in the compensation mode has not exceeded a value selected by the system installer in AMP.

The exit of the cutting tool for type B cutter compensation takes the shortest possible path to the end-point of the exit move only for inside corners. For outside tool corners, the cutting tool always remains at least the radius of the cutting tool away from the end-point of the last move in compensation. It is possible to redefine the start-point using an I, J, and/or K word as described later in this section. The end-point of the exit move is no longer offset to the left or right.

Figure 21.21 and Figure 21.22 show examples of typical exit moves using type B cutter compensation. All examples assume that the number of non-motion blocks before the designation of the G40 command has not exceeded the number allowed as determined by the system installer in AMP.

Figure 21.21
Tool Path For Exit Move Straight Line-to-Straight Line



If the last programmed move is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the end-point of that circular move.

Figure 21.22
Tool Path For Exit Move Arc-to-Straight Line

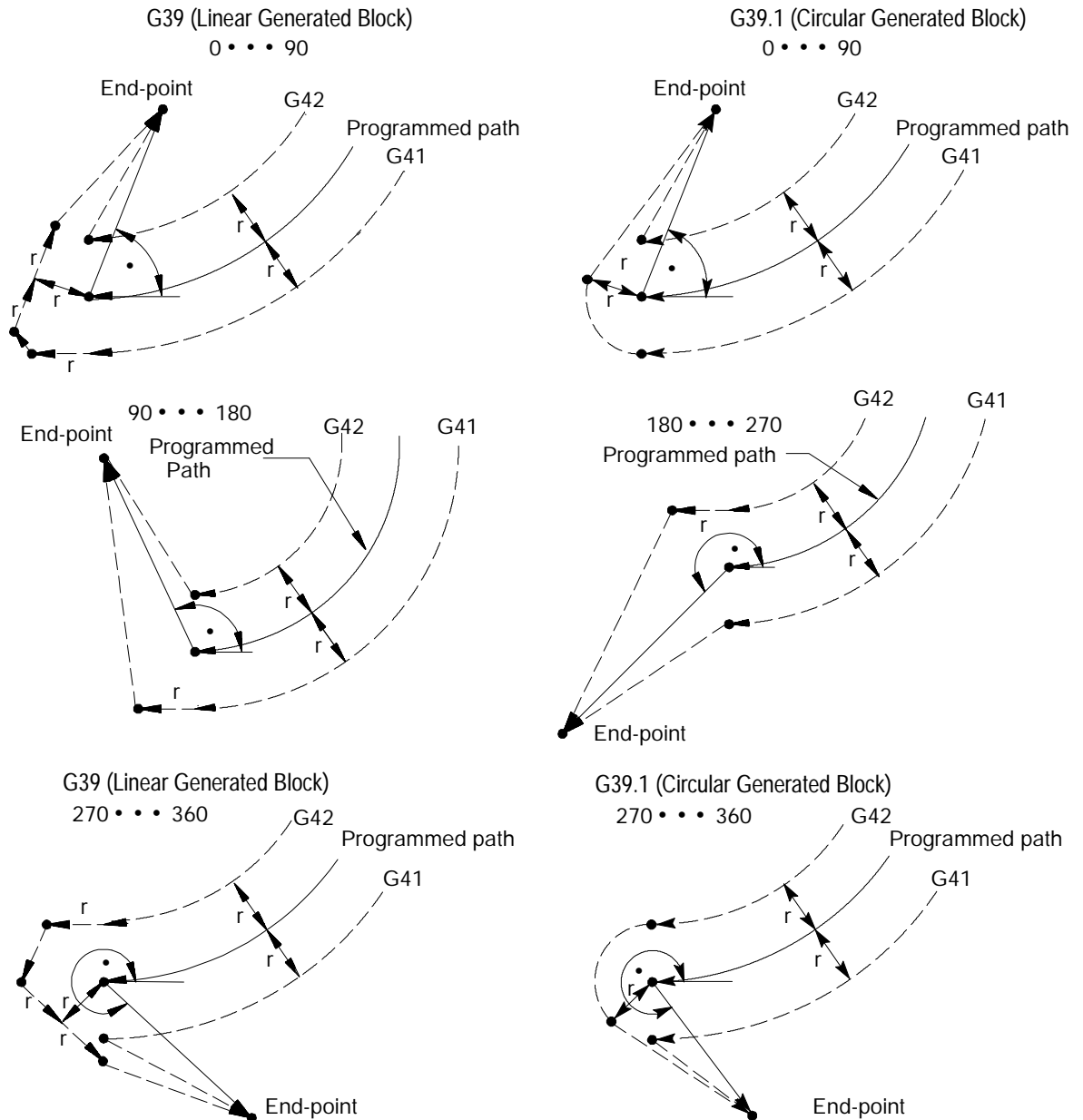


Figure 21.21 and Figure 21.22 assume that the number of blocks not containing axes motion in the current plane, following G40 before the exit move takes place, does not exceed an amount selected in AMP by the system installer. If the number of non-motion blocks following G40 exceeds the limit, the control generates its own exit move. This may often cause over-cutting of the part.

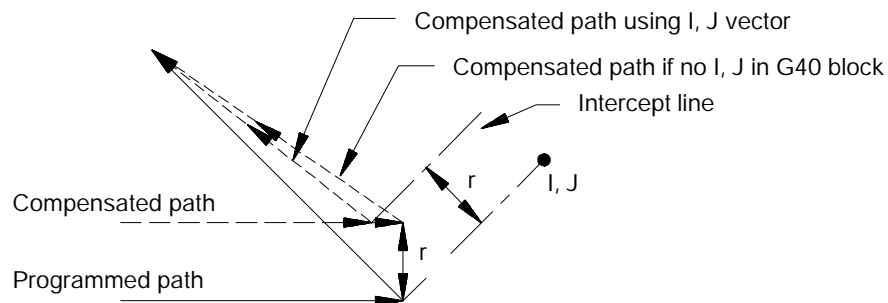
It is possible to modify the path that the tool takes for an exit move by including an I, J, and/or K word in the exit move. Only the I, J, or K words that represent values in the current plane are programmed in the block containing the exit move. I, J, and K correspond to the X, Y, and Z axes respectively.

The I, J, and K words in the exit move block define a vector that is used by the control to redefine the end-point of the previously compensated move. I, J, and K words are always programmed as incremental values regardless of the current mode (G90 or G91).

The vector defined by the I, J, and/or K words is along a line drawn from the end-point of the programmed path a distance as designated with the I, J, and/or K words. The I, J, and/or K words must be in the currently defined plane.

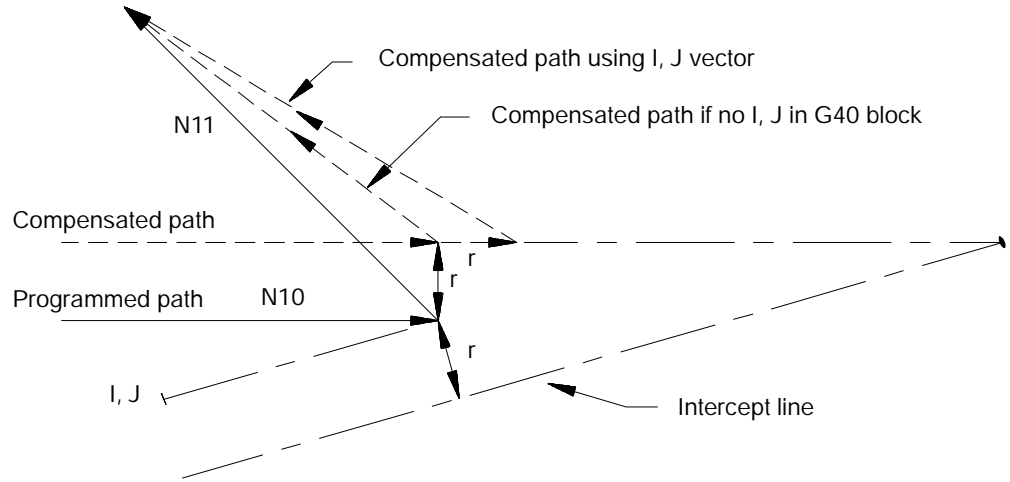
A new vector is then defined parallel to the vector defined by the I, J, and/or K word and offset from this vector in the direction and amount of the currently active offset (G41 or G42). The intersection of this new vector with the current compensated tool path define a point which is the new end-point of the last programmed compensation move.

Figure 21.23
Exit Move Defined by an I, J, K Vector



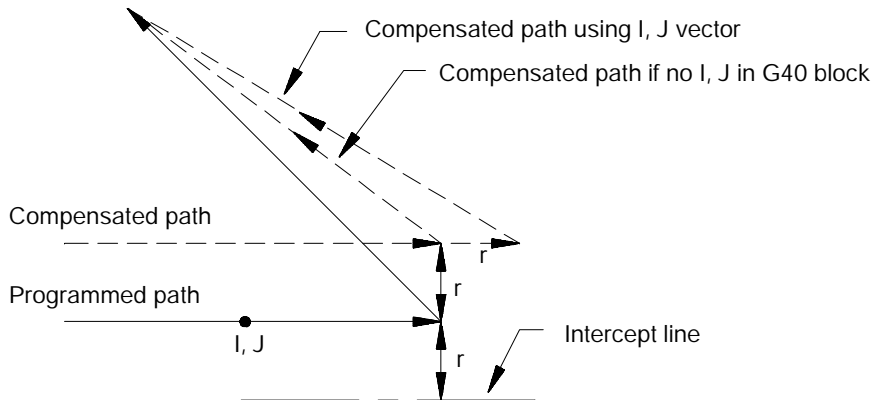
Exception is taken to the above figure when the change in length of the compensated path is more than one radius of the tool. In this special case, this offset is limited to one radius of the tool. The direction of the offset is towards the point of intersection of the I, J, or K vector and the current compensated tool path.

Figure 21.24
Exit Move Defined By An I, J, K Vector But Limited to Tool Radius



If the vector defined by I, J, and/or K is parallel to the programmed tool path, the resulting exit move are offset in the opposite direction of the I, J, K vector by one radius of the tool.

Figure 21.25
Exit Move When I, J, K Vector is Parallel to Programmed Tool Path



Important: If one I, J, and/or K value is programmed without the second one, the value of the second I, J, and/or K word defaults to 0.

21.5 Tool Path During Cutter Compensation

Except for entry and exit moves, the basic tool paths generated during cutter compensation are the same for types A and B cutter compensation. The paths taken are a function of the angle between tool paths (whether G41 tool-left or G42 tool-right is specified) and the radius of the cutting tool.

Important: If at any time during the execution of cutter compensation blocks a block reset is performed, the cutter compensation function is re-initialized, and the next move acts as an entry move as described in an earlier section.

Important: When cutting arcs with cutter compensation active, the control may need to adjust the programmed feedrate to maintain cutting speed. Refer to Chapter 18 for details on feedrates during cutter compensation.

The control generates extra motion blocks when necessary to keep the cutting tool in tolerance of the desired tool path. This becomes necessary when the intersection of tool paths is an outside tool path (as defined in section 21.0) that has an angle as follows:

- between 0 and 90 degrees during cutter compensation left (G41)
- between 270 and 360 degrees during cutter compensation right (G42)

Figure 21.26 through Figure 21.29 illustrate the basic motion of the cutting tool as it executes program blocks during cutter compensation:

Figure 21.26
Cutter Compensation Tool Paths Straight Line-to-Straight Line

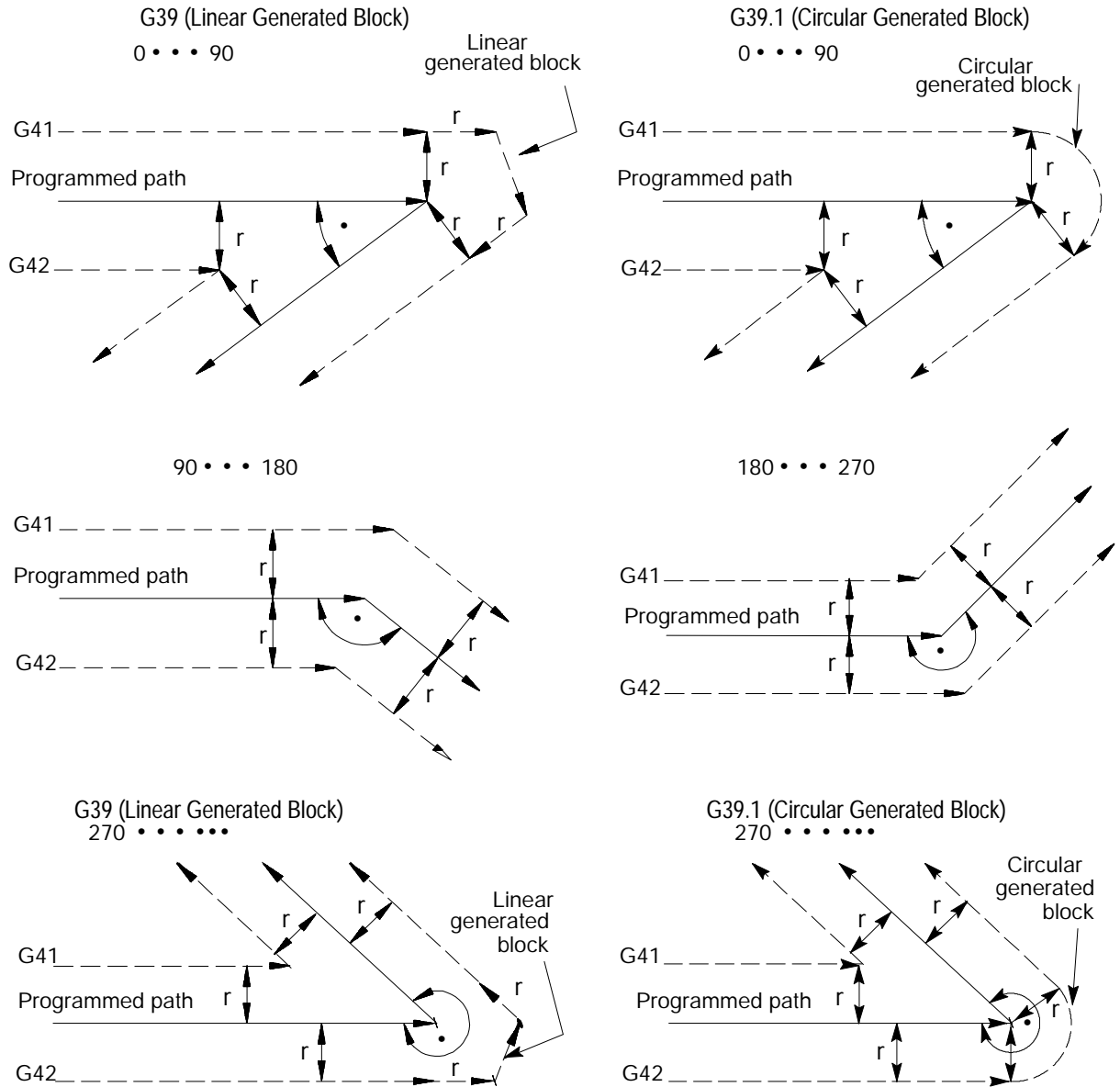


Figure 21.27
Cutter Compensation Tool Paths Straight Line-to-Arc

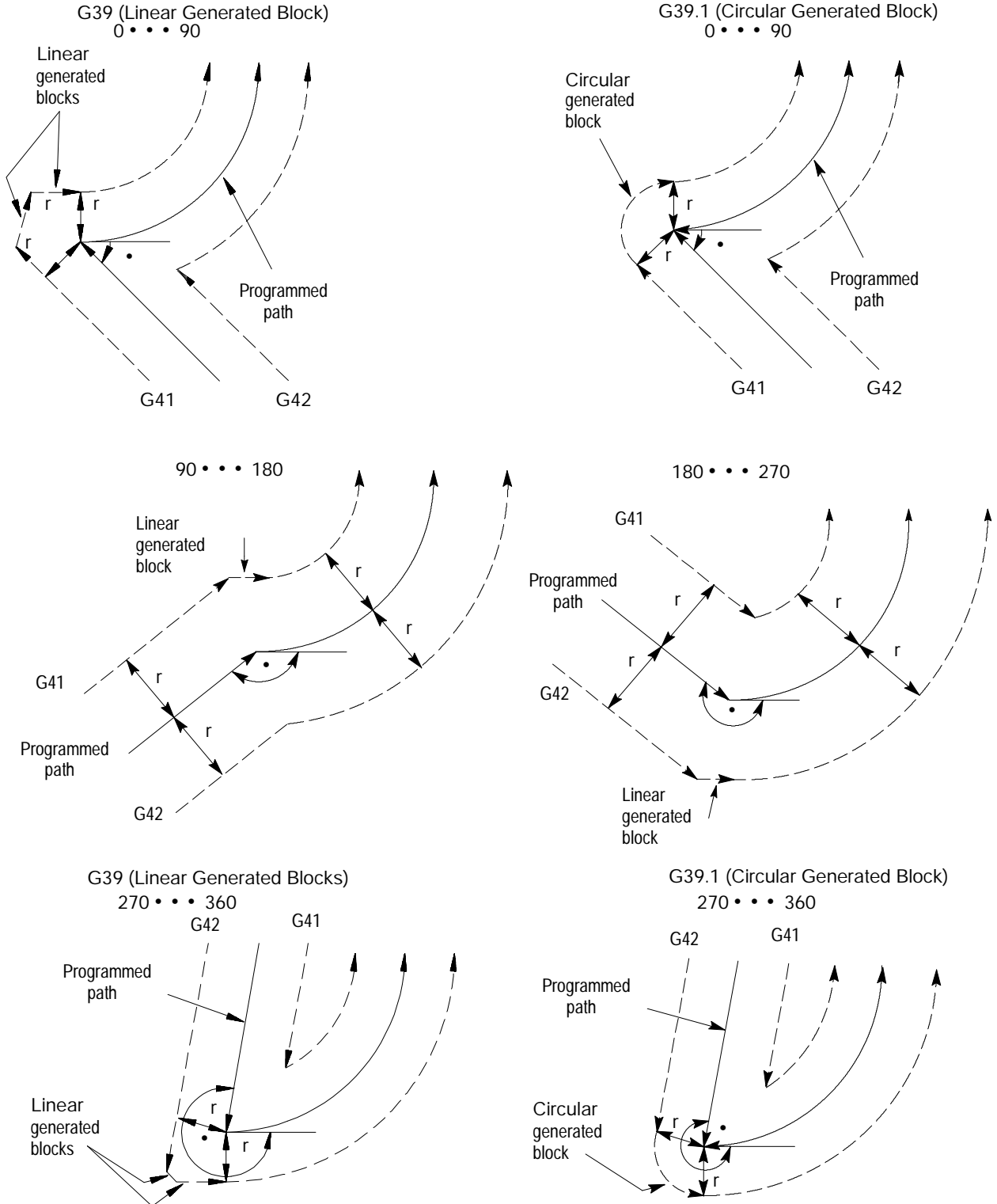


Figure 21.28
Cutter Compensation Tool Paths Arc-to-Straight Line

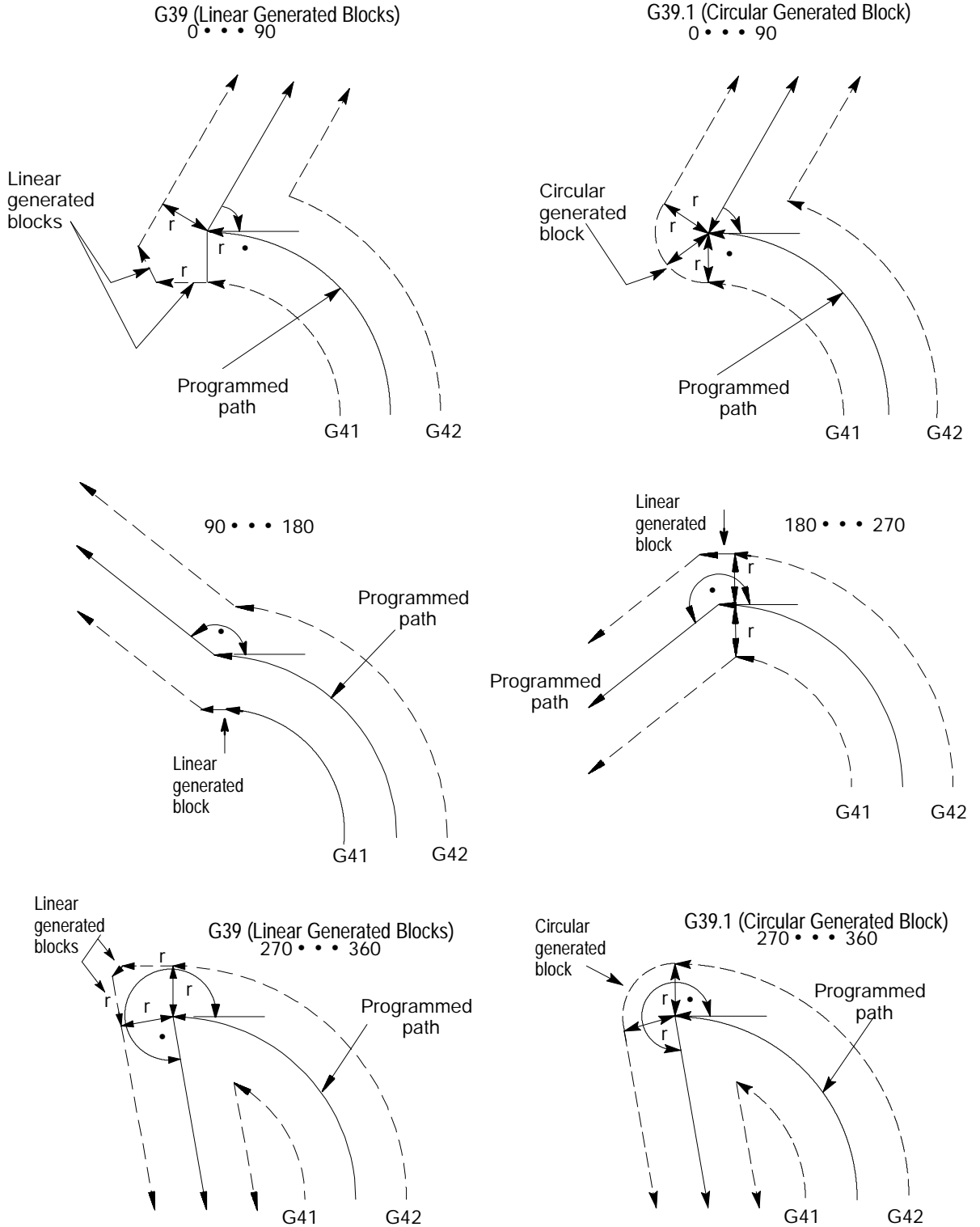
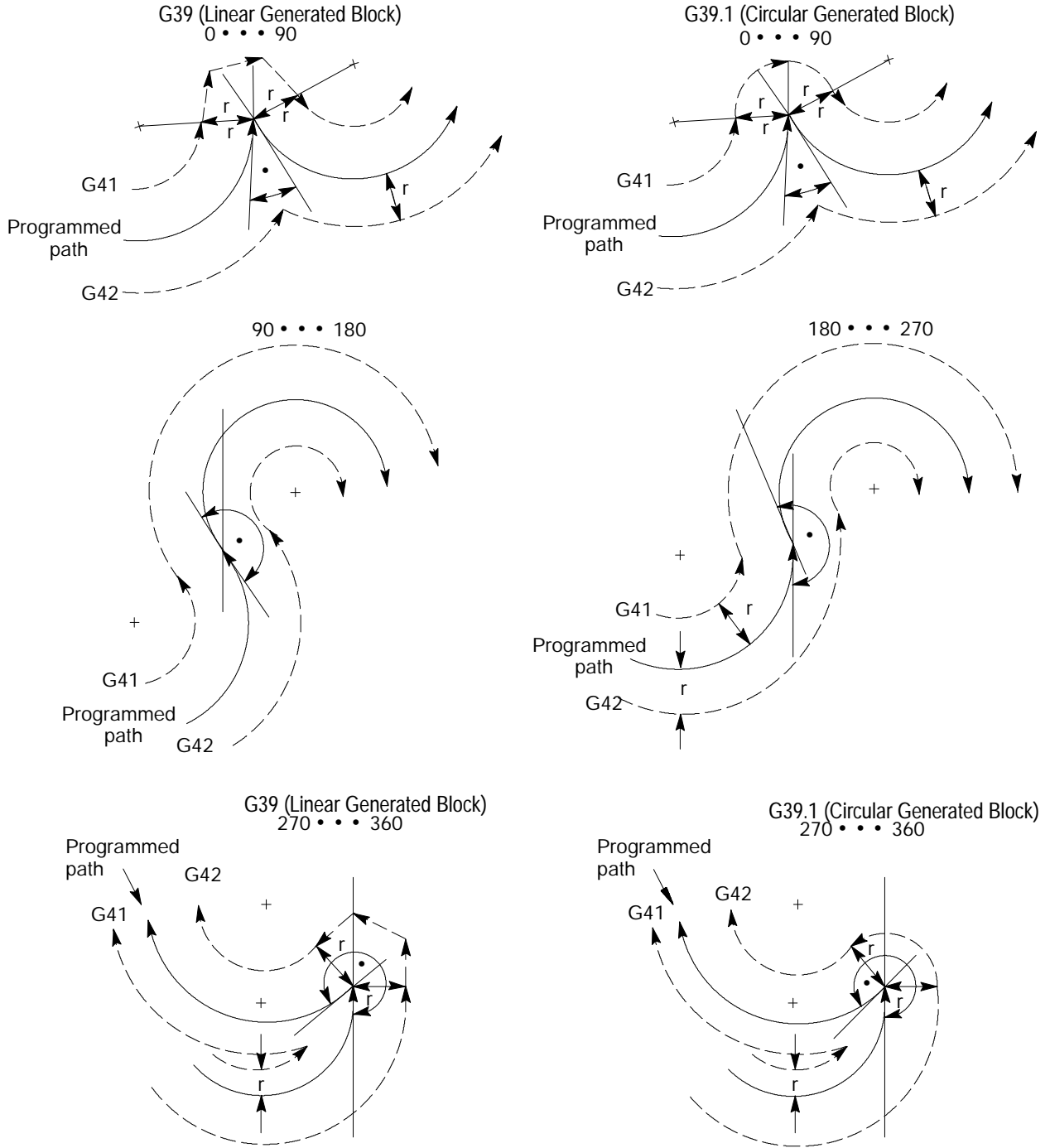


Figure 21.29
Cutter Compensation Tool Paths Arc-to-Arc



21.6 Cutter Compensation Special Cases

The following subsections describe possible tool paths that may be generated when programming one of the following during cutter compensation:

- changing cutter compensation direction (cross-over tool paths)
- exceeding the allowable number of consecutive non-motion blocks during cutter compensation
- corner movement following a generated block
- changing cutter radius during cutter compensation
- effect on cutter compensation when interrupting a program to execute either a MDI program or a manual move
- changing or offsetting current work coordinate system during cutter compensation
- moving to and from machine home and secondary machine home

21.6.1 Changing Cutter Compensation Direction

This section describes the resulting tool path when a change in compensation direction (left or right) is programmed. This may result in the cutting tool crossing over the programmed tool path as compensation changes from left to right or right to left.

Linear Tool Path-to-Linear Tool Path

The following figures show the tool path taken when cutter compensation is changed from G41 to G42 during the execution of two linear program moves.

The control generates two points when changing cutter compensation direction, called point 1 and point 2. Point 1 is the final tool position before compensation direction is changed (at right angles to the end-point of the programmed tool path offset by one tool radius). Point 2 is the desired tool position for the start of the first block using the changed compensation direction (at right angles to the start-point of the motion block that changes compensation direction and offset by the tool radius). The control generates the motion block that connects point 1 to point 2 as shown in Figure 21.30 through Figure 21.33.

Figure 21.30
Linear-to-Linear Change with Block Direction Reversed

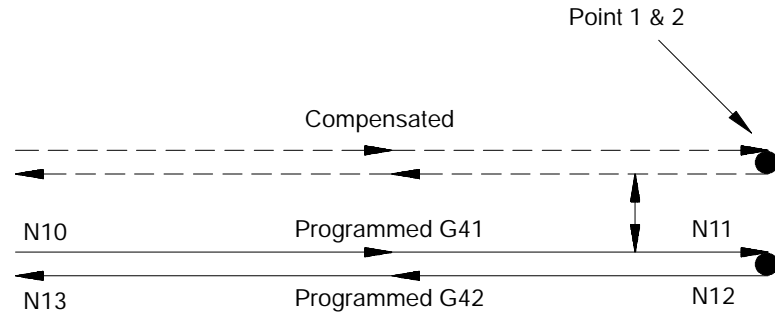


Figure 21.31
Linear-to-Linear Change with Tangential Motion Blocks

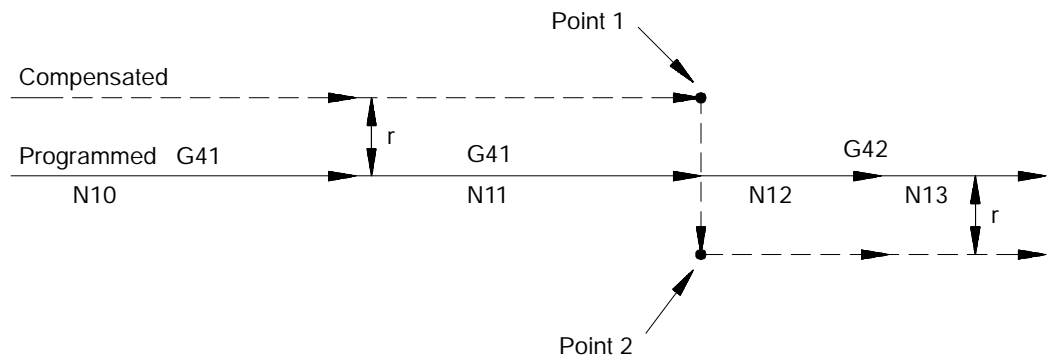


Figure 21.32
Linear-to-Linear Change with A Generated Block

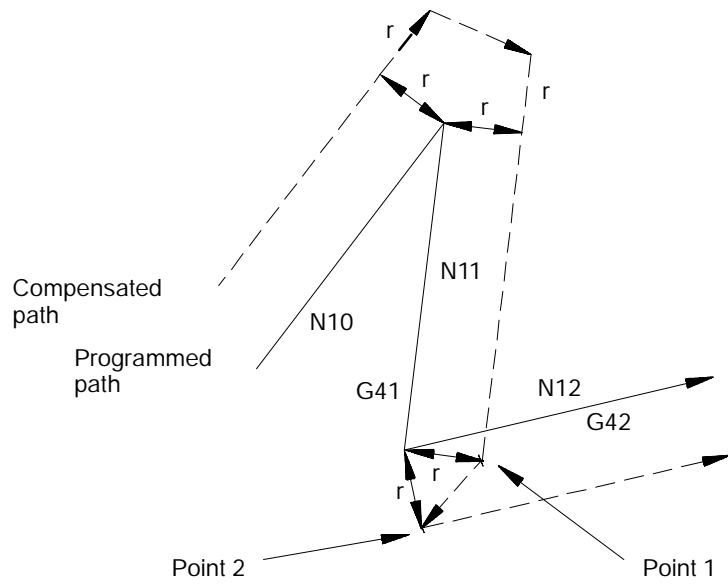
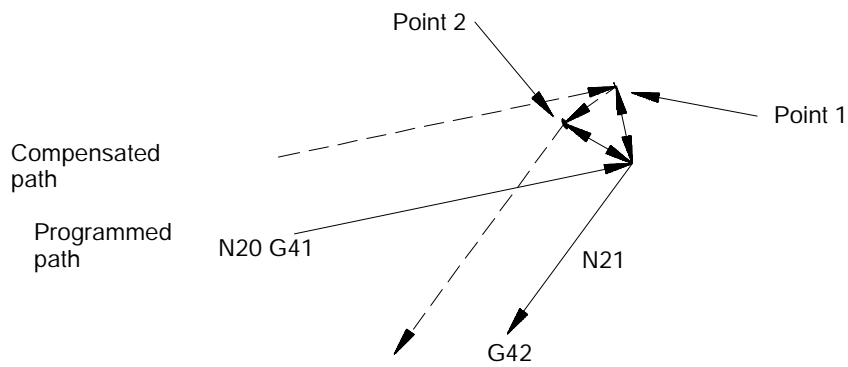


Figure 21.33
Linear-to-Linear Change with No Generated Block

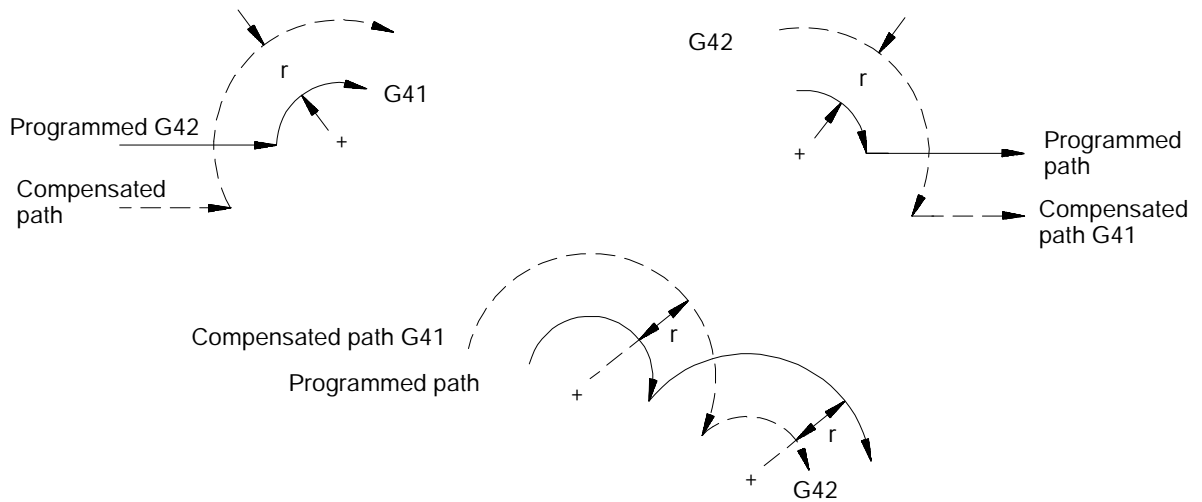


For one of the following cases that changes the cutter compensation direction, the control will attempt to find an intersection of the actual compensated tool paths:

Linear-to-Circular, Circular-to-Linear, or Circular-to-Circular Tool Paths

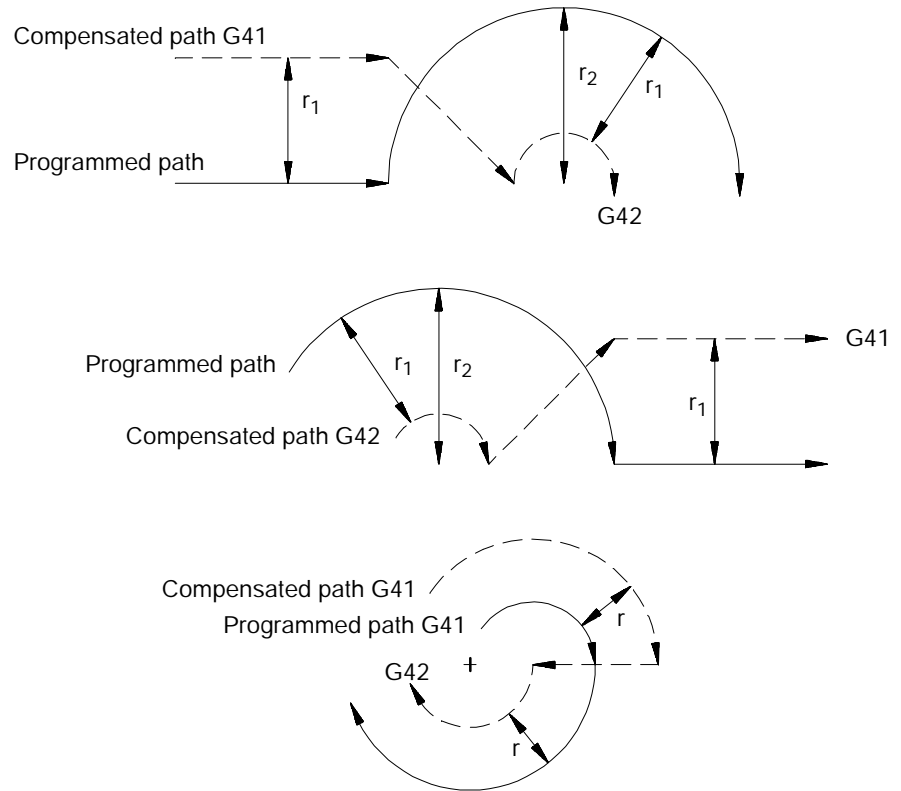
For the following cases that change the cutter compensation direction, the control attempts to find an intersection of the actual compensated tool paths. If the control finds an intersection, it modifies the end-point of the original compensated tool path and the start-point of the new compensated tool path to equal that intersection. (see Figure 21.34).

Figure 21.34
 Change in Compensation With Actual Tool Path Intersection



If no intersections of the actual tool paths exist, the compensated tool path is as shown in Figure 21.35. The end-point of the last move in the original compensation direction is at right angles to that block's programmed tool path, and the start-point of the move in the new compensation direction is at right angles to that block's start-point.

Figure 21.35
Change in Compensation with No Possible Tool Path Intersections



21.6.2 Too Many Non-Motion Blocks

The control is always looking ahead to the next motion block to determine the actual tool path taken for a motion block in cutter compensation. If the next block is not a motion block, the control continues to scan ahead for a motion block until it either detects one or the allowable number of non-motion blocks, as set in AMP, has been exceeded. Refer to documentation prepared by the system installer for the allowable number of non-motion blocks allowed in a specific system.

Important: The definition of a non-motion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in section 7.1.1 are also counted as a non-motion block in cutter compensation regardless of the content of the skipped block.

If the control, when scanning ahead, does not find a motion block before the number of non-motion blocks has been exceeded, it will not generate the normal cutter compensation move. Instead the control sets up the compensation move with an end-point one tool radius away from and at right angles to the programmed end-point.

In many cases, this may cause unwanted over-cutting of a work piece. Figure 21.36 and Figure 21.37 are example tool paths of programmed motion blocks followed by too many non-motion blocks before the next move was made:

Figure 21.36
 Too Many Non-Motion Blocks Following a Linear Move

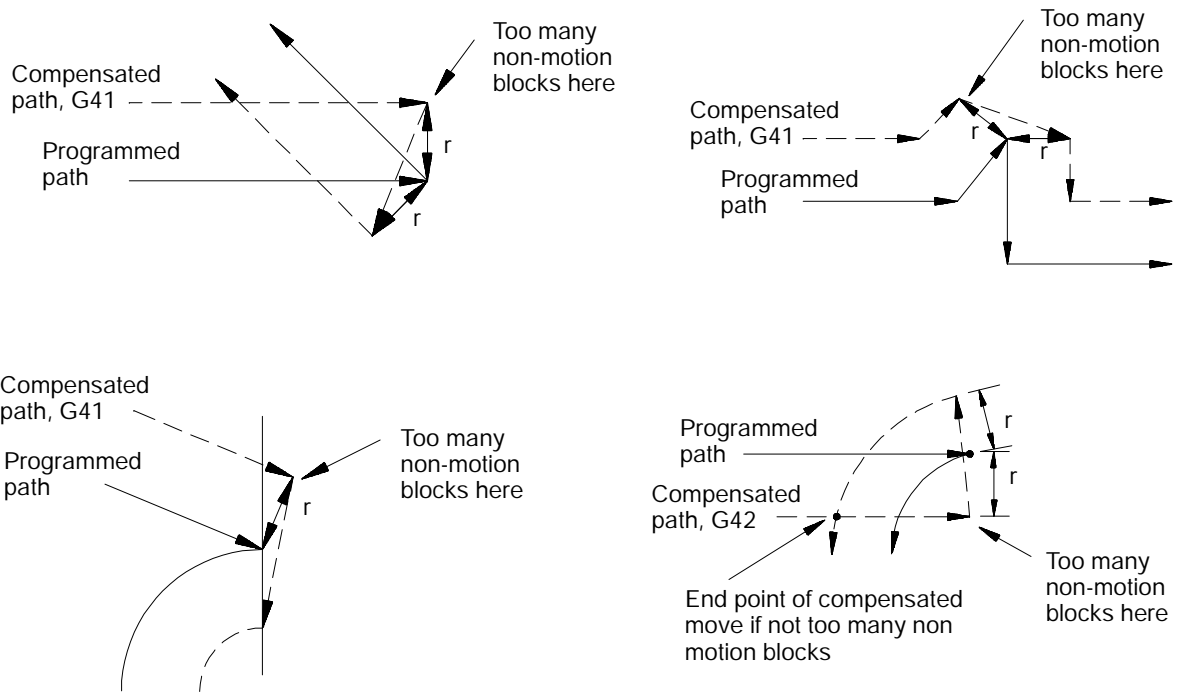
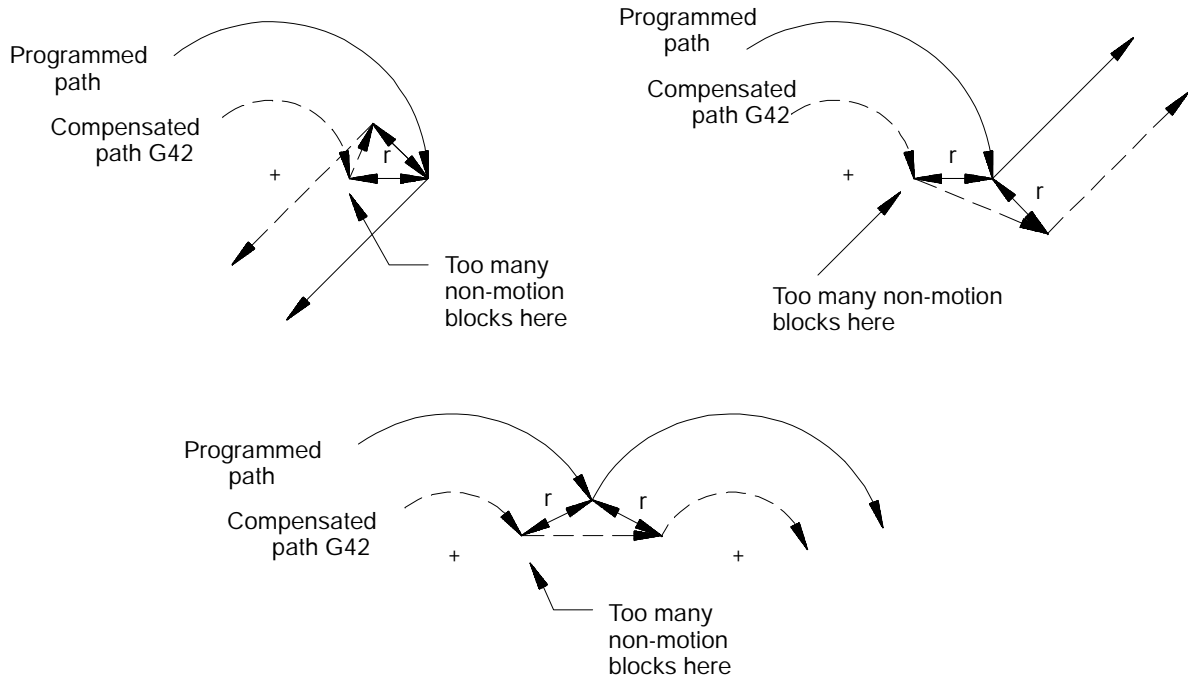


Figure 21.37
Too Many Non-Motion Blocks Following a Circular Move

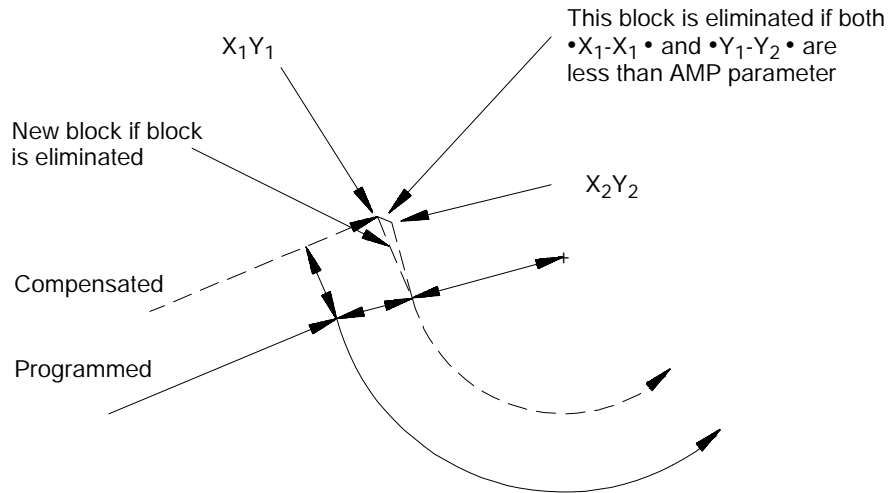


21.6.3 Corner Movement After Generated Blocks

Frequently it is necessary for the control to generate motion blocks to position the cutting tool in the proper alignment for a following compensated cutting move. These blocks are generated to make sure that the cutting tool remains at least one radius of the cutting tool away from the programmed cutting path at all times.

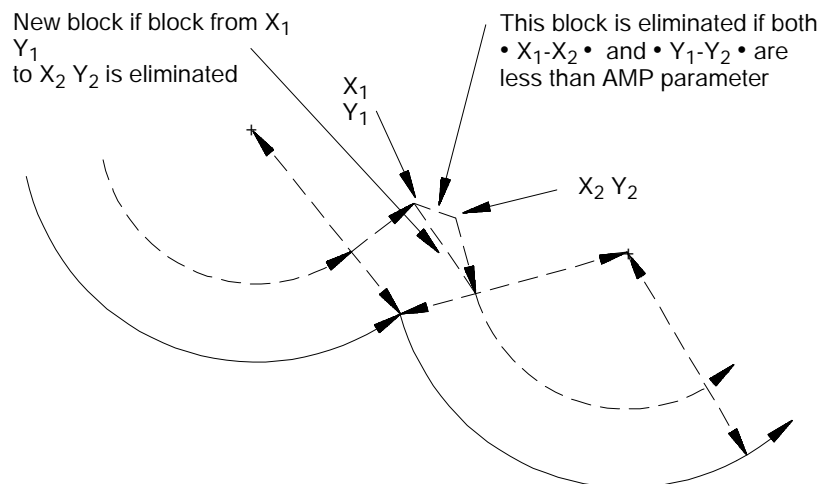
When the control generates two motion blocks, the length of the first generated block is checked against a minimum allowable length as determined in AMP by the system installer. The coordinate values for the current axes in the compensation plane are compared to the minimum allowed value. If both are less than the allowed value, the control does not execute the first generated block. The path of the second generated block is then altered to position the cutting tool along a linear path to the original end-point of the second generated block. See Figure 21.38 for a pictorial representation.

Figure 21.38
Compensation Corner Movement for Two Generated Blocks



When the control generates three motion blocks, the length of the second generated block is checked against a minimum allowable length as determined in AMP by the system installer. The amount of motion of the second move on the two axes in the compensation plane is compared to the minimum allowed value for each axis. If both are less than the allowed value, then the control does not execute the second generated block. The path of the third generated block is then altered to position the cutting tool along a linear path to the original end-point of the third generated block. See Figure 21.39 for a pictorial representation.

Figure 21.39
Compensation Corner Movement for Three Generated Blocks



21.6.4 Changing Cutter Radius During Compensation

If a tool becomes excessively worn, broken, or if any other reason requires the changing of the programmed tool radius, the cutter compensation should be cancelled and reinitialized after the tool has been changed. The following section describes the resulting tool path if, for some reason, it is desirable to program a change in cutter radius during cutter compensation.

Important: Slight overcutting may occur during Cutter Compensation, depending on the programmed path at the point where the change in cutter radius was made. To avoid overcutting, we recommend that you use a Mid-Start Program until the point of tool breakage.

See section 3.2 on changing the active tool offset and section 21.1 on changing the programmed compensation diameter offset number.

Figure 21.40 through Figure 21.42 are representations of the resulting tool paths after the programming of a change in the radius of the cutting tool. Assume in these figures that the programmed change to the tool radius is entered in block N11 which also contains the motion as described in the figure. The tool path taken when changing tool radius is dependant on the move immediately before the change in radius was programmed, the move that the change in radius was programmed in, and whether any generated motion blocks were made between these tool paths.

Figure 21.40 describes the tool path when the programmed moves are linear-to-linear.

Figure 21.40
Linear-to-Linear Change in Cutter Radius During Compensation

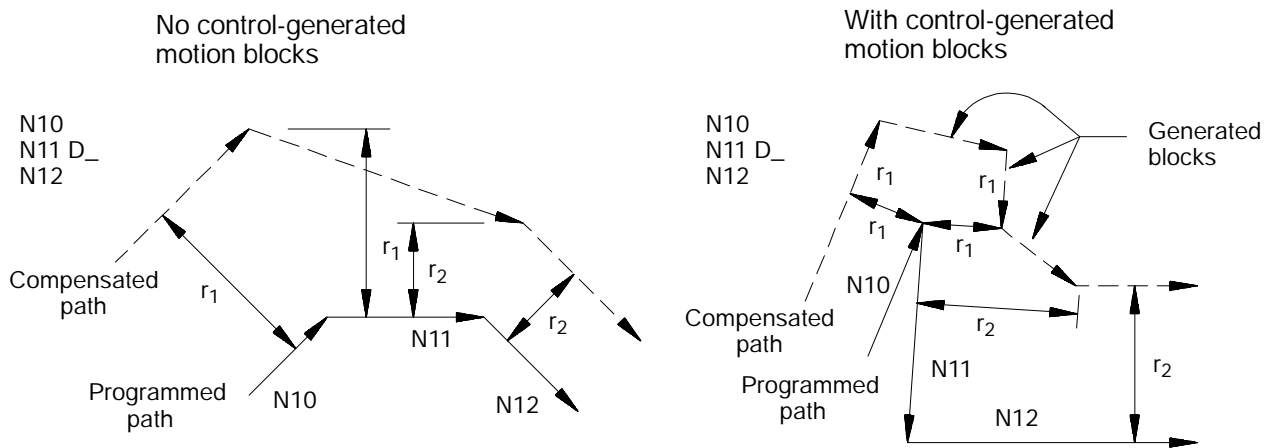


Figure 21.41 describes the tool path when the programmed moves are linear-to-circular.

Figure 21.41
 Linear-to-Circular Change in Cutter Radius During Compensation

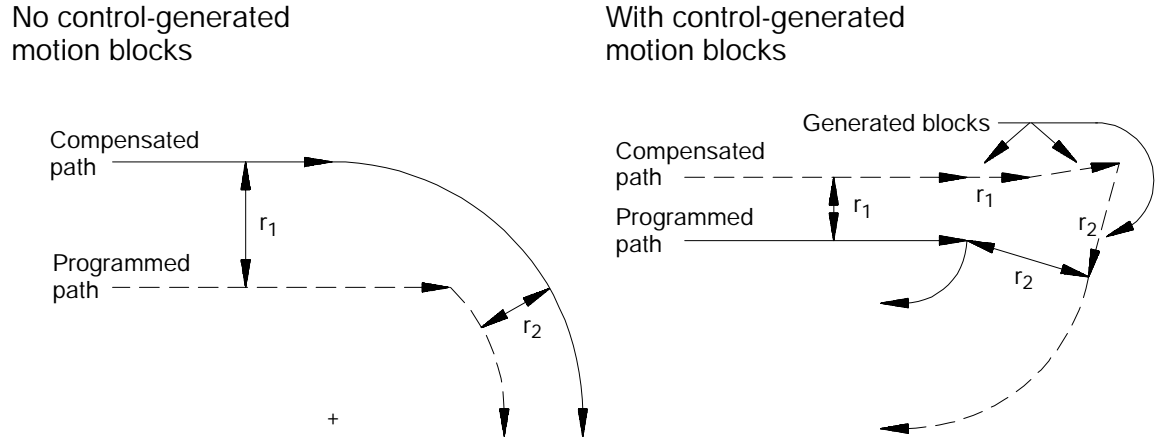
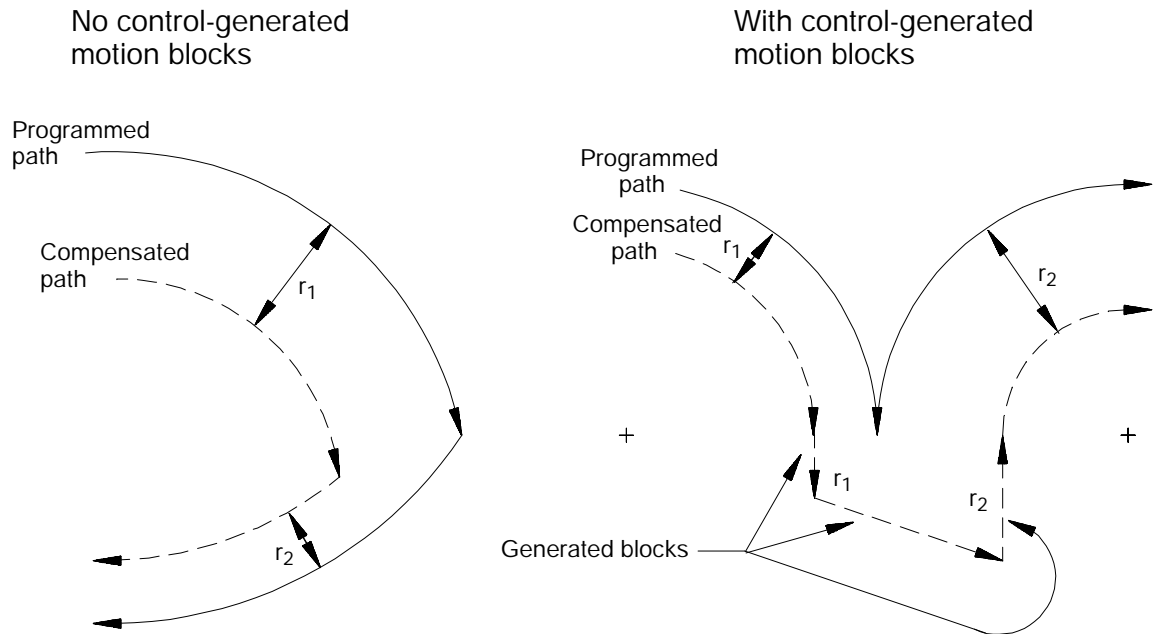


Figure 21.42 describes the tool path when the programmed moves are circular-to-circular.

Figure 21.42
 Circular-to-Circular Change in Cutter Radius During Compensation



Change in Cutter Radius During Jog Retract

This section describes a change in the cutter radius during a jog retract operation. This is a typical operation since the jog retract feature is often used when a tool becomes very worn or is broken. It may be necessary to replace the tool with a tool of a slightly different diameter. Cutter compensation is able to adjust to the new tool diameter.

Typically when the jog retract operation is performed, the tool is jogged away from the workpiece and then replaced. After it is replaced, it is necessary to activate a different tool diameter offset value. This is typically done in either of two methods:

- The new offset number is activated by programming a new D word in an MDI block.
- The new offset number is activated by using the {ACTIVE OFFSET} softkey found on the offset table screen. This feature is discussed in chapter 2.

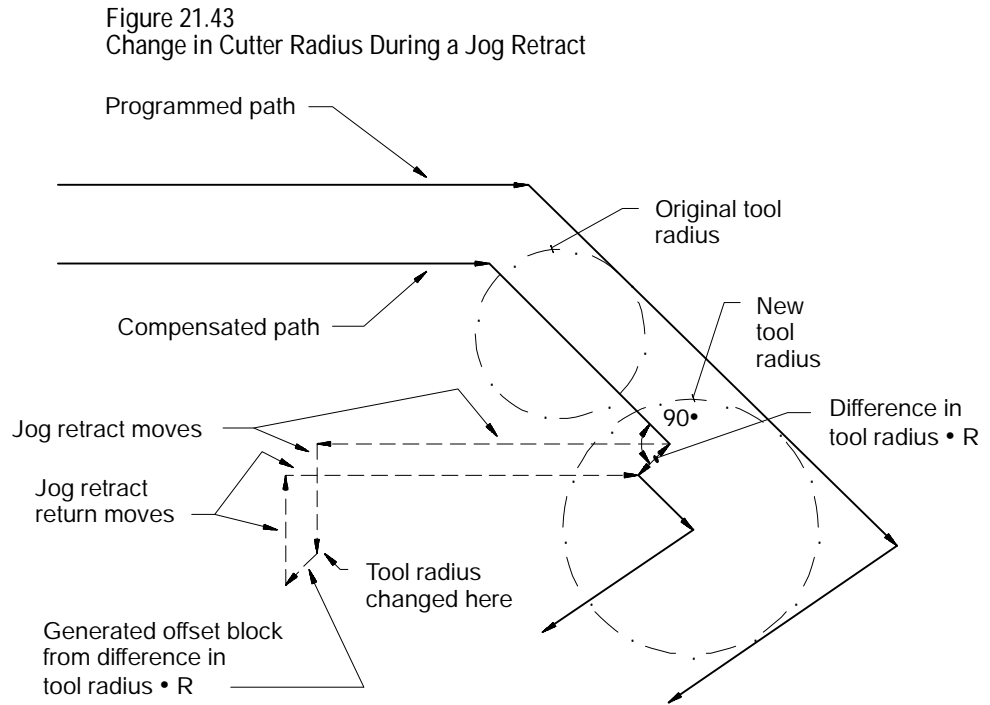
However the new offset is activated, cutter compensation is able to compensate for this new diameter by modifying the saved jogged path. This path is modified so that the new tool will cut the same part as the old tool. The absolute position of the machine, therefore, is different on the return path from what it was when jogging away from the part.

This jogged path is adjusted when the button is pressed to return from the jog retract. As soon as the <CYCLE START> button is pressed, the control generates a move that offsets the current tool position by the necessary distance. This distance is determined as being the necessary distance the tool would have to be positioned at so that the exact same jog return paths can be used to return to the part and yet have the end-point be offset from the original position by the difference in the cutter diameter.



CAUTION: Make sure that this offset path will not cause any collisions with the part or the machine fixtures. The position of the tool, when the tool change in jog retract is made, should be a safe distance from the part and machine fixtures.

Figure 21.43 gives an example of a typical change in tool radius during jog retract with cutter compensation active.



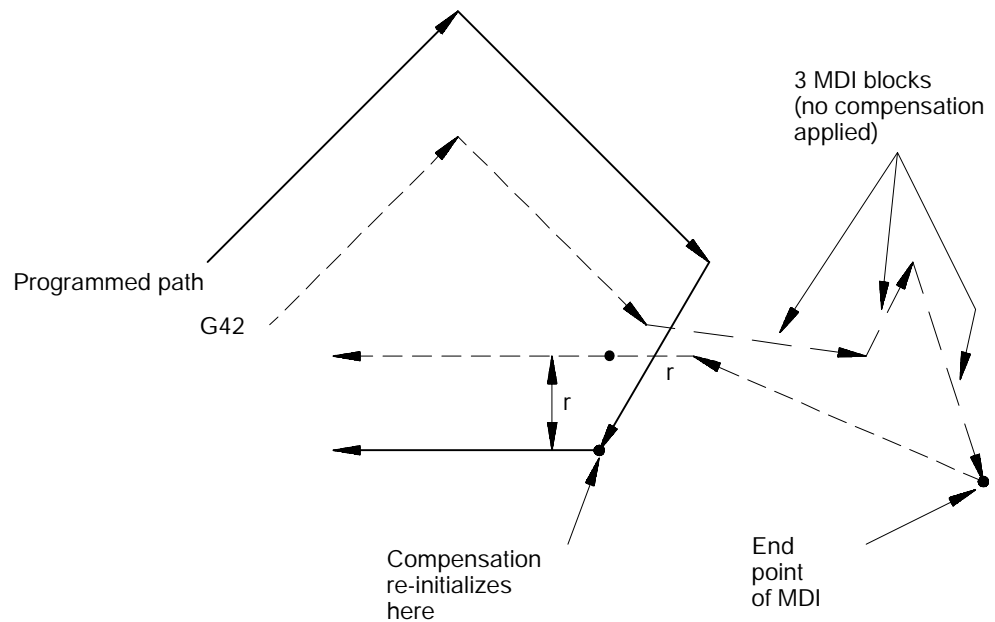
21.6.5 MDI or Manual Motion During Cutter Compensation

If exiting automatic mode and either a MDI motion block is executed or a manual jog motion is made, the cutter compensation feature, if active, re-initializes when the next motion block is executed in automatic mode. The compensation feature compensates the cutting tool one tool radius perpendicular to the tool path of the next motion block that is executed in automatic mode. In effect, the control generates its own entry move for compensation with the first compensated block being the next block executed in automatic operation.

Important: The cutter compensation feature is not available for any motion blocks that are programmed in MDI mode. The cutter compensation mode may be altered by programming either G41, G42, or G40; or the tool radius may be changed in an MDI program. However, none of the tool paths executed in MDI are compensated. No changes made to cutter compensation are applied until the next block executed in automatic mode.

Figure 21.44 is an example of the possible tool path that is taken when you interrupt an automatic operation during cutter compensation to execute MDI motion blocks. This same tool path applies also, if you interrupt cutter compensation to perform a manual jog move.

Figure 21.44
Cutter Compensation Interrupted with MDI Blocks



Important: If during cutter compensation, you switch out of automatic mode and either:

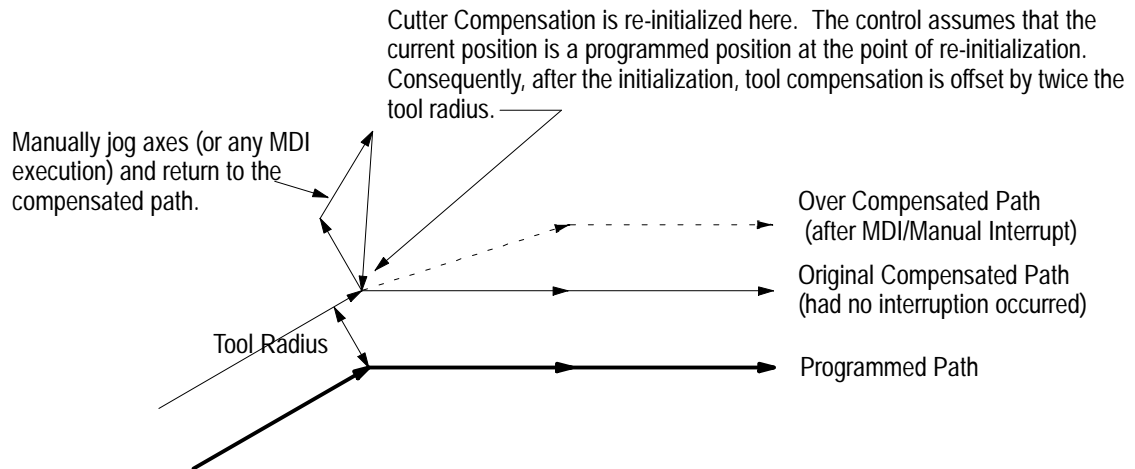
- generate axis motion in manual mode on an axis in the cutter compensation plane, or
- execute any block in MDI mode,

cutter compensation is re-initialize when you return to automatic mode.

This produces a path that is different from the path that would have been produced had the manual or MDI operation not been done, even if you returned the tool to the point of interrupt. In absolute mode the control returns to the originally compensated path after it executes a block that contains both axes in the compensation plane. In incremental mode, the compensated path remains offset by the additional tool radius.

Figure 21.45 illustrates these conditions.

Figure 21.45
Cutter Compensation Re-Initialized after a Manual or MDI Operation.



Use the Jog Retract feature if you must jog the axes away from a compensated path. Jog retract prevents the overcompensation from occurring.

If you interrupt cutter compensation with a manual or MDI operation and the next programmed block is a circular block, the control generates an error when it tries to re-initialize cutter compensation. You can avoid this by using the jog retract feature instead of manual or MDI when you need to interrupt cutter compensation.

21.6.6 Moving To/From Machine Home

It is recommended that cutter compensation be cancelled using a G40 command before the execution of a return to or from machine home, or a return to or from the secondary machine home. This refers to the operations performed when the control executes either the G28, G29, or G30 commands as described in section 21.3.

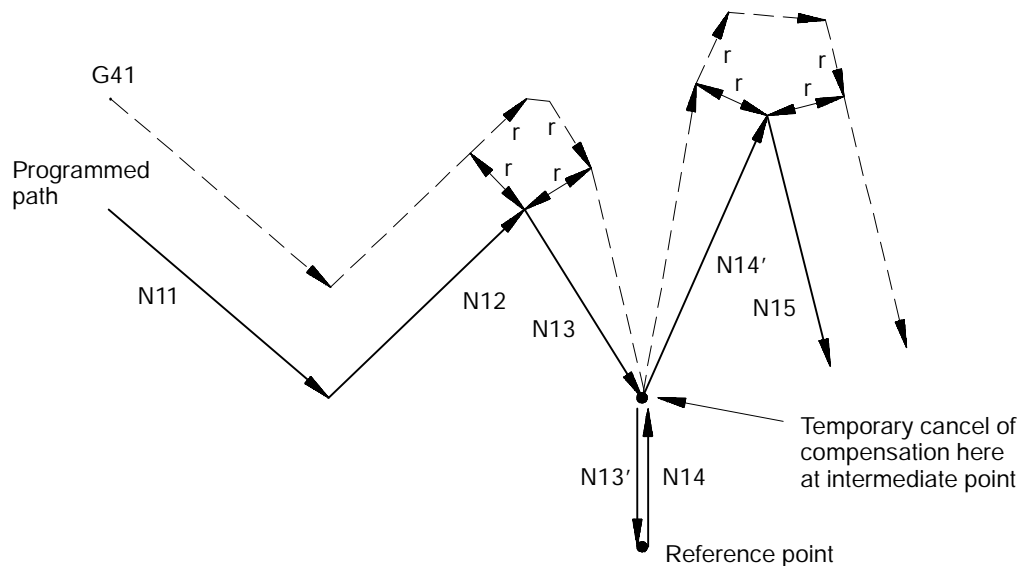
If compensation is not cancelled using a G40 command, the control automatically, temporarily cancels compensation for the return to machine home or secondary machine home operations. This is done by using the move to the intermediate point, designated when the operation is performed, as an exit move for compensation.

Important: An intermediate point should always be programmed for a return-to-home operation if cutter compensation is active. If no intermediate point is specified, the control executes the move prior to the return-to-home operation as an exit move. This may cause undesired over-cutting of the part.

If compensation was not cancelled using a G40 command before returning to machine or secondary home points, the control automatically re-initializes cutter compensation for the return from machine or secondary home points. This is done by using the move to the intermediate point, designated when the operation is performed, as an entry move for compensation.

Figure 21.46 gives an example of either a G28 or G30 block followed by a G29 block.

Figure 21.46
Cutter Compensation During G28, G30, and G29 Blocks



21.6.7 Changing or Offsetting Work Coordinate System

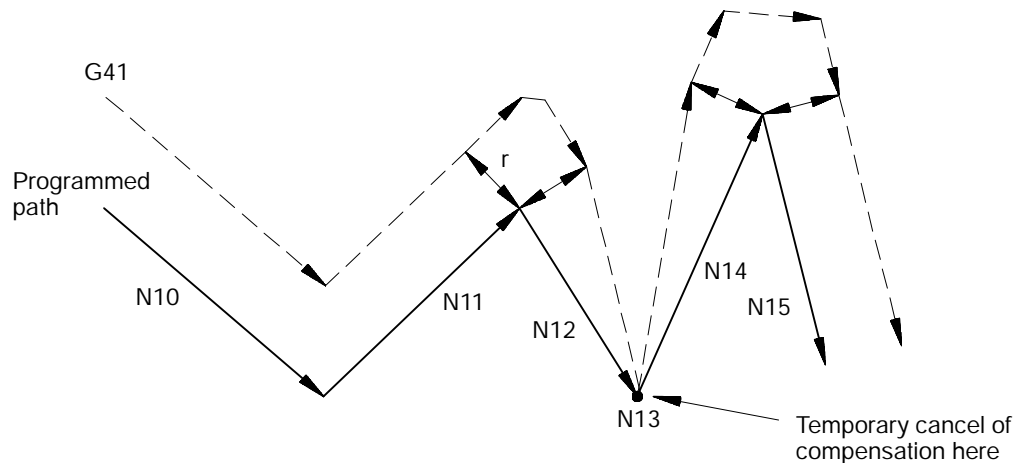
It is recommended that cutter compensation be cancelled using a G40 command before any modifications to the current work coordinate system are made including any offsets or any change of the coordinate system (G54-G59.3).

Important: Changes can only be made, to axes in the cutter compensation plane, in the executing part program or using MDI. Changes to axes not in the current cutter compensation plane can be made manually in the offset tables (using the softkeys). Changes to work coordinate system offsets for axes that are in the active cutter compensation plane can not be performed manually in the offset tables.

If compensation is not cancelled using a G40 command, the control automatically, temporarily cancels compensation for the change in work coordinate system. This is done by using the last compensated move in the current coordinate system as an exit move for compensation. The control then automatically re-initializes cutter compensation after the new work coordinate system is established. This is done by using the first move in the new coordinate system that is in the compensation plane as an entry move for compensation. This re-setup occurs even if the change to the coordinate system is not in the active cutter compensation plane.

Figure 21.47 gives an example of programming a G92 offset to the work coordinate system. The same figure, however, would apply to any change in the work coordinate system.

Figure 21.47
Cutter Compensation During G92 Offset to Work Coordinate System



21.6.8 Block Look-Ahead

During normal program execution, the control is constantly scanning blocks ahead to set up the necessary motions to correctly execute the current block. This is called Block Look-Ahead.

The control has a total of 21 set-up buffers. Different features require the use of some of these set-up buffers. One is always used for the currently executing block. Cutter compensation requires at least 3 of these buffers; other features also occasionally require the use of some of these buffers. Any remaining set-up buffers are used for Block Look-Ahead, one buffer for each block look-ahead.

At times (especially possible during cutter compensation) the control may not have enough look-ahead blocks to correctly execute the current block. When this happens, the control automatically starts disabling the block retrace feature. The block retrace feature uses one set-up buffer for every retraceable block. The number of re-traceable blocks is set in AMP by the system installer (a maximum of 15 is possible). As the control starts disabling the block-retrace feature, it decreases the number of available retraceable blocks until either there are sufficient set-up buffers available to successfully execute the current program, or until there are no more block-retrace blocks left. The control displays a message on line 2 of the CRT if it needs to disable some of the block-retrace feature.

Note that using too many buffers for block-retrace is not a recommended method of operating the control. The larger the number of look ahead blocks that the control is scanning, the more efficiently the control executes programs. It is recommended that the number of retrace blocks that is available to the block-retrace feature be limited to the minimum number required on the system (set the number of block-retrace blocks as low as possible for a specific application). This greatly improves program execution and prevent the necessity of the control deleting block retrace blocks.

21.7 Error Detection for Cutter Compensation

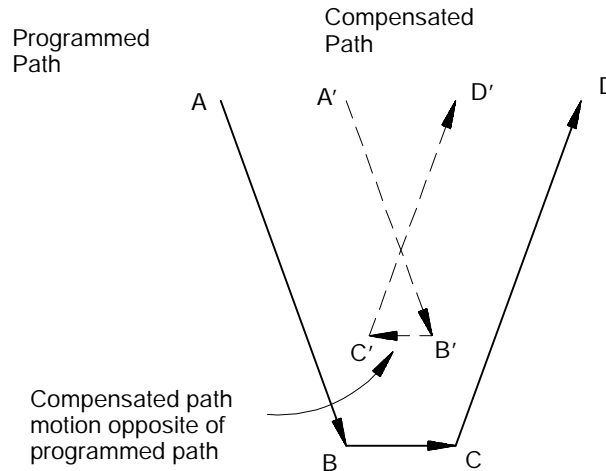
Error detection for cutter compensation blocks can be separated into three categories:

- Backwards motion detection
- Circular departure too small
- Interference

Backwards Motion Detection

The compensated tool path is parallel to, but in the opposite direction of, the programmed tool path.

Figure 21.48
Typical Backwards Motion Error

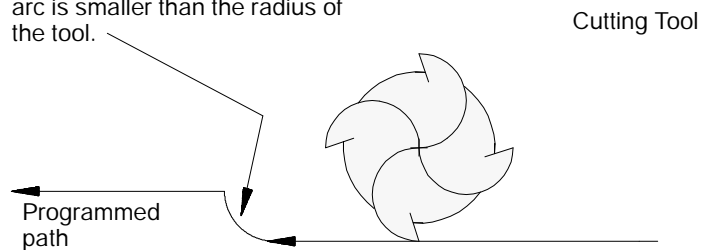


Circular Departure Too Small

This error is generated when the cutter radius is larger than the radius of the programmed arc. Note this form of compensation error cannot be disabled with an M-code. Programming this contour with tool tip radius compensation on always generates an error.

Figure 21.49
Typical Circular Departure Error

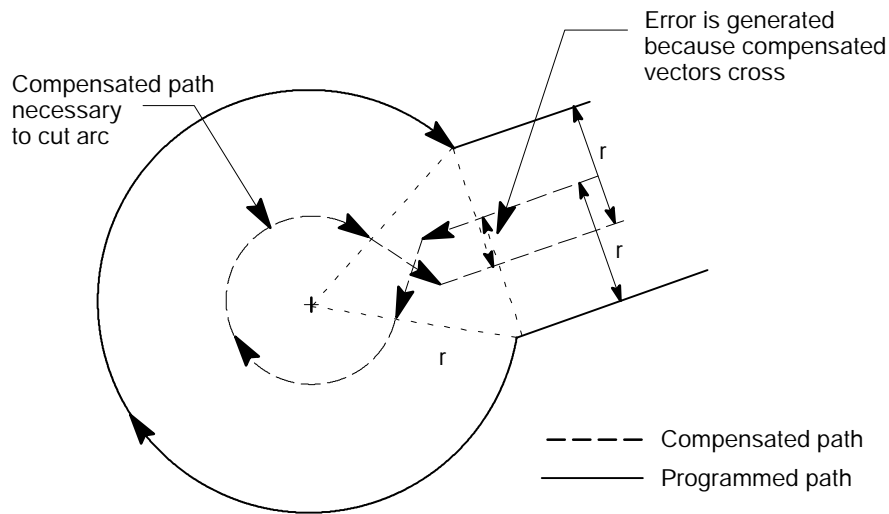
This arc cannot be cut because the radius of the programmed arc is smaller than the radius of the tool.



Interference

This error occurs when compensation vectors intersect. Normally when this intersection occurs, a backwards motion error is generated; however, a few special cases exist that are caught only by interference error detection.

Figure 21.50
Typical Interference Error



Disabling Error Detection

If so desired, all of the above error detection (with the exception of circular departure-too-small cases) can be disabled for a specific block or portion of a part program. To disable the error detection for a specific block, the system installer must have defined an M-code in AMP. By programming this M-code in a block, all error detection for cutter compensation may be disabled. Error detection is disabled until another M-code defined in AMP to re-enable error detection is programmed in a block.

Important: Circular departure too small cases cannot be disabled. The control cannot execute a compensated path when this error occurs.

The default condition is error detection enabled. Default values for these M-codes are:

M-code:	Error detection:
M800	disables
M801	enables

Error detection M-codes are only functional when cutter compensation is active. Cutter compensation is active when the control is in G41 or G42 mode and has already made the entry move into compensation. If an M800 or M801 is programmed in G40 mode or before the entry move into cutter compensation takes place, the M code is ignored.

If error detection is disabled in cutter compensation, and cutter compensation is exited (G40 programmed), the next time cutter compensation is reactivated error detection will be reactivated automatically. Error detection is always automatically enabled when cutter compensation is activated.

Refer to documentation prepared by your system installer for the M-codes used on your specific system.

END OF CHAPTER

Using Pocket Milling Cycles

22.0 Chapter Overview

Use pocket milling cycles to cut circular, rectangular, hemispherical pockets and posts, or irregular pockets and posts. Pocket milling cycles are cycles that make multiple passes along the X, Y, and Z axes to cut out a pocket in a workpiece. There are 8 pocket milling cycles. These include:

- five G88.1 Pocket Milling Roughing Cycles
- three G88.2 Pocket Milling Finishing Cycles

Important: You must turn cutter compensation off before executing any of these pocket cycles. An error “Illegal G Code During G41/G42” is displayed if cutter compensation is on when the control executes one of these pocket cycles.

Important: The Irregular Pocket Milling Cycles feature (G89.1 and G89.2) is only available prior to system software release 12.xx. Any attempt to program a G89.1 or G89.2 in release 12.xx or later will result in the error message, “Illegal G-code”.

22.1 Pocket Milling Roughing Cycle (G88.1)

Use the G88.1 pocket milling roughing cycle to rough out rectangular or circular pockets, slots, and to enlarge an existing rectangular or circular pocket.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane. If the current plane is not the XY plane, the operation of these cycles will rotate accordingly.

Important: Tool length and diameter offsets must be entered and active prior to the G88 block.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.1 roughing cycle for each of the possible pockets.

22.1.1 Rectangular Pocket Roughing Using G88.1

Use the G88.1 pocket milling roughing cycle to rough out a rectangular pocket in a workpiece. This cycle makes multiple rectangular cuts at a programmed width and depth.

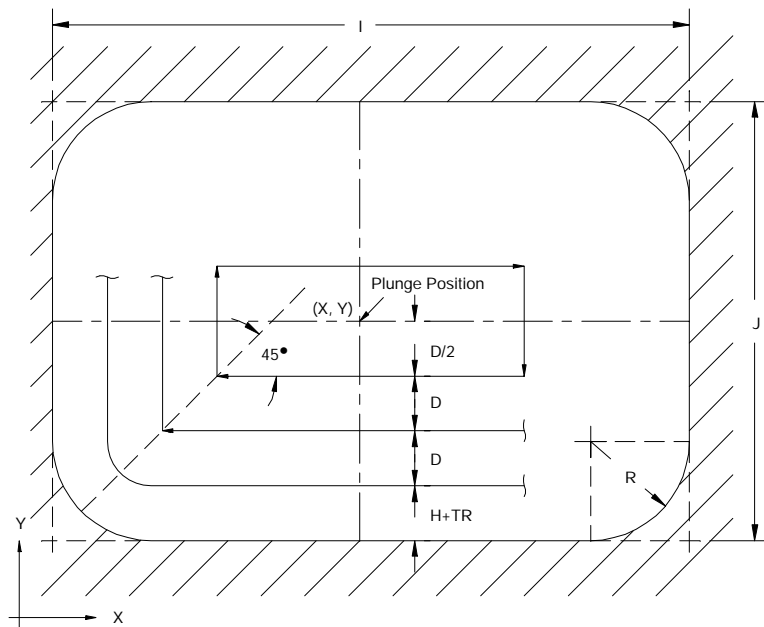
The G88.1 block used to rough out a rectangular pocket has this format:

G88.1 X_Y_Z_I_J_(, R or, C) _P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the rectangular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the rectangular pocket's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. Important: It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively. Important: When roughing out a rectangular pocket, the tool diameter can not exceed the length of the shortest side of the rectangular pocket. If it does, the control enters Cycle-Stop mode and displays the error message "TOOL RADIUS TOO LARGE" on the CRT.
, R	Corner radius. This is an optional parameter that is used to program rounded interior corners in the rectangular pocket.
, C	Corner chamfer. This is an optional parameter that is used to program chamfered interior corners in the rectangular pocket. Important: In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.
P	Direction of roughing cut. This parameter determines whether the roughing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. Important: Cutter compensation (G41/G42) must be disabled prior to the G88.1 block. The control generates an error if compensation is not disabled.
H	The finish allowance that will be left on the sides of the pocket. This finish allowance can be removed later using a G88.2 finishing cycle. To leave a finish allowance on the pocket bottom, program a pocket depth (Z parameter) that is at the desired finish allowance above the actual pocket bottom. This finish allowance can be removed later using a G88.2 finishing cycle.
D	Roughing cut thickness. This parameter specifies the maximum width of any XY axis roughing cuts. This is an optional parameter. If not programmed, the control uses the default thickness, which is equal to half of the current tool diameter. Important: The roughing cut thickness can not be greater than the current tool diameter. If it is, the control will enter Cycle-Stop mode and display the error message "D-WORD LARGER THAN TOOL DIAMETER" on the CRT.
L	Incremental plunge depth of each cutting pass along the Z axis. If L is not programmed, the plunge amount will be equal to the programmed depth of the pocket. This is an optional parameter.

Where :	Is :
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed, the roughing feedrate (F) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis moves. If not programmed, the existing (modal) feedrate will be used.
<p>Important: The rectangular pocket does not have to be parallel to the axes of the selected plane. It may be rotated by rotating the work coordinate system (G68). Refer to chapter 13 for additional information on rotating the work coordinate system.</p>	

Figure 22.1
Rectangular Pocket Roughing Using G88.1



Important: The tool should be positioned near the center of the pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the rectangular pocket is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate specified by the F parameter.

After the plunge operation, the control performs a single-axis roughing cut outward along either the -X or -Y axis. The roughing cut is performed along which ever axis intersects the longer side of the rectangular pocket. The control then moves the tool in a rectangular path, defined by the programmed sides of the rectangular pocket, that starts and ends at the same point on either the -X or the -Y axis.

After completing the rectangular path, the control makes another single-axis roughing cut outwards along the -X or -Y axis. The control then moves the tool in a rectangular path that starts and ends at the same point on -X or -Y axis.

This process is repeated until the side of the pocket, less the finish allowance H, is reached. The tool is then simultaneously raised to the clearance amount and moved to the plunge-position. This completes the machining of one L level.

The width of the first roughing cut is equal to the programmed rough cut thickness, D, divided by two then multiplied by 95% $((D/2) \times (.95))$. The width of the last roughing cut is equal to the tool radius plus the finish allowance $(H + TR)$.

The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

If ,R or ,C is not programmed in the G88.1 block, each corner of the rectangular pocket is squared off as much as the tool radius will allow. If ,R or ,C is programmed in the G88.1 block, the corners of the rectangular pocket will either be rounded or chamfered. Refer to chapter 16, Using Chamfers and Corner Radius, for additional information on chamfers and corner rounding.

22.1.2 Rectangular Pocket Enlarging Using G88.1

Use the G88.1 pocket milling roughing cycle to enlarge an existing rectangular pocket in a workpiece. This cycle makes multiple rectangular cuts at a programmed width and depth.

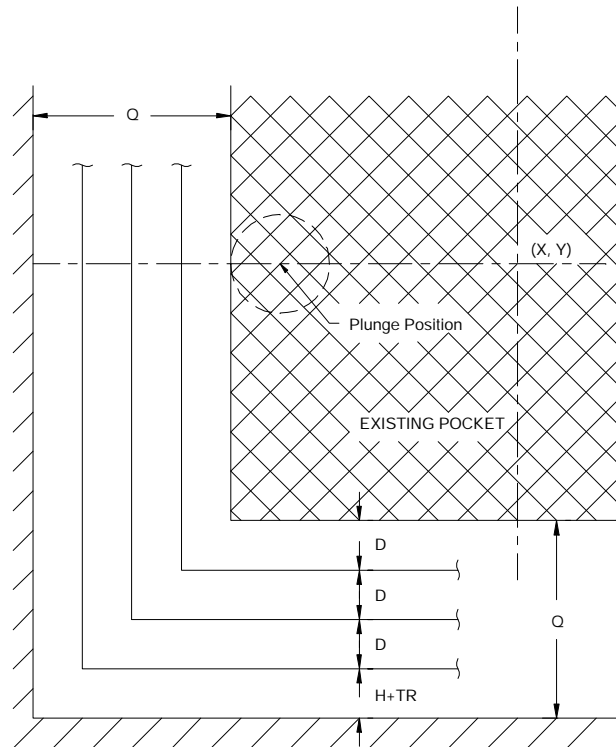
The G88.1 block used to enlarge an existing rectangular pocket has this format:

G88.1 X_Y_Z_I_J_Q_(,R or,C)_P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the original rectangular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket.
I J	Length of the original rectangular pocket's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. Important: It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
Q	Unsigned incremental value that specifies how much the original rectangular pocket should be enlarged. I and J are both enlarged by this amount.
, R	Corner radius.
, C	Corner chamfer.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Important: The rectangular pocket does not have to be parallel to the axes of the selected plane. It may be rotated by rotating the work coordinate system (G68). Refer to chapter 13 for additional information on rotating the work coordinate system.

Figure 22.2
Rectangular Pocket Enlarging Using G88.1



Important: The tool should be positioned near the center of the original pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to system installers literature) while moving it to the longest side of the original pocket. This move is always along the -X or -Y axis depending on whether the rectangular pocket is longer along the X or Y axis (shorter axis move is the first move). This position is the plunge- position, the position where the plunge to the programmed depth takes place.

Important: If the original pocket has rounded or chamfered corners, excess material may be in the corners that may have to be cleaned out before attempting to enlarge the original pocket, since the G88.1 cycle does not compensate for rounded or chamfered corners.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. The plunge takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate specified by the F parameter.

After the plunge operation, the control performs a single-axis cut outwards to a point on the longest side of the rectangular pocket. This move is always along the -X or -Y axis depending on whether the rectangular pocket is longer along the X or Y axis (shorter axis move is the first move). The control then moves the tool in a rectangular path that starts and ends at either the -X or -Y axis.

This process is repeated until the sides of the enlarged pocket, less the finish allowance H, are reached. The tool is then raised by the clearance amount and moved at rapid feedrate back to the plunge- position. This completes the machining of one L level.

The width of the last roughing cut is equal to the tool radius plus the finish allowance (H + TR). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

If ,R or ,C is not programmed in the G88.1 block, each corner of the rectangular pocket is squared off as much as the tool radius will allow. If ,R or ,C is programmed in the G88.1 block, the corners of the rectangular pocket will either be rounded or chamfered. Refer to chapter 16, Using Chamfers and Corner Radius, for additional information on chamfering and corner rounding.

22.1.3 Slot Roughing Using G88.1

Use the G88.1 pocket milling roughing cycle to rough out a slot in a workpiece. This cycle makes multiple cuts at a programmed length and depth.

The G88.1 block used to rough out a slot has this format:

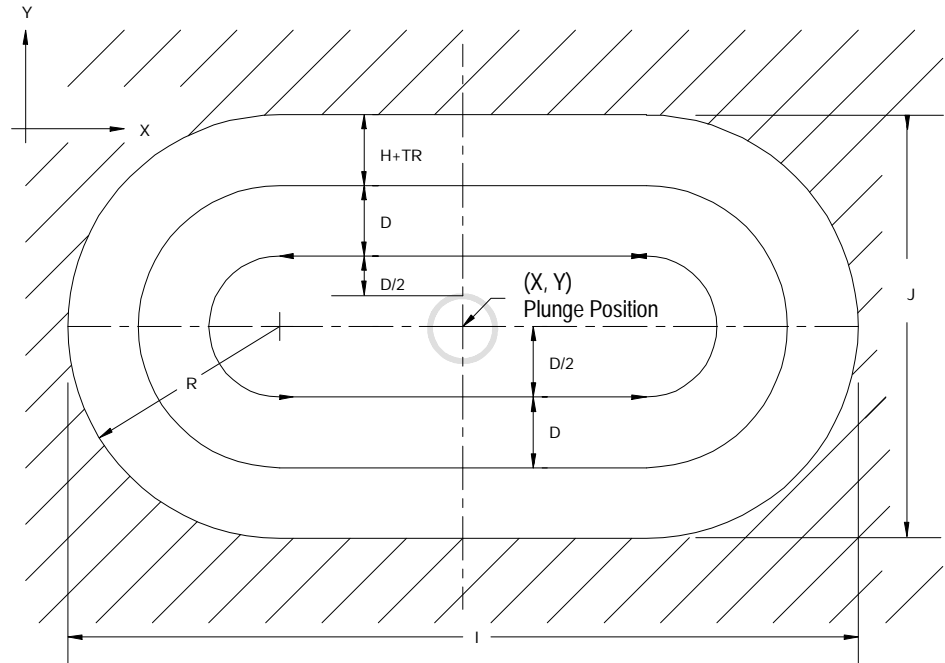
G88.1 X_Y_Z_I_R_P_H_D_L_E_F_; (X axis slot)

OR

G88.1 X_Y_Z_J_R_P_H_D_L_E_F_; (Y axis slot)

Where :	Is :
X Y	The coordinates that specify the center of the slot.
Z	The coordinate (along the plunging axis) that specifies the bottom of the slot.
I J	The length of the slot as measured from the points where the axis intersects the arc at each end of the slot. I specifies the length of a X axis slot. J specifies the length of a Y axis slot. These are unsigned incremental values. Important: It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
R	The radius of the arc at each end of the slot. The radius equals the slot-width/2. If not programmed, the control uses the tool radius value. Important: When roughing out a slot, the tool radius can not exceed the radius of the arc at each end of the slot. If it does, the control enters Cycle-Stop mode and displays the error message "TOOL RADIUS TOO LARGE" on the CRT.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 22.3
Slot Roughing Using G88.1



Important: The tool should be positioned at the center of the slot prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

If the programmed R parameter is greater than the tool radius, this cycle is processed similar to a G88.1 roughing cycle for a rectangular pocket. The difference being that the R parameter programmed in a slot roughing cycle specifies the radius of the arc at the end of the slot versus the radius of the corners in a rectangular roughing cycle. The control cuts an arc at the short side of the rectangular pocket based on the programmed R parameter and the arc-center determined by the control.

If the programmed R parameter is equal to the tool radius, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the arc-center at the -X or -Y end of the slot. This simultaneous move takes place at the rapid feedrate. This position is the plunge-position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. The plunge takes place at the plunge feedrate specified by the E parameter. After the plunge operation a roughing cut is made at the feedrate specified by the F parameter to the arc-center at the +X or +Y end of the slot.

A plunge to the next incremental L level or to the programmed Z level is made. A roughing cut is made at the feedrate F to the arc-center at the -X or -Y end of the slot. This process is repeated at each L level until the slot is machined out. When the slot is machined out the control raises the tool to the initial Z level plus the clearance amount and then moves it to the pre-cycle position of the tool.

22.1.4 Circular Pocket Roughing Using G88.1

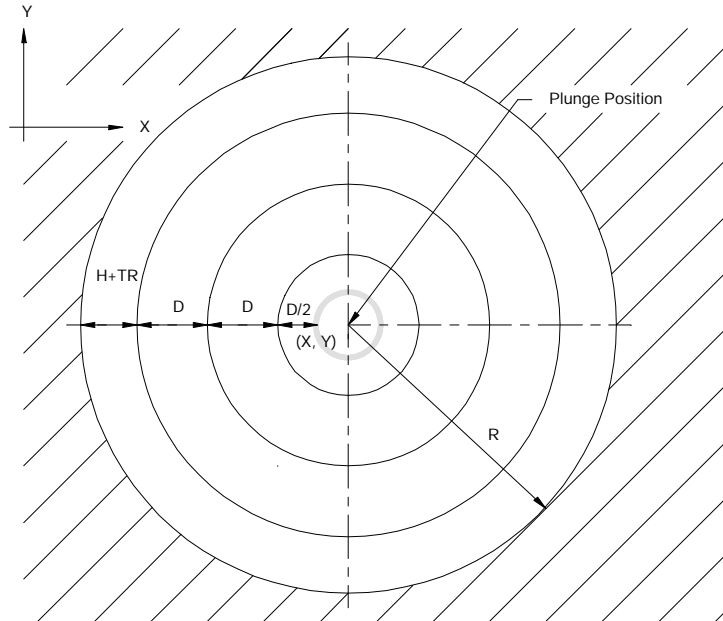
Use the G88.1 pocket milling roughing cycle to rough out a circular pocket in a workpiece. This cycle makes multiple circular cuts at a programmed width and depth.

The G88.1 block used to rough out a circular pocket has this format:

G88.1 X_Y_Z_R_P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the circular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the circular pocket. This parameter must be programmed.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 22.4
Circular Pocket Roughing Using G88.1



Important: The tool should be positioned near the center of the pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the circular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the pocket is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate.

After the plunge operation, the control performs a single-axis cut outwards along the -X axis of the circular pocket. The control then moves the tool in a 360 degree circular path, defined by the center of the circular pocket and its radius, around the plunge- position.

After completing the 360 degree circular path, the control makes a single-axis rough cut outwards along the -X axis then cuts another 360 degree circular path. This process is repeated until the sides of the pocket, less the finish allowance H, are reached. The tool is then simultaneously raised by the clearance amount and moved at rapid feedrate back to the plunge-position. This completes machining of one L level.

The width of the first roughing cut is equal to the programmed rough cut thickness, D, divided by two then multiplied by 95% $((D/2) \times (.95))$. The width of the last roughing cut is equal to the tool radius plus the finish allowance $(H + TR)$.

The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

22.1.5
Circular Pocket Enlarging
Using G88.1

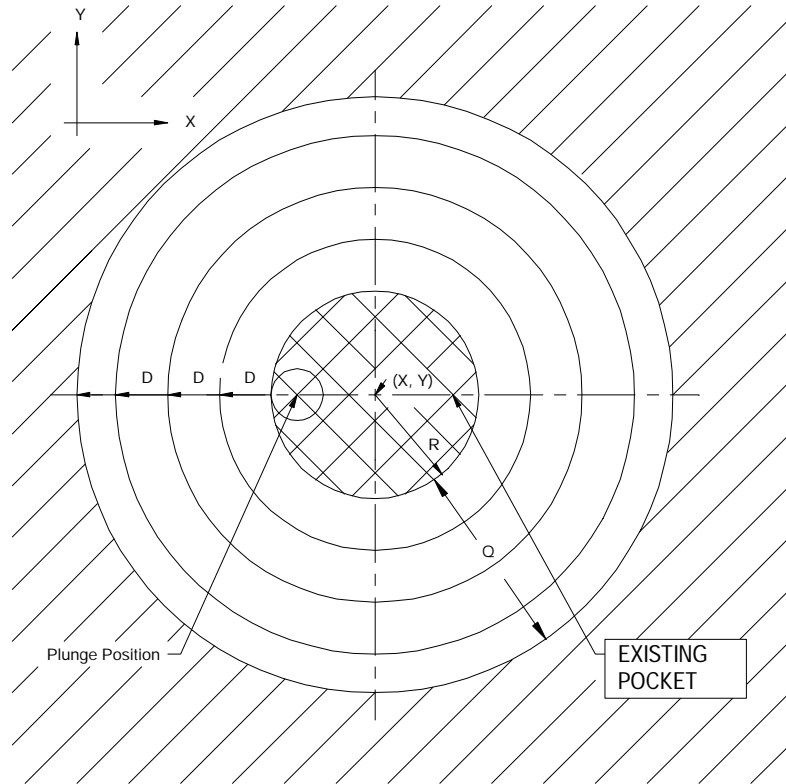
Use the G88.1 pocket milling roughing cycle to enlarge an existing circular pocket in a workpiece. This cycle makes multiple circular cuts at a programmed width and depth.

The G88.1 block used to enlarge an existing circular pocket has this format:

G88.1 X_Y_Z_R_Q_P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the original circular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the original circular pocket. This parameter must be programmed.
Q	Unsigned incremental value that specifies how much the original circular pocket should be enlarged. This parameter must be programmed. The radius of the enlarged circular pocket is equal to R + Q.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 22.5
Circular Pocket Enlarging Using G88.1



Important: The tool should be positioned near the center of the pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer the literature provided by your system installer) while moving it to the center of the original circular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The tool is then moved along the -X axis to the left side of the original circular pocket. This position is the plunge-position of the cycle.

After the plunge operation, the control performs a single-axis cut outwards along the -X axis of the circular pocket. The control then moves the tool in a 360 degree circular path that starts and ends at the -X axis.

After completing the 360 degree circular path, the control makes a single-axis rough cut outwards along the -X axis then cuts another 360 degree circular path. This process is repeated until the sides of the pocket, less the finish allowance H, are reached. The tool is then simultaneously raised by the clearance amount and moved at rapid feedrate back to the plunge-position. This completes machining of one L level.

The width of the last roughing cut is equal to the tool radius plus the finish allowance ($H + TR$). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

22.2 Pocket Milling Finishing Cycle (G88.2)

Use the G88.2 pocket milling finishing cycle to remove the finish allowance left on the sides of the rectangular or circular pockets, and slots. Use this cycle to finish a pocket formed by using a G88.1 roughing cycle. Typically a tool change is made between the G88.1 and the G88.2 cycles.

Important: Remember:

- the active plane is selected using G17, G18, or G19

In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

- tool length and diameter offsets must be entered and active prior to the G88 block
- if the radius of the finishing tool is larger than the radius of the roughing tool, some material may be left in the corners of the pocket after the finishing pass

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.2 finishing cycle for each of the possible pockets.

22.2.1 Rectangular Pocket Finishing Using G88.2

Use the G88.2 pocket milling finishing cycle to finish a rectangular pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a rectangular pocket during a G88.1 roughing cycle.

The G88.2 block used to finish a rectangular pocket has this format:

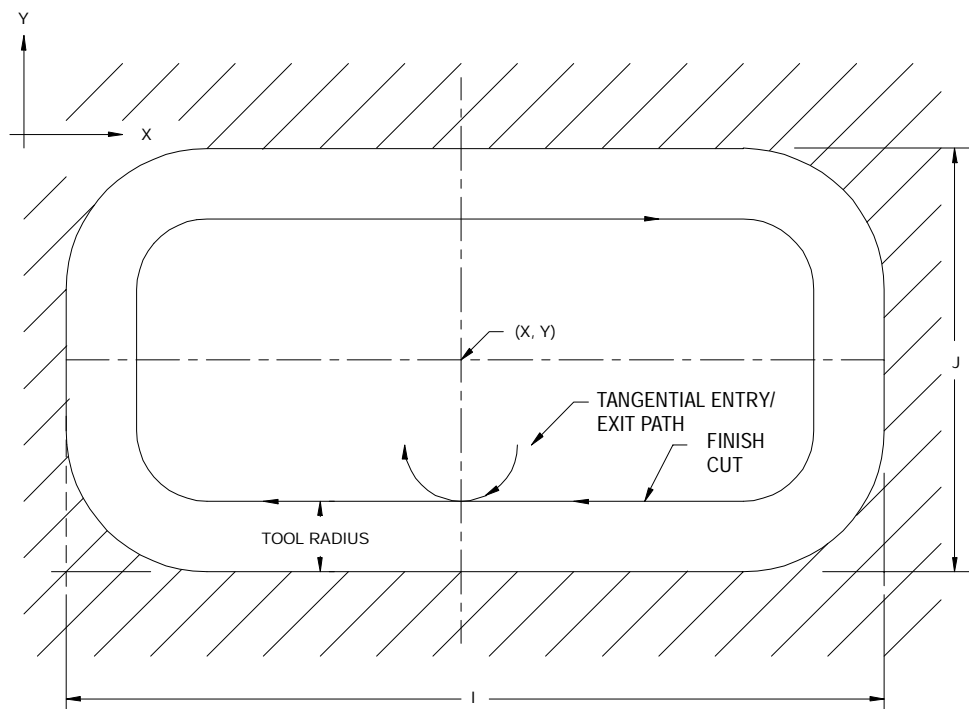
G88.2 X_Y_Z_I_J_(, R or , C) _P_H_L_F_;

Where :	Is :
X Y	The coordinates that specify the center of the rectangular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the rectangular pocket's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. Important: It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
, R	Corner radius. This is an optional parameter that is used to program rounded interior corners in the rectangular pocket.
, C	Corner chamfer. This is an optional parameter that is used to program chamfered interior corners in the rectangular pocket. Important: In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.
P	Direction of finishing cut. This parameter determines whether the finishing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. Important: Cutter compensation (G41/G42) must be disabled prior to the G88.2 block. The control generates an error if compensation is not disabled.
H	The finish allowance that will be left on the sides of the pocket. This is an optional parameter that is provided to allow for multiple finishing cuts.
L	Incremental plunge depth of each cutting pass plunge along the Z axis. If L is programmed, a finish pass is made at each L level. If L is not programmed, only one finishing pass is made at the programmed Z depth. This is an optional parameter. It is typically programmed when a very deep pocket is being finished.
F	Finishing feedrate. If not programmed the existing (modal) feedrate will be used.

Important: The rectangular pocket does not have to be parallel to the axes of the selected plane. It may be rotated by rotating the work coordinate system (G68). Refer to chapter 13 for additional information on rotating the work coordinate system.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool diameter by the control. The tangential entry/exit point is along the -X or -Y axis depending on which axis intersects the center of the longer side of the rectangular pocket. If the pocket is square, the tangential entry/exit point is along the -X axis.

Figure 22.6
Rectangular Pocket Finishing Using G88.2



Important: The tool should be positioned near the center of the pocket prior to the G88.2 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

The control starts the finish pass by moving the tool from the pocket center to the start point of the tangential entry/exit path. This start point is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter.

After the plunge operation, the tool is moved along the tangential entry/exit path to the tangential entry/exit point on the side of the pocket. A finish pass that ends at the tangential entry/exit point is then cut around the sides of the rectangular pocket. The control ends the finish pass by moving the tool from the tangential entry/exit point to the end point of the tangential entry/exit path.

If the programmed Z depth of the pocket has not been reached, the tool is simultaneously raised by the clearance amount and moved back to the tangential entry/exit path start point at the rapid feedrate. A plunge to the next level is then made. A finish pass is made at each L level until the pocket bottom is reached.

When all finish passes are completed, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the pocket by the clearance amount. This simultaneous move takes place at the rapid feedrate. The control then retracts the tool at rapid feedrate in all three axes back to the pre-cycle position of the tool.

22.2.2 Circular Pocket Finishing Using G88.2

Use the G88.2 pocket milling finishing cycle to finish a circular pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a circular pocket during a G88.1 cycle.

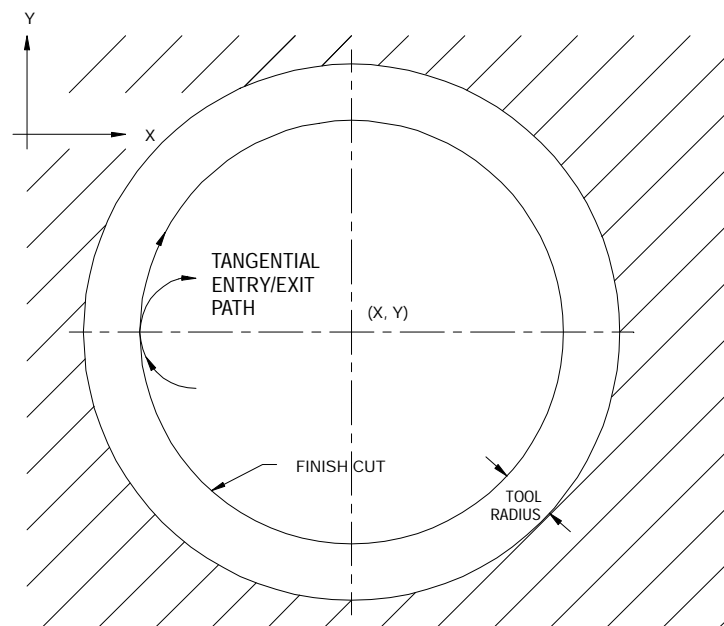
The G88.2 block used to finish a circular pocket has this format:

G88.2 X_Y_Z_R_P_H_L_F_;

Where :	Is :
X Y	The coordinates that specify the center of the circular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the circular pocket. This parameter must be programmed.
P	Direction of finishing cut.
H	Finish allowance.
L	Incremental plunge depth.
F	Finishing feedrate.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool radius by the control. The tangential entry/exit point will always be at the left side of the circular pocket along the -X axis.

Figure 22.7
Circular Pocket Finishing Using G88.2



Important: The tool should be positioned near the center of the pocket prior to the G88.2 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

Except for the shape of the pocket, the rest of the circular pocket finishing cycle is identical to that of a rectangular pocket finishing cycle.

22.2.3 Slot Finishing Using G88.2

Use the G88.2 pocket milling finishing cycle to finish a slot in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a slot during a G88.1 cycle.

The G88.2 block used to finish a slot has this format:

G88.2 X_Y_Z_I_R_P_H_L_F_; (X axis slot)

OR

G88.2 X_Y_Z_J_R_P_H_L_F_; (Y axis slot)

Where :	Is :
XY	The coordinates that specify the center of the slot.
Z	The coordinate (along the plunging axis) that specifies the bottom of the slot.
I J	The length of the slot as measured from the point where the axis intersects the arc at each end of the slot. I specifies the length of a X axis slot. J specifies the length of a Y axis slot. These are unsigned incremental values.
R	The radius of the arc at each end of the slot. The radius equals the slot-width/2. If not programmed, the control uses the tool radius value.
P	Direction of finishing cut.
H	Finish allowance.
L	Incremental plunge depth.
F	Finishing feedrate.

Important: The tool should be positioned near the center of the pocket prior to the G88.2 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

If the programmed R parameter is greater than the tool radius, this cycle is processed similar to a G88.2 finishing cycle for a rectangular pocket. The difference being that the R parameter programmed in a slot finishing cycle specifies the radius of the arc at the end of the slot versus the radius of the corners in a rectangular finishing cycle. The control cuts an arc at the short side a the rectangular pocket based on the programmed R parameter and the arc-center determined by the control.

If the programmed R parameter is equal to the tool radius, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it at the rapid feedrate to the arc-center at the -X or -Y end of the slot. This position is the plunge-position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. After the plunge operation a finishing cut is made at the finishing feedrate specified by the F parameter to the opposite arc-center at the +X or +Y end of the slot.

The tool is then raised to the initial Z level plus the clearance amount. If the programmed Z depth of the slot has not been reached, the tool is moved at rapid feedrate back to the plunge-position. A plunge to the next incremental L level or to the programmed Z level is made. Another finishing cut is made at the finishing feedrate to the arc-center at the +X or +Y end of the slot.

This process is repeated at each L level until the sides of the slot are finished. When all finish passes are completed, the control retracts the tool at rapid feedrate in all three axes back to the pre-cycle position of the tool.

END OF CHAPTER

Using Post Milling Cycles

23.0 Chapter Overview

This chapter describes how to use G88.3 and G88.4 to program post milling cycles. Use this table to find the information:

Information on:	On page:
Rectangular Post Roughing Using G88.3	23-2
Circular Post Roughing Using G88.3	23-5
Post Milling Finishing Cycle (G88.4)	23-7
Rectangular Post Roughing Using G88.4	23-8
Circular Post Finishing Using G88.4	23-11

23.1 Post Milling Roughing Cycle (G88.3)

Use the G88.3 post milling roughing cycle to rough out material outside of a specified area or post.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length, work coordinates, and diameter offsets must be entered and active prior to the G88 block. The radius/diameter of the tool can not exceed the length of the shortest side of the rectangular pocket. If it does, the control enters Cycle-Stop mode and displays the error message "TOOL RADIUS TOO LARGE" on the CRT.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey
- Work coordinate offset changes through the offset softkeys

The following subsections cover using the G88.3 roughing cycle for rectangular or circular posts.

23.1.1 Rectangular Post Roughing Using G88.3

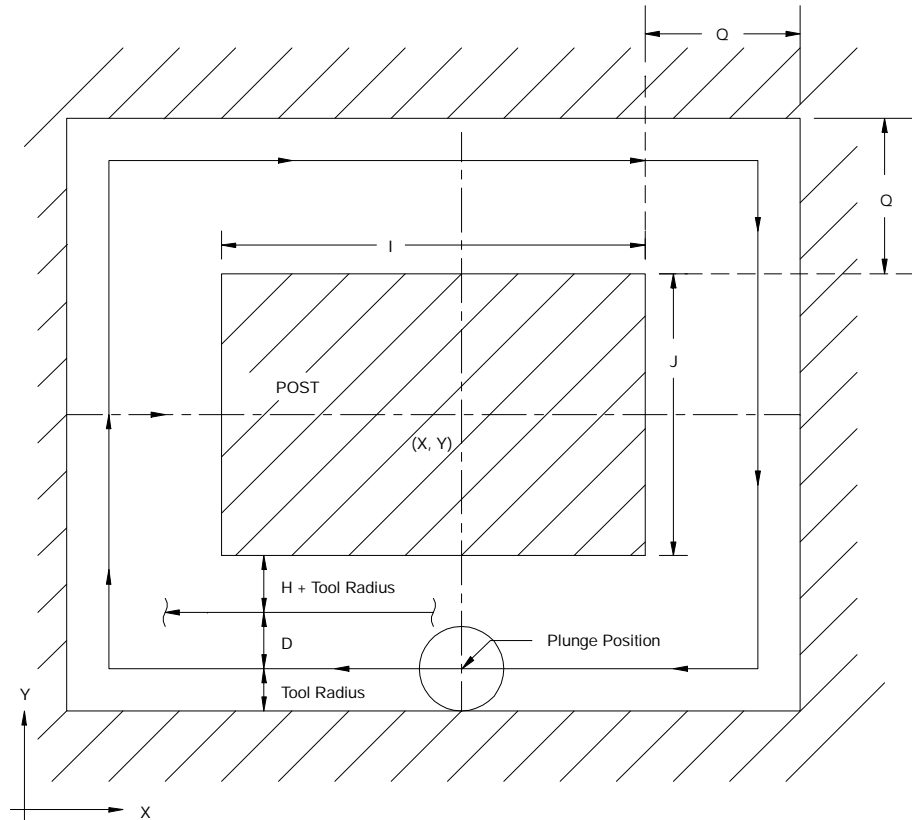
Use the G88.3 post milling roughing cycle to rough out a rectangular post in a workpiece. This cycle makes multiple cuts at a programmed width and depth.

The G88.3 block used to rough out a rectangular post has this format:

G88.3 X_Y_Z_I_J_Q_(, R or, C) _P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the rectangular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the post's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. Important: It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
Q	Unsigned incremental value that specifies the distance from the sides of the post to the outer perimeter of the pocket. This distance is the same whether measured along the X or the Y axis.
, R	Corner radius. This is an optional parameter that is used to program rounded interior corners in the rectangular pocket.
, C	Corner chamfer. This is an optional parameter that is used to program chamfered interior corners in the rectangular pocket. Important: In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.
P	Direction of roughing cut. This parameter determines whether the roughing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. Important: If cutter compensation (G41/G42) was enabled prior to the G88.3 block, it is disabled when G88.3 is enabled.
H	The finish allowance that will be left on the sides of the post. This finish allowance can be removed later using a G88.4 finishing cycle. To leave a finish allowance on the pocket bottom, program a pocket depth (Z parameter) that is at the desired finish allowance above the actual pocket bottom. This finish allowance can be removed later using a G88.4 finishing cycle.
D	Roughing cut thickness. This parameter specifies the maximum width of any XY axis roughing cuts. This is an optional parameter. If not programmed, the control uses the default thickness, which is equal to half of the current tool diameter. Important: The roughing cut thickness can not be greater than the diameter of the current tool. If it is, the control enters Cycle-Stop mode and displays the error message "D-WORD LARGER THAN TOOL DIAMETER" on the CRT.
L	Incremental plunge depth of each cutting pass along the Z axis. This is an optional parameter. If not programmed, the plunge amount will be equal to the programmed depth of the pocket.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis moves. If not programmed the existing (modal) feedrate will be used.

Figure 23.1
Rectangular Post Roughing Using G88.3



Important: The tool should be positioned near the center of the post prior to the G88.3 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

Depending on which axis intersects the longer side of the pocket, the tool is moved along either the -X or the -Y axis to the outer perimeter of the pocket to be machined around the post. This position is the plunge-position of the cycle, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate.

After the plunge operation, the control moves the tool in a rectangular path, defined by the programmed sides of the rectangular post, that starts and ends at the same point on either the -X or the -Y axis. After completing a rectangular path, the control makes a single-axis rough cut along the -X or -Y axis towards the center of the rectangular post. Another rectangular path is cut that ends at the -X or -Y axis. This process is repeated until the sides of the post, less the finish allowance H, are reached.

The width of the first roughing cut is equal to the tool radius. The width of the last roughing cut is equal to the tool radius plus the finish allowance ($H + TR$). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

The tool is then simultaneously raised by the clearance amount and moved along either the -X or -Y axis at rapid feedrate back to the plunge-position. This completes machining of one L level.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

Once the post has been machined out, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the post by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate along the X, Y, and Z axes to the pre-cycle position of the tool.

If ,R or ,C is not programmed in the G88.3 block, each corner of the rectangular post is squared off as much as the tool radius will allow. If ,R or ,C is programmed in the G88.3 block, the corners of the post will either be rounded or chamfered. Refer to chapter 16, Using Chamfers and Corner Radius, for additional information on chamfering and corner rounding.

23.1.2
Circular Post Roughing
Using G88.3

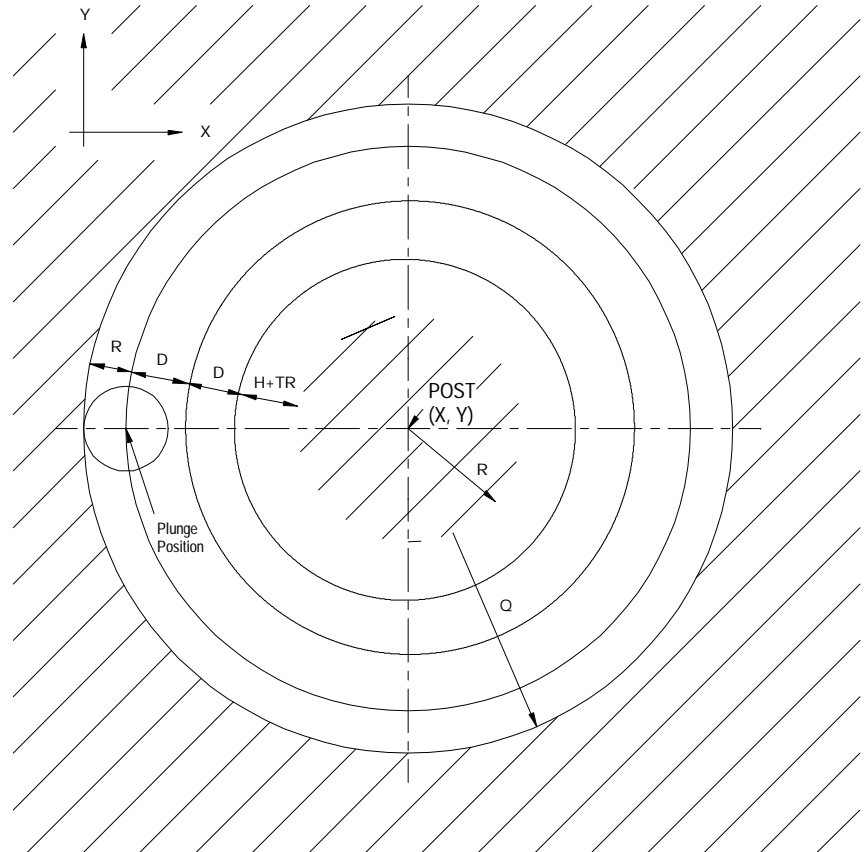
Use the G88.3 post milling roughing cycle to rough out a circular post in a workpiece. This cycle makes multiple circular cuts at a programmed width and depth.

The G88.3 block used to rough out a circular post has this format:

G88.3 X_Y_Z_R_Q_P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the circular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the circular post. This parameter must be programmed.
Q	Unsigned incremental value that specifies the distance from the sides of the circular post to the outer perimeter of the circular pocket.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 23.2
Circular Post Roughing Using G88.3



Important: The tool should be positioned near the center of the post prior to the G88.3 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

The tool is then moved along the -X axis to the outer perimeter of the pocket. This position is the plunge- position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate.

After the plunge operation, the control moves the tool in a circular path that starts and ends at the same point on the -X axis. After completing a circular path, the control makes a single-axis rough cut towards the post along the -X axis. Another circular path is cut that start and ends at the -X axis. This process is repeated until the sides of the pocket, less the finish allowance H, are reached.

The width of the first roughing cut is equal to the tool radius. The width of the last roughing cut is equal to the tool radius plus the finish allowance (H + TR). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

The tool is then simultaneously raised by the clearance amount and moved at rapid feedrate along the -X axis back to the plunge-position. This completes the machining of one L level.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

Once the post has been machined out, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the post by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate along the X, Y, and Z axes to the pre-cycle position of the tool.

23.2 Post Milling Finishing Cycle (G88.4)

Use the G88.4 post milling finishing cycle to remove the finish allowance left on the sides of a rectangular or circular post. You can use this cycle to finish a post formed by using a G88.3 roughing cycle. Typically a tool change is made between the G88.3 and the G88.4 cycles.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length, work coordinates, and diameter offsets must be entered and active prior to the G88 block. The radius/diameter of the tool can not exceed the length of the shortest side of the pocket. If it does, the control enters Cycle-Stop mode and displays the error message “TOOL RADIUS TOO LARGE.”

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey
- Work coordinate offset changes through the offset softkeys

The following subsections cover using the G88.4 finishing cycle for rectangular or circular posts.

23.2.1 Rectangular Post Finishing Using G88.4

Use the G88.4 post milling finishing cycle to finish a rectangular post in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a rectangular post during a G88.3 cycle.

The G88.4 block used to finish a rectangular post has this format:

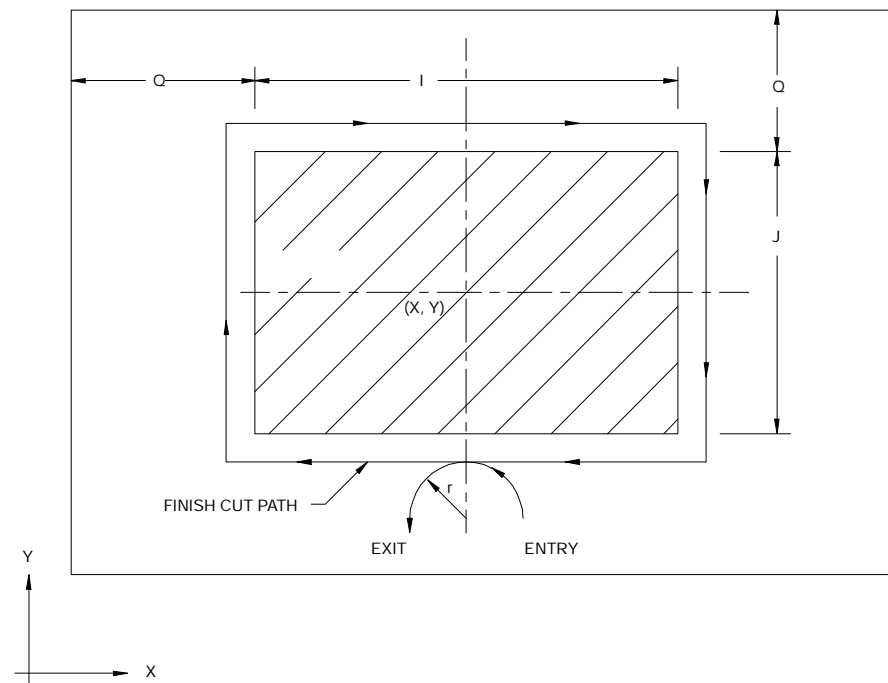
G88.4 X_Y_Z_I_J_Q_(, R or , C)_P_H_L_F_;

Where :	Is :
X Y	The coordinates that specify the center of the rectangular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the post's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. Important: It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
Q	Unsigned incremental value that specifies the distance from the sides of the post to the outer perimeter of the pocket. This distance is the same whether measured along the X or the Y axis.
, R	Corner radius. This is an optional parameter that is used to program rounded post corners.
, C	Corner chamfer. This is an optional parameter that is used to program chamfered rounded post corners. Important: In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.
P	Direction of finishing cut. This parameter determines whether the finishing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. Important: If cutter compensation (G41/G42) was enabled prior to the G88.4 block, it is disabled when G88.4 is enabled.
H	The finish allowance that will be left on the sides of the post. This is an optional parameter that is provided to allow for multiple finishing cuts.

Where :	Is :
L	Incremental plunge depth of each cutting pass along the Z axis. If L is programmed, a finish pass is made at each L level. If L is not programmed, only one finishing pass is made at the programmed Z depth. This is an optional parameter. It is typically programmed when a very deep pocket is being finished.
F	Finishing feedrate. If not programmed the existing (modal) feedrate will be used.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool radius by the control. The tangential entry/exit point is along the -X or -Y axis depending on which axis intersects the center of the longer side of the rectangular post. If the rectangular post is square, the tangential entry/exit point will be along the -X axis.

Figure 23.3
Rectangular Post Finishing Using G88.4



Important: The tool should be positioned near the center of the post prior to the G88.4 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

The control starts the finish pass by moving the tool from the post center to the start point of the tangential entry/exit path. This start point is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter.

After the plunge operation, the tool is moved along the tangential entry/exit path to the tangential entry/exit point on the side of the post. A finish pass that ends at the tangential entry/exit point is then cut around the sides of the rectangular post. The control ends the finish pass by moving the tool from the tangential entry/exit point to the end point of the tangential entry/exit path.

If the programmed Z depth of the post has not been reached, the tool is simultaneously raised by the clearance amount and moved back to the tangential entry/exit path start point at the rapid feedrate. A plunge to the next level is then made. A finish pass is made at each L level until the pocket bottom is reached.

When all finish passes are completed, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the post by the clearance amount. This simultaneous move takes place at the rapid feedrate. The control then retracts the tool at rapid feedrate in all three axes back to the pre-cycle position of the tool.

23.2.2
Circular Post Finishing
Using G88.4

Use the G88.4 post milling finishing cycle to finish a circular post in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a circular post during a G88.3 cycle.

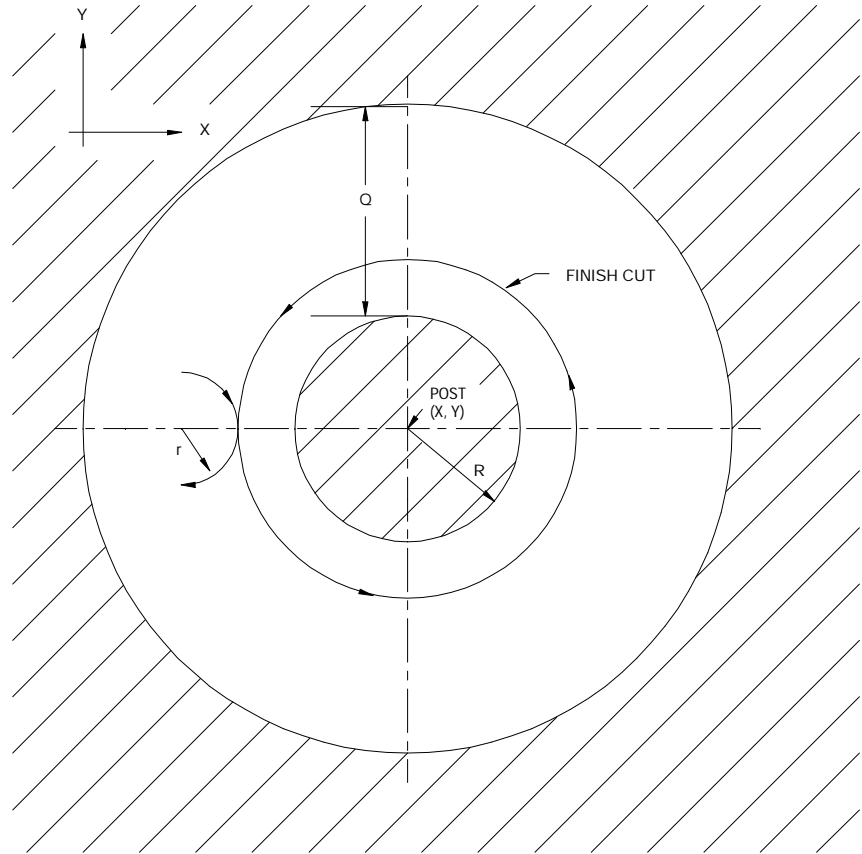
The G88.4 block used to finish a circular post has this format:

G88.4 X_Y_Z_Q_R_P_H_L_F_;

Where :	Is :
X Y	The coordinates that specify the center of the circular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
Q	Unsigned incremental value that specifies the distance from the sides of the circular post to the outer perimeter of the circular pocket.
R	The radius of the circular post. This parameter must be programmed.
P	Direction of finishing cut.
H	Finish allowance.
L	Incremental plunge depth.
F	Finishing feedrate.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool radius by the control. The tangential entry and exit will always occur on the left side of the circular post at the -X axis.

Figure 23.4
Circular Post Finishing Using G88.4



Important: The tool should be positioned near the center of the post prior to the G88.4 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the circular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

Except for the shape of the post, the rest of the circular post finishing cycle is identical to that of a rectangular post finishing cycle.

END OF CHAPTER

Using Hemisphere Milling Cycles

24.0 Chapter Overview

This chapter describes how to use G88.5 and G88.6 to program hemisphere milling cycles. Use this table to find information:

Information on:	On page:
Hemisphere Milling Roughing Cycle (G88.5)	24-1
Concave Hemisphere Roughing Using G88.5	24-2
Convex Hemisphere Roughing Using G88.5	24-5
Hemisphere Milling Finishing Cycle	24-7
Concave Hemisphere Finishing Using G88.6	24-8
Convex Hemisphere Finishing Using G88.6	24-10

24.1 Hemisphere Milling Roughing Cycle (G88.5)

Use the G88.5 hemisphere milling roughing cycle to rough out concave or convex hemispherical pockets.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length and diameter offsets must be entered and active prior to the G88 block.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.5 roughing cycle for concave or convex hemispheres.

24.1.1
Concave Hemisphere
Roughing Using G88.5

Use the G88.5 concave milling roughing cycle to rough out a concave pocket in a workpiece. This cycle makes multiple concentric circular cuts at a programmed width and depth.

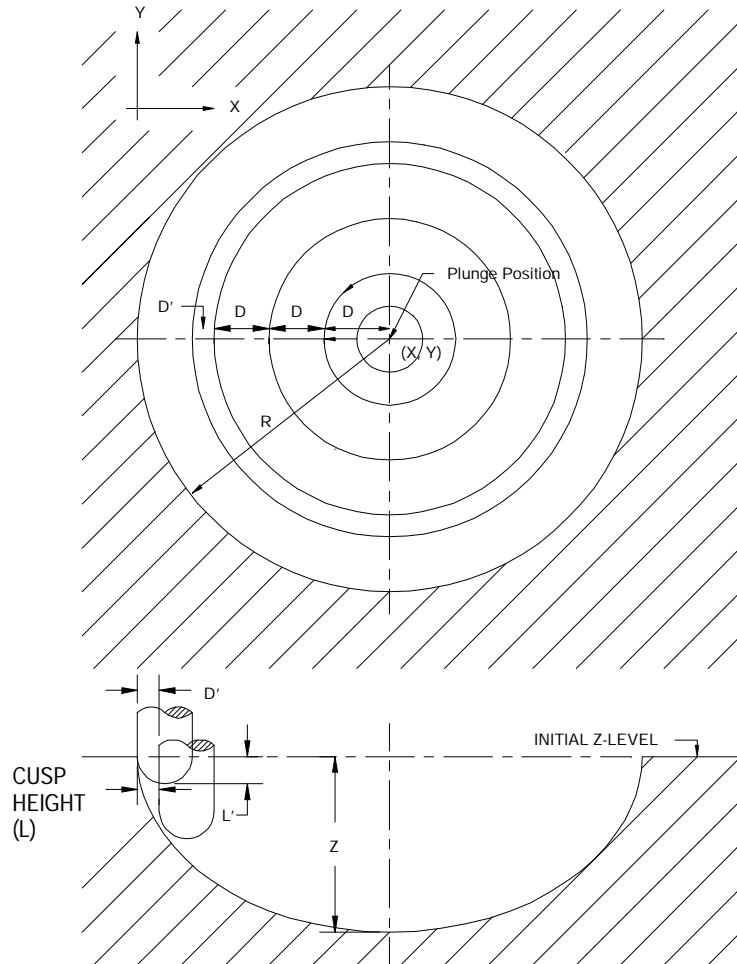
The G88.5 block used to rough out a concave pocket has this format:

G88.5 X_Y_Z_R_Q0_P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the concave hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the bottom of the hemispherical pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
R	Radius of the concave hemisphere.
Q0	Code specifying a concave hemisphere.
P	Direction of roughing cut. This parameter determines whether the roughing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. Important: If cutter compensation (G41/G42) was enabled prior to the G88.5 block, it is disabled when G88.5 is enabled.
H	The finish allowance that will be left on the sides of the concave hemisphere. This finish allowance can be removed later using a G88.6 finishing cycle.
D	Roughing cut thickness. This parameter specifies the maximum width of any XY axis roughing cuts. This is an optional parameter. If not programmed, the control uses the default thickness, which is equal to half of the current tool diameter. Important: The roughing cut thickness can not be greater than the diameter of the current tool. If it is, the control enters Cycle-Stop mode and displays the error message "D-WORD LARGER THAN TOOL DIAMETER" on the CRT.
L	Roughing cusp height. This value of this parameter is a measurement that corresponds to the height of the material left along the sides of the hemisphere after each X-Z roughing plunge.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis moves. If not programmed the existing (modal) feedrate will be used.

If Q0 is programmed, the control generates a concave hemisphere. The control makes multiple concentric circular cuts at each level of the concave hemisphere. The depth of the pocket increases and the diameter of each level decreases until the bottom center of the concave hemisphere is reached.

Figure 24.1
Concave Hemisphere Roughing Using G88.5



Important: The tool should be positioned near the center of the concave hemisphere prior to the G88.5 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the concave hemisphere specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the concave hemisphere is the plunge position, the position where the plunge to the programmed depth takes place.

Prior to each plunge, the control computes a delta rough cut thickness, D' , and a delta plunge depth, L' . These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.5 block, and the tool radius programmed prior to the G88.5 block.

With the axis positioned at the plunge-position, a plunge along the Z axis, of depth L' is performed. This plunge takes place at the plunge feedrate specified by the E parameter.

After the plunge operation, the control performs a single-axis rough cut outwards towards the left on the $-X$ axis of the hemisphere. The width of the first roughing cut is equal to the programmed rough cut thickness, D , divided by two then multiplied by 95% ($(D/2) \times (.95)$). The control then moves the tool in a 360 degree circular path around the plunge-position.

After completing a 360 degree circular path, the control makes another single-axis rough cut outwards along the $-X$ axis by the rough cut thickness, D . Another 360 degree circular path is cut that ends at the $-X$ axis. This process is repeated until the side of concave hemisphere, less the finish allowance H , is reached.

Important: The thickness of the last rough cut is D' not the programmed D parameter.

The tool is then raised by the clearance amount (AMP selectable, refer to system installers literature) and moved at rapid feedrate back to the plunge-position. This completes machining of one L level.

If the programmed Z depth of the pocket has not been reached, another plunge takes place along the Z axis to the next L level. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position.

24.1.2
Convex Hemisphere
Roughing Using G88.5

Use the G88.5 convex milling roughing cycle to rough out a convex pocket in a workpiece. This cycle makes multiple concentric circular cuts at a programmed width and depth from the top center of the convex hemisphere to the outermost diameter of the convex hemisphere.

The G88.5 block used to rough out a convex pocket has this format:

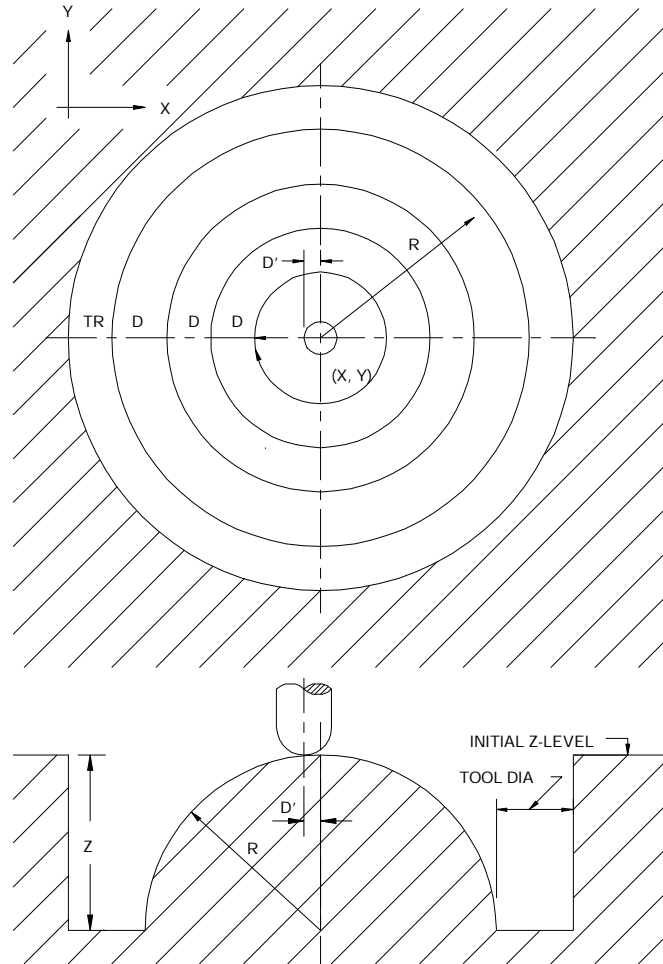
G88.5 X_Y_Z_R_Q1_P_H_D_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the convex hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the base of the convex hemisphere.
R	Radius of the convex hemisphere.
Q1	Code specifying a convex hemisphere.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Roughing cusp height.
E	Plunge feedrate.
F	Roughing feedrate.

If Q1 is programmed, the control generates a convex hemisphere. The control makes multiple concentric circular cuts at each level of the convex hemisphere. The depth of the pocket and the diameter of each level increases until the outer diameter of the convex hemisphere is reached.

Important: The tool should be positioned near the center of the convex hemisphere prior to the G88.5 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

Figure 24.2
Convex Hemisphere Roughing Using G88.5



From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the convex hemisphere specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the convex hemisphere is the plunge position, the position where the plunge to the programmed depth takes place.

Prior to each plunge, the control computes a delta rough cut thickness, D' , and a delta plunge depth, L' . These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.5 block, and the tool radius programmed prior to the G88.5 block.

With a convex hemisphere, the plunge is actually a contour move to the outward along the -X axis. This move cuts along the spherical contour, axes X and Z, at the plunge feedrate specified by the E parameter. This plunge simultaneously moves the X and Z axes by the D' and L' amounts.

After the plunge, the control moves the tool in a 360 degree circular path around the plunge-position. After completing this circular path, the control makes a rough cut, D parameter, outwards along the -X axis. Another 360 degree circular path is cut that ends at the -X axis. This process is repeated until the outer diameter of the pocket, plus the finish allowance H, is reached. The tool is then raised by the clearance amount and moved at rapid feedrate back to the initial cut at the current level. This completes machining of one L level.

Important: The outer diameter of the pocket is equal to the radius of the hemisphere plus the tool diameter and the finish allowance.

If the programmed Z depth of the pocket has not been reached, another plunge takes place simultaneously along the X and Z axes to the next L' level. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth of the convex hemisphere is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position.

24.2 Hemisphere Milling Finishing Cycle (G88.6)

Use the G88.6 hemisphere milling finishing cycle to remove the finish allowance left on the sides of a concave or convex hemisphere. You can use this cycle to finish a hemisphere formed by using a G88.5 roughing cycle. Typically a tool change is made between the G88.5 and the G88.6 cycles.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length and diameter offsets must be entered and active prior to the G88 block.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.6 finishing cycle for concave or convex hemispheres.

24.2.1 Concave Hemisphere Finishing Using G88.6

Use the G88.6 concave milling finishing cycle to finish a concave pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of the concave hemisphere during the G88.5 roughing cycle.

The G88.6 block used to finish a concave pocket has this format:

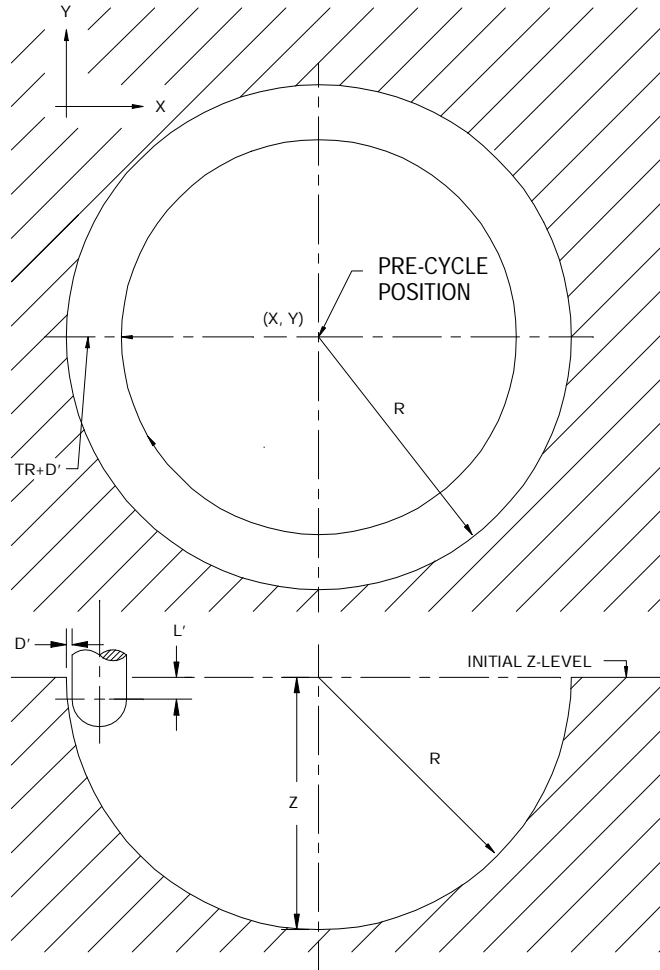
G88.6 X_Y_Z_R_Q0_P_H_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the concave hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the bottom of the hemispherical pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
R	Radius of the concave hemisphere.
Q0	Code specifying a concave hemisphere.
P	Direction of finishing cut. This parameter determines whether the finishing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. Important: If cutter compensation (G41/G42) was enabled prior to the G88.6 block, it is disabled when G88.6 is enabled.
H	The finish allowance that will be left on the sides of the concave hemisphere. This is an optional parameter that is provided to allow for multiple finishing cuts.
L	Finishing cusp height. This value of this parameter is a measurement that corresponds to the height of the material left along the sides of the hemisphere after each X-Z roughing plunge.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F) will be used.
F	Finishing feedrate. If not programmed the existing (modal) feedrate will be used.

If Q0 is programmed, the control makes a finish pass around the sides of a concave hemisphere. The control makes multiple circular finish cuts at each level of the concave hemisphere. The depth of the pocket increases and the diameter of each level decreases until the bottom center of the concave hemisphere is reached.

Prior to each finish plunge, the control computes a delta finish cut thickness, D', and a delta plunge depth, L'. These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.6 block, and the tool radius programmed prior to the G88.6 block.

Figure 24.3
Concave Hemisphere Finishing Using G88.6



Important: The tool should be positioned near the center of the concave hemisphere prior to the G88.6 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

The control raises the tool by the clearance amount (AMP selectable, refer to system installers literature) and moves it at the rapid feedrate to the center of the concave hemisphere, X and Y coordinates.

The control initially moves the tool down L' from the initial Z level and offsets it from the side of the concave hemisphere by D' . A 360 degree circular path is cut, defined by the center of the concave hemisphere and the radius of the hemisphere, that ends at the -X axis.

If the programmed Z depth of the pocket has not been reached, another plunge takes place simultaneously along the X and Z axes to the next L' level. Another 360 degree circular path is cut. This process is repeated until the programmed Z depth of the concave hemisphere is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position. This completes the finishing cycle.

24.2.2 Convex Hemisphere Finishing Using G88.6

Use the G88.6 convex milling finishing cycle to finish a convex pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of the convex hemisphere during the G88.5 roughing cycle.

The G88.6 block used to finish a convex pocket has this format:

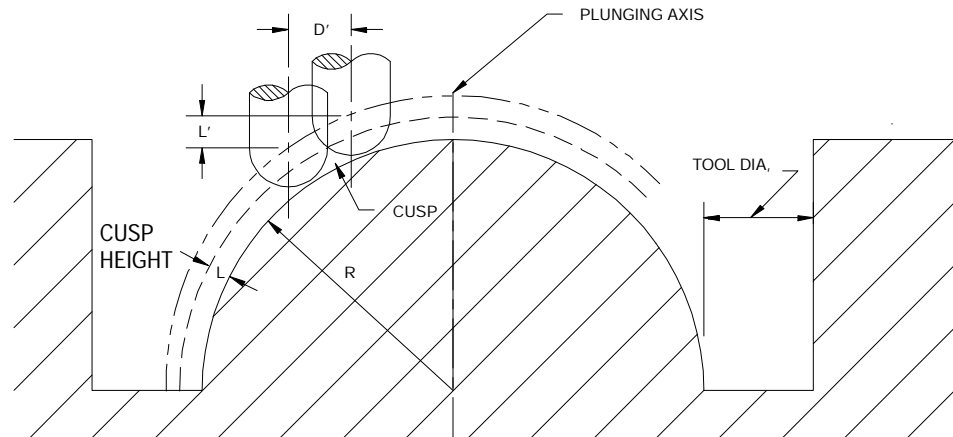
G88.6 X_Y_Z_R_Q1_P_H_L_E_F_;

Where :	Is :
X Y	The coordinates that specify the center of the convex hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the bottom of the hemispherical pocket.
R	Radius of the convex hemisphere.
Q1	Code specifying a convex hemisphere.
P	Direction of finishing cut.
H	Finish allowance.
L	Finishing cusp height.
E	Plunge feedrate.
F	Finishing feedrate.

If Q1 is programmed, the control makes a finish pass around the sides of a convex hemisphere. The control makes multiple circular finish cuts at each level of the convex hemisphere. The depth of the pocket and the diameter of each level increases until the outer diameter of the convex hemisphere is reached.

Prior to each finish plunge, the control computes a delta finish cut thickness, D', and a delta plunge depth, L'. These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.6 block, and the tool radius programmed prior to the G88.6 block.

Figure 24.4
Convex Hemisphere Finishing Using G88.6



Important: The tool should be positioned near the center of the convex hemisphere prior to the G88.6 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

The control raises the tool by the clearance amount (AMP selectable, refer to system installers literature) and moves it at the rapid feedrate to the center of the convex hemisphere, X and Y coordinates.

With a convex hemisphere, the plunge is actually a contour move to the left along the -X axis. This move cuts along the spherical contour, axes X and Z, at the plunge feedrate specified by the E parameter. This plunge simultaneously moves the X and Z axes by the D' and L' amounts.

After the plunge, the control moves the tool in a 360 degree circular path, defined by the center of the convex hemisphere and the radius of the hemisphere, around the plunge-position.

Important: The thickness of the first rough cut is D' not the programmed D parameter.

If the programmed Z depth of the pocket has not been reached, another plunge takes place simultaneously along the X and Z axes to the next L' level. This plunge simultaneously moves the X and Z axes by the D' and L' amounts. This level is then finished as described in the previous paragraphs. This process is repeated until the programmed Z depth of the convex pocket is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position. This completes the finishing cycle.

Important: The radius of the pocket surrounding the convex hemisphere is equal to the radius of the hemisphere plus the tool radius and the finish allowance.

END OF CHAPTER

Irregular Pocket Milling Cycles

25.0 Chapter Overview

Important: The Irregular Pocket Milling Cycles feature (G89.1 and G89.2) is only available prior to release 12.xx. Any attempt to program a G89.1 or G89.2 in release 12.xx or later will result in the error message, “Illegal G-code”.

This chapter describes how to use G89.1 and G89.2 to program irregular pocket milling cycles. Use this table to find information:

Information on:	On page:
Irregular Pocket Roughing	25-2
Irregular Pocket Finishing	25-10

25.1 Irregular Pocket Milling

Use the G89.1 irregular pocket milling roughing cycle to rough out irregular pockets. Irregular pockets are pockets that are not rectangular, circular, or hemispherical. It is possible to form an irregular post inside an irregular pocket by combining two or more irregular pocket cycles.

You can use the irregular pocket milling finishing cycle (G89.2) to finish an irregular pocket in a workpiece. This cycle is typically used to finish an irregular pocket formed using a G89.1 irregular pocket roughing cycle. A tool change may be performed between the G89.1 and G89.2 cycles.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length and diameter offsets must be entered and active prior to the G89 block.

These features are prohibited during execution of irregular pocket milling cycles:

- MDI mode
- Tool offset changes

The following subsection covers using the G89.1 roughing cycle for irregular pockets.

25.1.1
Irregular Pocket Roughing
(G89.1)

Use the irregular pocket milling roughing cycle (G89.1) to rough out an irregular pocket in a workpiece. This cycle makes multiple cuts at a programmed depth, one cutter radius in width.

The G89.1 block used to rough out an irregular pocket has this format:

G89.1 X_Y_Z_P_Q_H_E_F_L_;

Where :	Is :
X Y	The coordinates that specify the start/end corner of the irregular pocket in the selected plane. These parameters must be programmed.
Z	The coordinate (along the plunging axis) that specifies the bottom of the irregular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start-point to the pocket bottom. This parameter must be programmed.
P	The sequence number of the first block in the set of blocks that define the pocket shape. The coordinates in this block specify the end-point of the pocket's first wall.
Q	The sequence number of the last block in the set of blocks that define the pocket shape. The coordinates in this block specify the end-point of the pocket's last wall. This endpoint must correspond to the start/end corner specified by the X and Y coordinates. This parameter must be programmed. Important: The set of blocks specified by P and Q must be at least 5 blocks long to qualify as an irregular pocket. The set of blocks specified by P and Q may be located anywhere in the same program as the G89.1 block. These blocks can not be called from a subprogram or a macro unless the G89.1 block is in that subprogram or macro.
H	The finish allowance that will be left on the sides of the irregular pocket. This finish allowance can be removed later using a G89.2 finishing cycle. To leave a finish allowance on the pocket bottom, program a pocket depth (Z parameter) that is at the desired finish allowance above the actual pocket bottom. This finish allowance can be removed later using a G89.1 roughing cycle programmed with the actual pocket bottom.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F parameter) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis cutting moves. If not programmed the existing (modal) cutting feedrate will be used.
L	Incremental plunge depth of each cutting pass along the Z axis. If not programmed, the plunge amount will be equal to the programmed depth of the pocket.

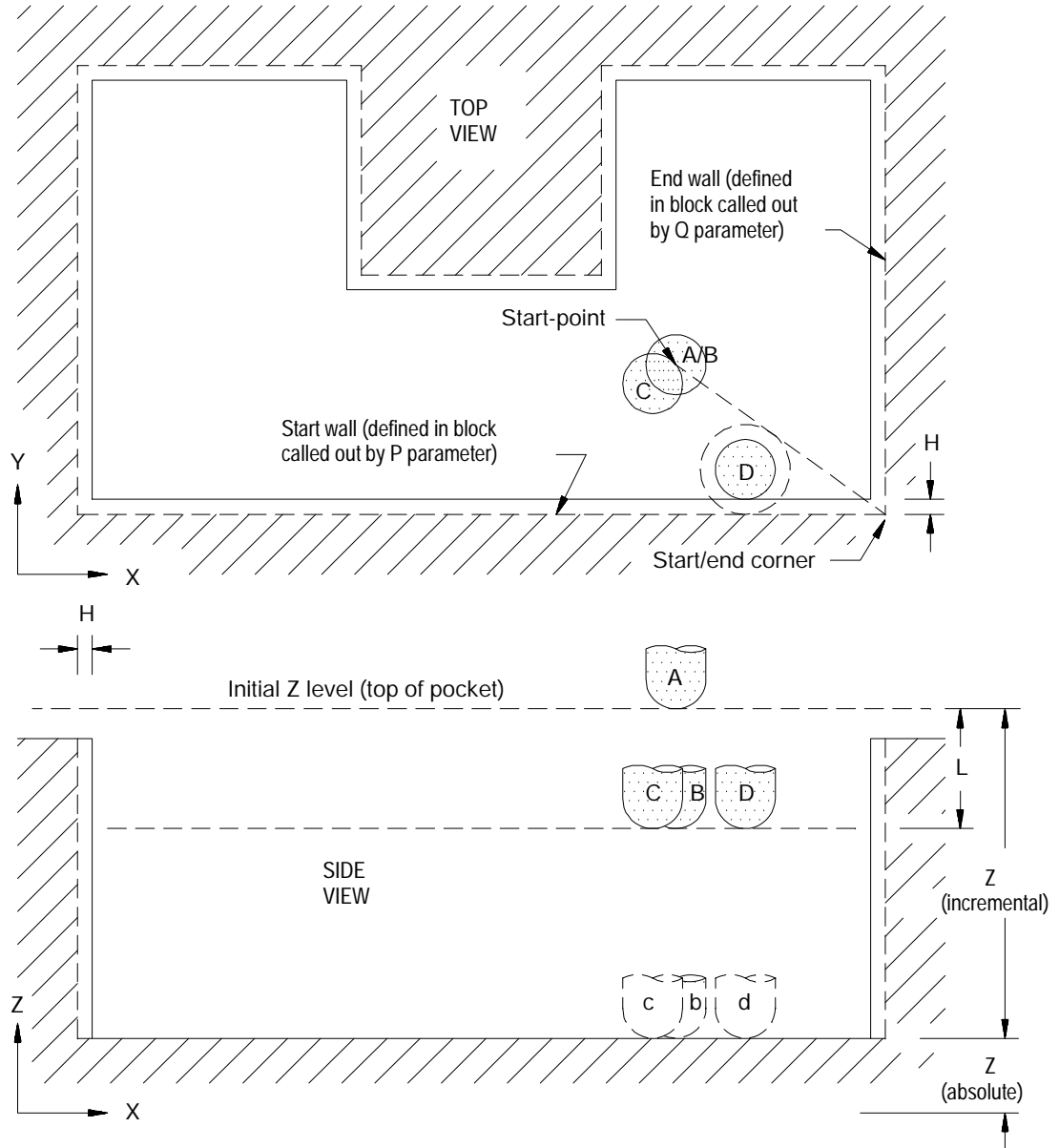
Before invoking the G89.1 cycle, the programmer must activate cutter compensation left or right by programming G41 or G42. This allows the control to begin interpreting the blocks that define the contour of the pocket as they are encountered.

Prior to the G89.1 block, the tool should be positioned near the start/end corner of the pocket and should be just above but not touching the part. This position is referred to as the start-point of the cycle (A in figure 16.16). From the start-point the cutter must be able to move down into the part and then directly over to the start/end corner of the pocket without cutting into any wall of the pocket. The Z coordinate of this position determines the initial Z level or top of the pocket.

Once the axes are positioned at the start-point, a plunge along the Z axis takes place. If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter (B in Figure 25.1). If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter (B in Figure 25.1). This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate specified by the F parameter.

After the plunge, the control uses the active cutter compensation mode (G41/G42) to offset the cutter one cutter radius perpendicular to the line from the start-point to the start/end corner (C in Figure 25.1). The control then moves the cutter from this offset position to a point where it is located one cutter radius plus the finish allowance away from both the first wall of the pocket and the line from the start-point to the start/end corner (D in Figure 25.1). This move takes place at the roughing feedrate.

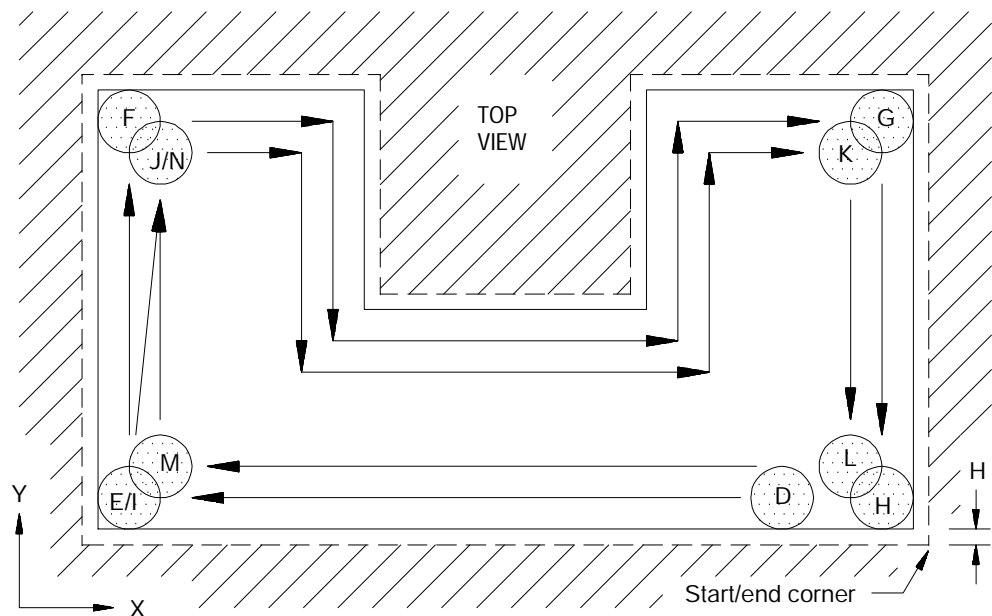
Figure 25.1
Irregular Pocket Roughing Cycle Entry Moves



From the final cutter position of Figure 25.1, the control moves the cutter twice around the programmed contour of the pocket as shown in Figure 25.2. The first pass around the pocket removes material twice the cutter radius (cutter diameter) in width (D through I in Figure 25.2). The second pass removes material that is one cutter radius in width (I through N in Figure 25.2).

These two passes cut a channel around the inside perimeter of the pocket that provides clearance for the cutter to be raised and lowered as necessary at the beginning and end of the rest of the roughing passes. While cutting this channel, the control automatically adjusts the roughing feedrate so that the volume of material being removed per unit time is the same as will be removed later during normal roughing passes.

Figure 25.2
Irregular Pocket Initial Roughing Passes

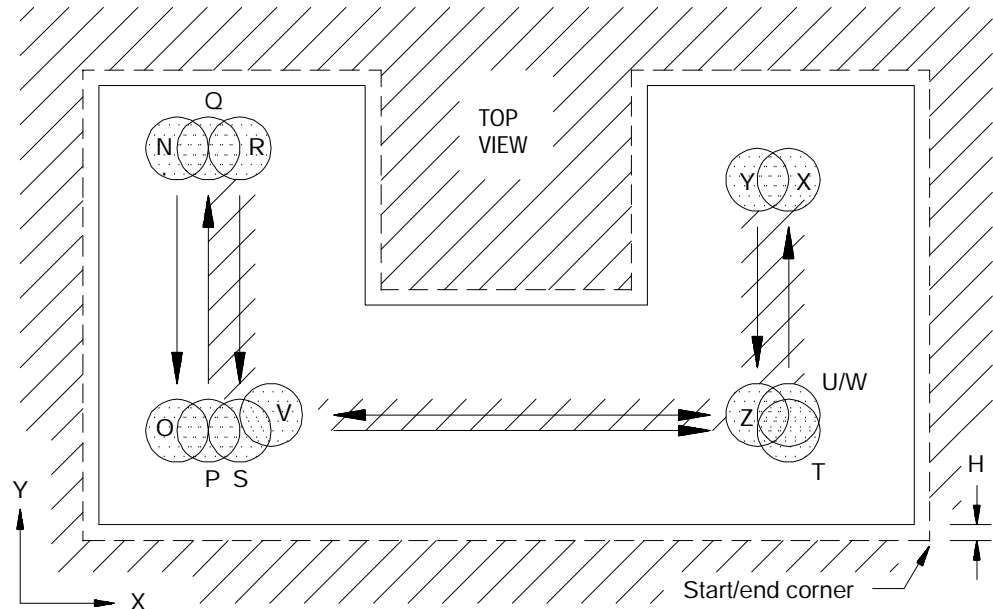


After the initial roughing passes are complete, the cutter is near the pocket corner that corresponds to the endpoint of the block that follows the block specified by the P parameter in the G89.1 block (point N in Figure 25.3).

The control analyzes the remaining pocket area to be machined-out and determines which area to machine first. Any undone area within one cutter radius of the current cutter position will be machined-out first.

The area within one cutter radius of the current cutter position will be machined-out in a series of straight-line step-over roughing passes. The orientation of these passes, either parallel to the X or the Y axis, is dependent upon the pocket contour. The control determines whether passes parallel to the X or to the Y axis will machine-out the most material in the fewest amount of passes and orients the passes accordingly (N through Z in Figure 25.3).

Figure 25.3
Roughing-Out Adjacent Areas in an Irregular Pocket

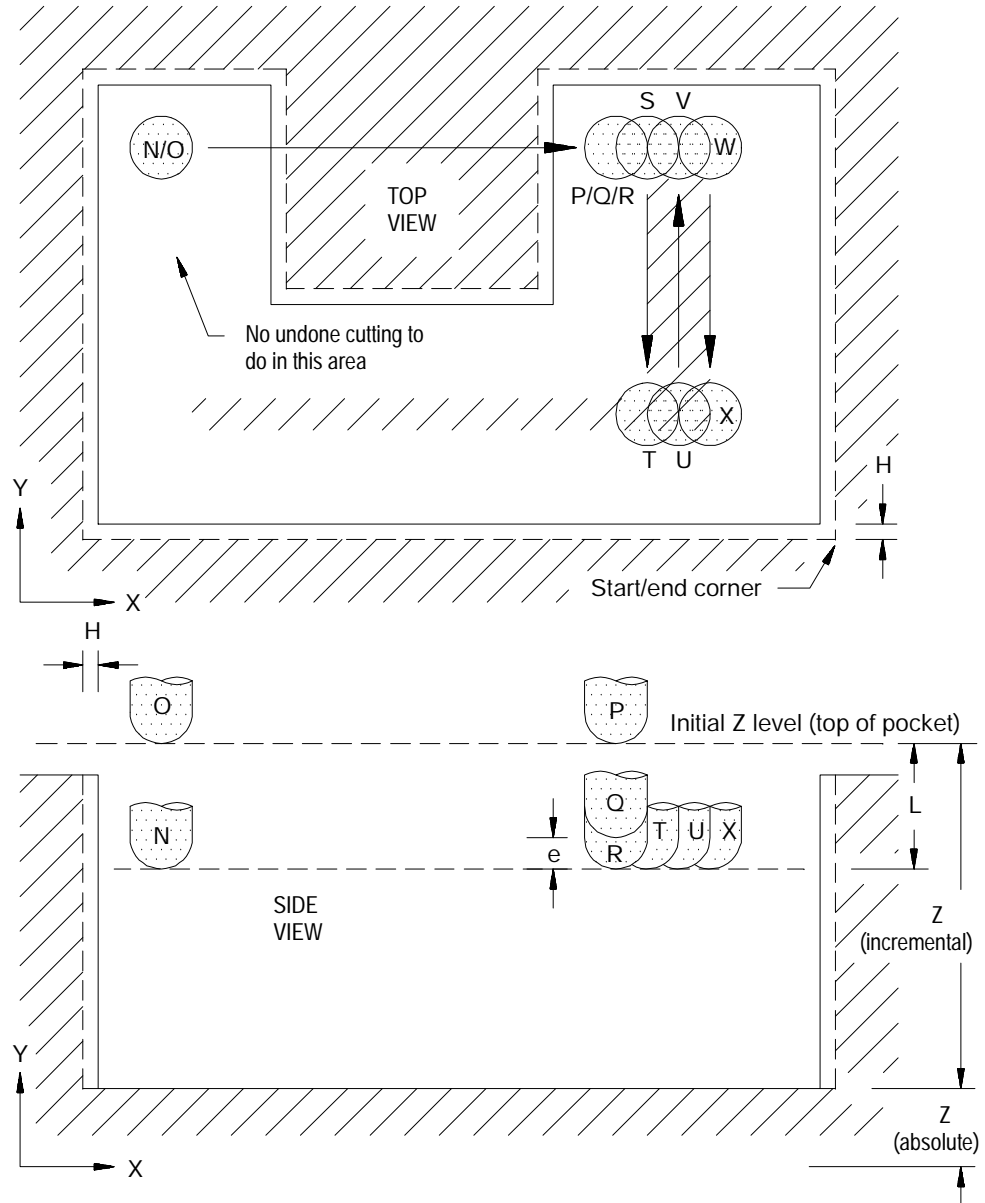


If there is no undone area within one cutter radius of the current cutter position, the control raises the cutter to the initial Z level (point O in figure 16.19). The cutter is then moved at the rapid feedrate to the nearest position just outside the nearest undone area (point P) and lowered to the current plunge-level minus the clearance amount (point Q) at the rapid feedrate. The clearance amount (e) is AMP selectable, refer to the literature provided by your system installer. The cutter is then lowered to the current plunge-level (point R) at the plunge feedrate.

Important: The control will move the cutter to the nearest undone area without raising the cutter to the initial Z level if it can do so without interfering with a pocket wall. In this case, the control will raise the cutter the clearance amount (e) and move it to the new position. The cutter is then lowered to the current plunge-level at the plunge feedrate.

The area within one cutter radius of the current cutter position will be machined-out in a series of straight-line step-over roughing passes. The orientation of these passes, either parallel to the X or the Y axis, is dependent upon the pocket contour. The control determines whether passes parallel to the X or to the Y axis will machine-out the most material in the fewest amount of passes and orients the passes accordingly (R through X in Figure 25.4).

Figure 25.4
Roughing-Out Non-Adjacent Areas in an Irregular Pocket



Once the current plunge-level has been machined-out , the cutter is moved back to the start-point. The control raises cutter to the initial Z level then moves it to the start-point.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

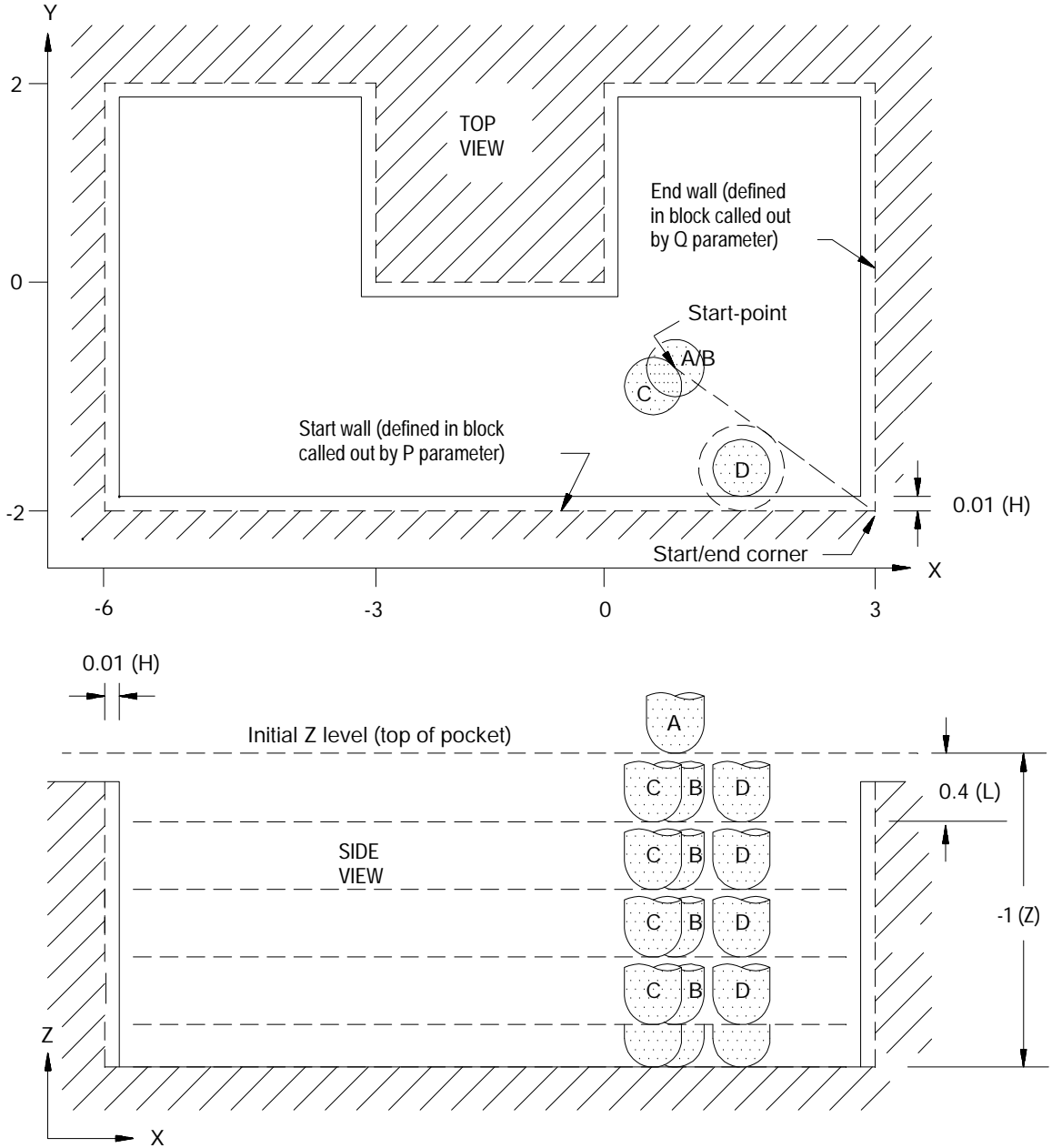
Once the programmed depth is reached, the control raises the cutter to the initial Z level then moves it to the start-point. This completes the irregular pocket roughing cycle. Example 26.1 shows an irregular pocket roughing cycle.

Example 26.1
Irregular Pocket Roughing Cycle

Program Block

```
N46 G92 X1 Y-1 Z0;  
N47 G10 L12 P1 R.125;  
N48 G90 G42 D1;  
N49 G89.1 X3 Y-2 Z-1 P50 Q57 H.01 E5 F100 L.4;  
N50 X-6;  
N51 Y2;  
N52 X-3;  
N53 Y0;  
N54 X0;  
N55 Y2;  
N56 X3;  
N57 Y-2;  
N58 M30;
```

Figure 25.5
Results of Example 26.1



25.1.2
Irregular Pocket Finishing
(G89.2)

Use the irregular pocket milling finishing cycle (G89.2) to finish an irregular pocket in a workpiece. This cycle is typically used to finish an irregular pocket formed using a G89.1 irregular pocket roughing cycle. A tool change is usually performed between the G89.1 and G89.2 cycles. You can use this cycle to finish a post that was formed by combining two pocket cycles.

The G89.2 block has this format:

G89.2 X_Y_Z_P_Q_H_F_L_;

Where :	Is :
X Y	The coordinates that specify the start/end corner of the irregular pocket in the selected plane. These parameters must be programmed.
Z	The coordinate (along the plunging axis) that specifies the bottom of the irregular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start-point to the pocket bottom. This parameter must be programmed.
P	The sequence number of the first block in the set of blocks that define the pocket shape. The coordinates in this block specify the start-point of the pocket's first wall. This start-point must correspond to the start/end corner specified by the X and Y coordinates. This parameter must be programmed.
Q	The sequence number of the last block in the set of blocks that define the pocket shape. The coordinates in this block specify the endpoint of the pocket's last wall. This endpoint must correspond to the start/end corner specified by the X and Y coordinates. This parameter must be programmed. The set of blocks specified by P and Q may be located anywhere after the calling block (even after an end of program command), as long as the calling block is in the same program as the set of blocks. This means that blocks defining the pocket shape can not be called from a subprogram or a macro unless the calling block is in that subprogram or macro.
H	The finish allowance that will be left on the sides of the irregular pocket. This is an optional parameter that provides for multiple finishing cycles.
F	Finishing feedrate. This parameter determines the feedrate of any XY axis cutting moves. If not programmed the existing (modal) cutting feedrate will be used.
L	Incremental plunge depth of each cutting pass along the Z axis. If L is programmed, a finish pass is made at each L level. If L is not programmed, only one finishing pass is made at the programmed Z depth. This is an optional parameter. It is typically programmed when a very deep pocket is being finished.

Before invoking the G89.2 cycle, the programmer must activate cutter compensation left or right by programming G41 or G42. This allows the control to begin interpreting the blocks that define the contour of the pocket as they are encountered.



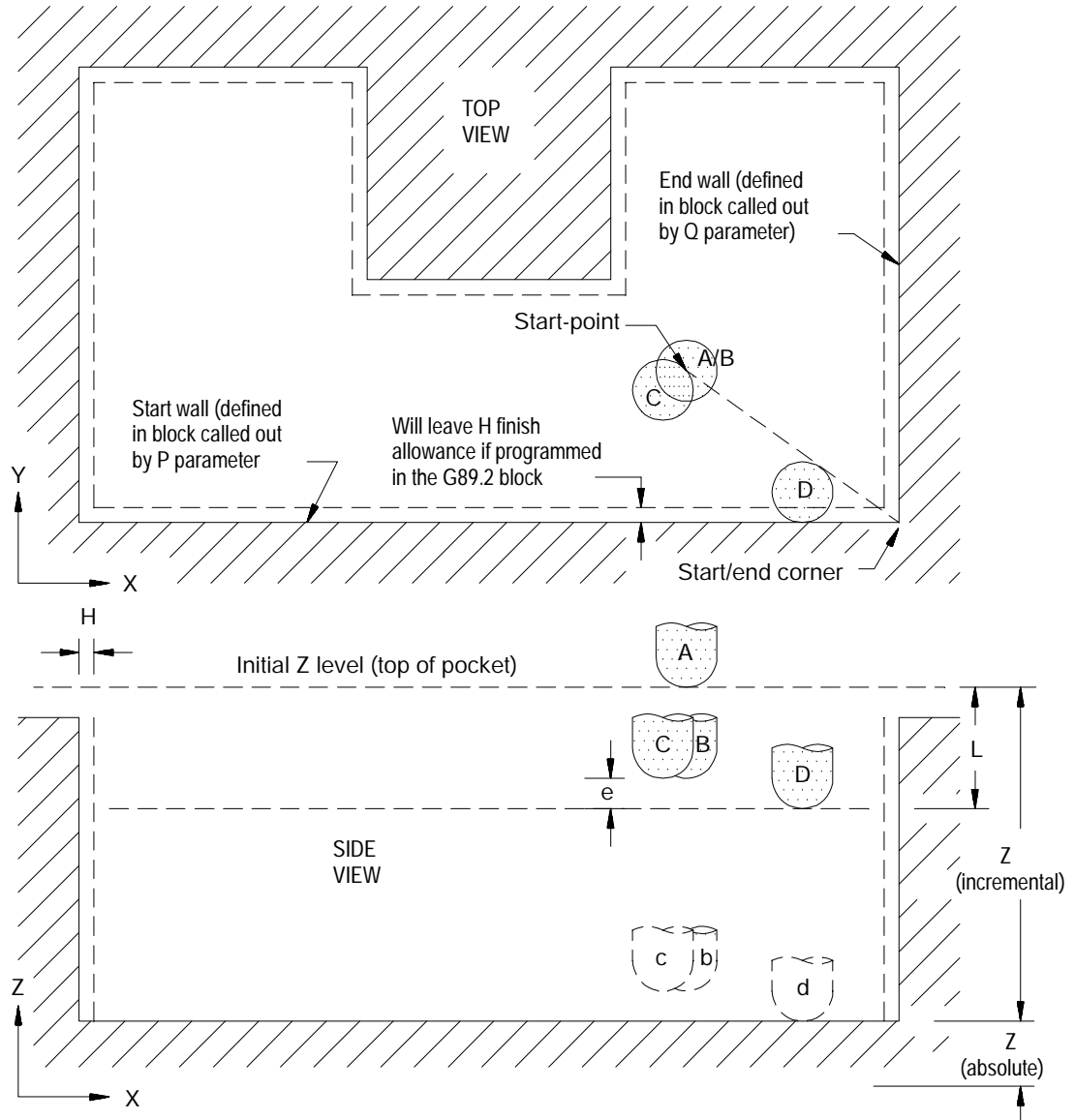
CAUTION: From the start-point the cutter must be able to move down into the part and then directly over to the start/end corner of the pocket (A through D in Figure 25.6) without cutting into any wall of the pocket.

Also, the cutter must be able to move from the end-point of the P block to the start-point (I through K in Figure 25.7) without cutting into any wall of the pocket.

The cutter should be positioned at the start-point as described for the G89.1 cycle (A in Figure 25.6). Once the axes are positioned at the start-point, a plunge along the Z axis takes place. If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter minus the clearance amount (B in Figure 25.6). If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter minus the clearance amount (B in Figure 25.6). The plunge takes place at the rapid feedrate.

After the plunge, the control uses the active cutter compensation mode (G41/G42) to offset the cutter one cutter radius perpendicular to the line from the start-point to the start/end corner (C in Figure 25.6). The control then moves the cutter from this offset position to a point where it is located one cutter radius plus the finish allowance (if H is programmed) away from both the first wall of the pocket and the line from the start-point to the start/end corner. At the same time, the cutter is lowered the clearance amount along the Z axis (D in Figure 25.6). This move takes place at the finishing feedrate.

Figure 25.6
Irregular Pocket Finishing Cycle Entry Moves

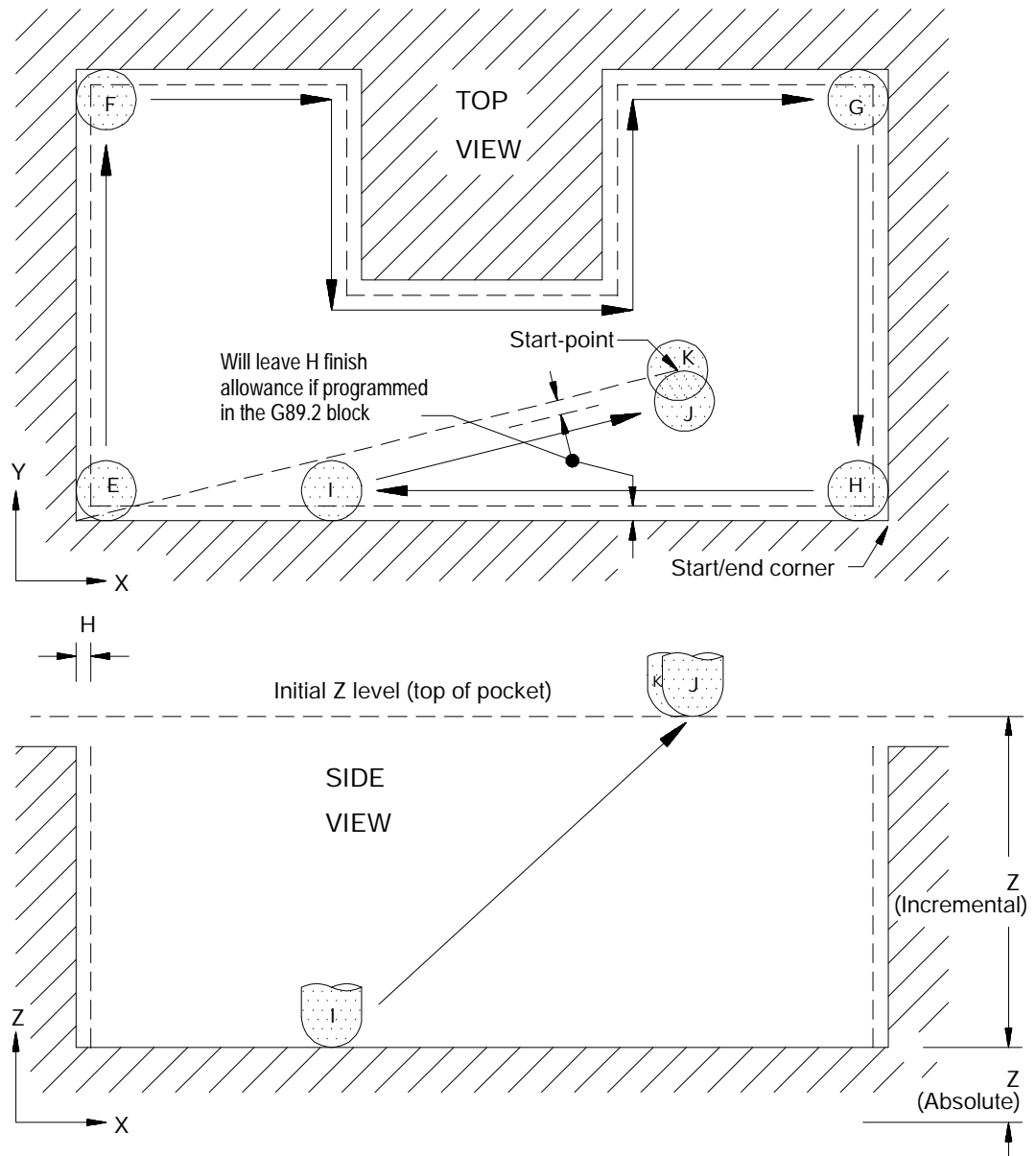


If H is programmed in the G89.2 block, an initial finish pass that leaves a finish allowance is made around the sides of the pocket. This typically would be done when it is desirable to clean out the corners in an irregular pocket before actually removing the finish allowance.

If H is not programmed in the G89.2 block, a finish pass is made around the sides of the pocket that removes all the finish allowance that was left during a G89.1 roughing cycle.

The finish pass ends at a point along the start-wall that is determined by the angle formed by the start-wall and a line drawn from the endpoint of the start-wall to the start-point. An example of this is shown in the following figure. From this point the cutter is moved back to the start-point of the cycle.

Figure 25.7
Irregular Pocket Finishing Cycle Exit Moves





CAUTION: The cutter must be able to move from the end-point of the P block to the start-point (I through K in Figure 25.7) without cutting into any wall of the pocket.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then finished as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

Once the programmed depth is reached, the control simultaneously raises the cutter and moves it towards the start-point. The control shuts off cutter compensation prior to reaching the start-point (J and K in Figure 25.7). This completes the irregular pocket finishing cycle.

END OF CHAPTER

Milling Fixed Cycles

26.0 Chapter Overview

This chapter covers the G-word data blocks in the milling fixed-cycle group. The operations of the milling fixed cycles are explained in these sections:

Information on:	On page:
Milling Fixed Cycles	26-2
Positioning and Hole Machining Axes	26-4
Parameters	26-7
Milling Fixed Cycle Operations	26-8
Altering Milling Fixed Cycle Operating Parameters	26-38
Examples of Milling Cycles	26-40

For this chapter, as well as this manual, make the following assumptions:

- **X and Y axes are the positioning axes (G17 plane).**
- **Z axis is the hole machining axis for drilling, boring, and tapping applications.**

26.1 Milling Fixed Cycles

Milling fixed cycles (sometimes referred to as canned cycles or autocycles) repeat a series of basic machining operations, such as, boring, drilling or tapping. These operations, designated by a single block command, usually consist of a fixed series of steps that are dependent on the type of machining application.

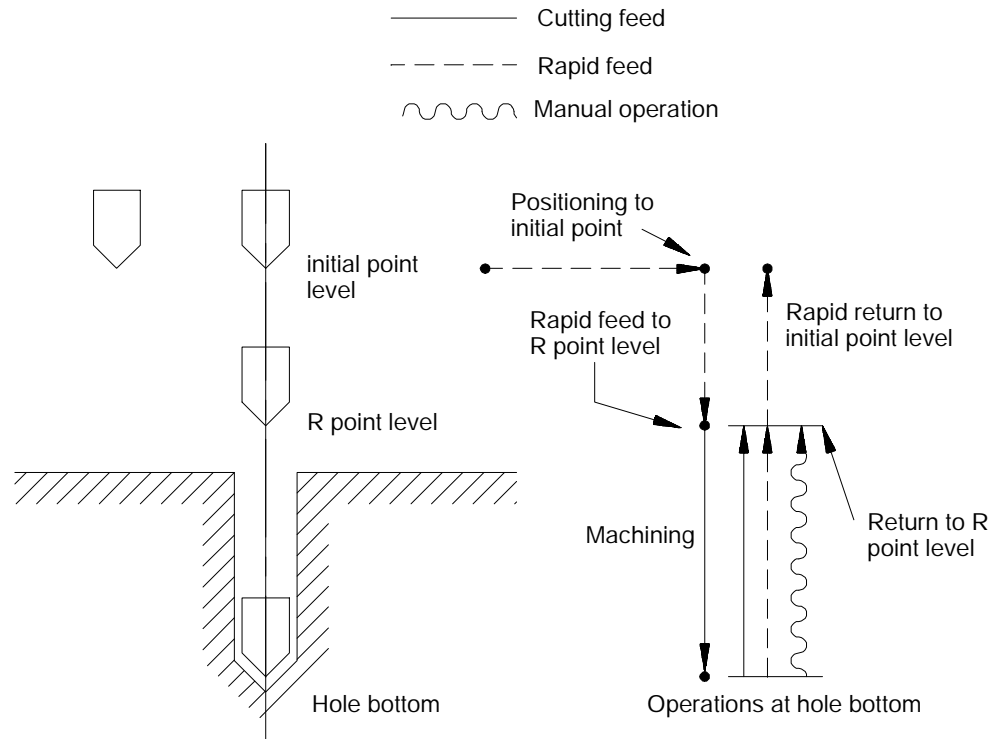
The control provides the milling fixed cycles shown in Table 26.A.

Table 26.A
Milling Fixed Cycles

G-code	Application	Tool Movement	Operation At Hole Bottom	Retraction Movement
G73	Deep Hole Peck Drilling Cycle with Dwell	Intermittent Feed	Retract	Rapid Traverse
G74	Left-Hand Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G74.1	Left-Hand Solid-Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G76	Boring Cycle, Spindle Shift	Feed	Oriented Spindle Stop / Retract	Rapid Traverse
G80	Cancel Or End Fixed Cycle	N/A	N/A	N/A
G81	Drilling Cycle, No Dwell/Rapid Out	Feed	Retract	Rapid Traverse
G82	Drilling Cycle, Dwell/Rapid Out	Feed	Dwell /Retract	Rapid Traverse
G83	Deep Hole Drilling Cycle	Intermittent Feed	Retract	Rapid Traverse
G84	Right-Hand Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G84.1	Right-Hand Solid-Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G85	Boring Cycle, No Dwell/Feed Out	Feed	Retract	Feed
G86	Boring Cycle, Spindle Stop/Rapid Out	Feed	Spindle Stop /Retract	Rapid Traverse
G87	Back Boring Cycle	Feed	Oriented Spindle Stop / Retract	Rapid Traverse
G88	Boring Cycle Spindle Stop/ Manually Out	Feed	Dwell / Retract Spindle Stop / Retract	Manual/Rapid Traverse
G89	Boring Cycle, Dwell/Feed Out	Feed	Dwell / Retract	Feed

In general, milling fixed cycles consist of the following operations (see Figure 26.1):

Figure 26.1
Milling Fixed Cycle Operations



The system installer determines if the positioning to initial point is always a rapid move, or if it is necessary to program a G00 or G01 to select a mode. This manual assumes rapid positioning.

26.2
Positioning and Hole
Machining Axes

This section assumes that the programmer can determine the hole machining axis using the plane select G-codes (G17, G18, and G19). Refer to the system installer's documentation to make sure that a specific axis has not been selected in AMP to be the hole machining axis.

G-codes, G17, G18 or G19, determine the plane, the positioning axes and the hole machining axis. The two axes that define the selected plane are used as positioning axes; the axis perpendicular to the plane is the hole machining axis.

Table 26.B below assumes that the system installer has not altered the default values defining the G17, G18 or G19 plane select codes.

Table 26.B
Plane Selection vs Machining Axis

Plane	Hole Machining Axis	Positioning Axes
XY (G17)	Z axis or its parallel axis	X and Y axes or their parallel axes
ZX (G18)	Y axis or its parallel axis	Z and X axes or their parallel axes
YZ (G19)	X axis or its parallel axis	Y and Z axes or their parallel axes

Example 26.1 shows you how to change the hole machining axis to a parallel axis. A G80 should be executed to cancel any active milling mode, prior to changing the hole machining axis.

Example 26.1
Altering the Machining Axis to a Parallel Axis

Program Block	Comment
The W axis is parallel to the Z axis.	
G17;	XY plane active
G81X ___ Y ___ ;	Drilling cycle, Z is the hole machining axis
.	
.	
G80;	Cancel milling cycle mode
G81X ___ Y ___ W ___;	Drilling cycle, W is the hole machining axis
.	W must be programmed in every subsequent block to remain the drilling axis. If it is not programmed, Z becomes the drilling axis.
.	
.	

The plane selection codes (G17-G19) can be included in the milling fixed cycle block, or can be programmed in a previous block.

Figure 26.2 shows typical milling fixed cycle motions in absolute (G90) or incremental (G91) modes. Note the changes in how the R point and Z level are referenced.

Figure 26.2
Milling Fixed Cycle Parameters in G90 and G91 Modes

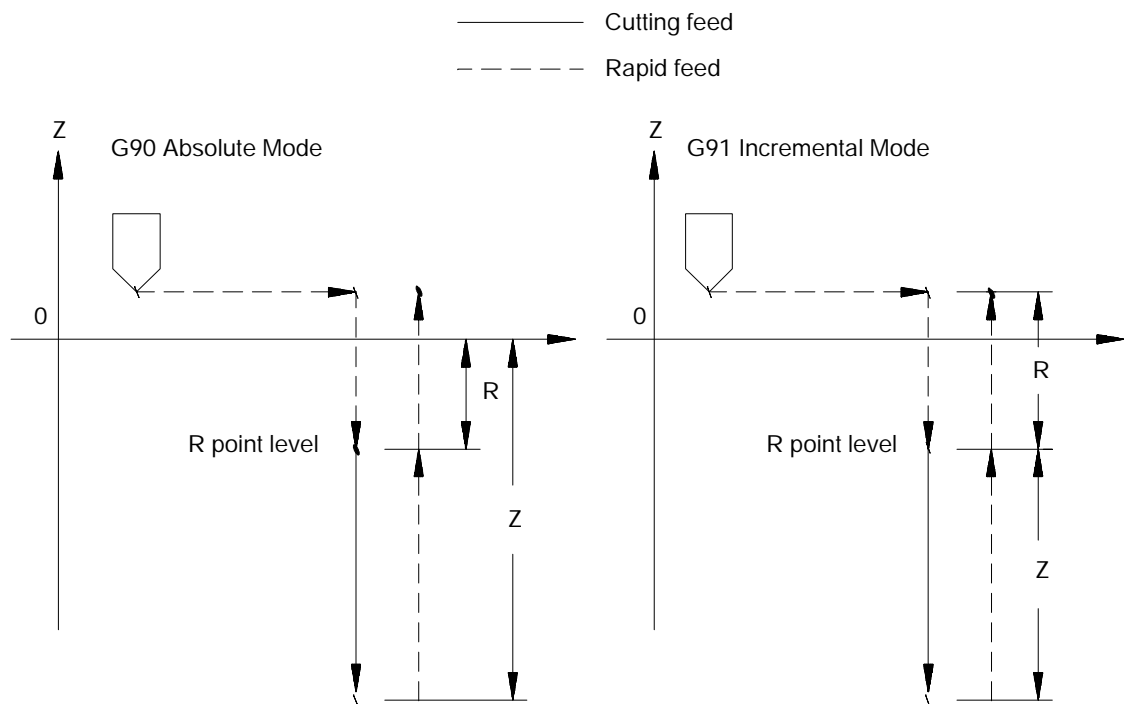
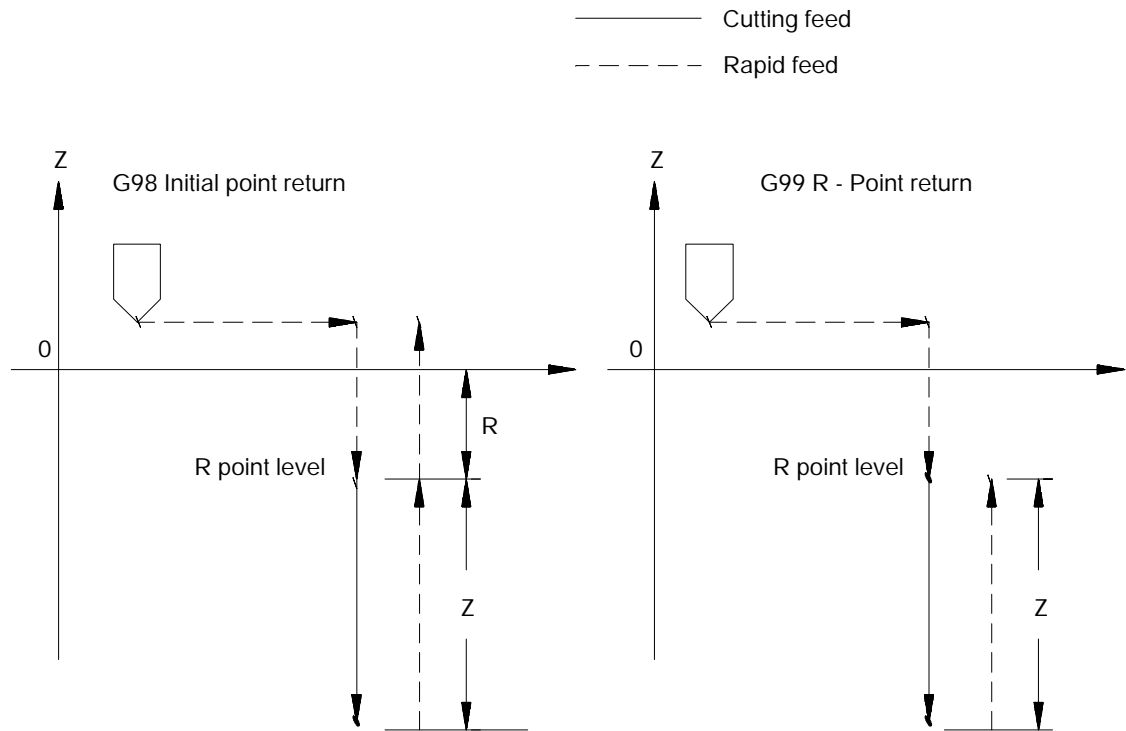


Figure 26.3 shows the two different modes available for selecting the return level in the Z axis after the hole has been drilled. These two modes are selected with G98 (which returns to the same level the cycle started at) and G99 (which returns to the level defined by the R point).

Figure 26.3
Milling Fixed Cycle Parameters in G98 or G99 Modes



Important: In the following sections, figures and examples are assumed to be programmed in the incremental mode (G91) and initial point return mode (G98).

26.3
Parameters

The following section provides a detailed explanation of each parameter that can be programmed for the milling fixed cycles. Some of these parameters are not valid with all cycles. Refer to the specific description of each cycle in section 26.4. To alter milling cycle operation parameters, refer to section 26.5.

We describe these milling fixed-cycle parameters below.

$$X_Y_Z_R_ \left\{ I_J_K_ \right\} P_F_L_Q_D_S_;$$

Where :	Is :
X, Y	specifies the location of the hole position in the selected plane. In the absolute mode (G90), program the hole position using the coordinate values in the active coordinate system. In incremental mode (G91), program the hole position using the distance from the current tool position to the required hole position.
Z	defines the hole bottom. In absolute mode (G90), program the hole bottom level using the coordinate value in the active coordinate system. In the incremental mode (G91), program the distance from the R point level to the hole bottom level.
R	defines the R point level. In the absolute mode (G90), program the R point level as a coordinate value in the active coordinate system. In the incremental mode (G91), program the R point level by the distance from the initial point level to the R point level.
I, J, K, or Q	Q defines the infeed amount for each move made in the hole in G83; I, J, and K, or Q defines the shift amount for G76 and G87.
P	defines the dwell period at hole bottom. P programs the dwell in the same way as G04: seconds if in feedrate mode (G94), spindle revolutions if in revolution mode (G95). (The allowable dwell time range in seconds is 0.001-99999.99. The allowable dwell range in revolutions is also 0.001-99999.999.) The P-word does not apply in all milling fixed cycles.
F	defines the cutting feedrate. If this parameter is not specified, the control will use the currently active feedrate for the cutting feedrate. For G74.1 and G84.1, F = tap thread lead in inches/mm per revolution.
L	defines the number of times the milling fixed cycle is repeated. The maximum number of repeats is 9999. <ul style="list-style-type: none"> In absolute mode, the control drills in the same location the number of times specified by the L-word. In incremental mode, the L-word drills the number of holes specified by the L-word at equally spaced positions, determined by axis positioning parameters X and Y. If an L0 is programmed, the control stores the milling cycle information but does not execute the drilling cycle. If no L-word is programmed, the control defaults to L1.
Q	In G83, Q defines the infeed amount for each move made in the hole. In G86.1 and G87, Q defines the shift amount (as do I, J, and K). In G74.1 and G84.1, Q defines the angle at which to orient the spindle before starting the tap. If you don't program the Q-word, the spindle is not oriented before the tap begins. This means that the hole is not re-tappable unless a Q-word is programmed in the cycle block. The spindle is brought to a stop prior to the initiation of the tapping phase even if Q is <u>not</u> programmed; this happens after the move to the R-plane.
D	defines the return spindle speed so that, if you want, the tap-out move can be performed faster or slower than the tap-in. Tool selection by D-word is not possible while in the solid tapping mode.
S	defines spindle speed in rpm.

Important: After programming a milling fixed cycle block, parameters X, Y, Z and R can be programmed in later blocks with different values. This, of course, permits axis motion to be changed. Parameters Q, P, I and K can only be programmed in the calling block for the milling fixed cycle. They cannot be programmed following the calling block. If they are, the control will ignore them.

26.4 Milling Fixed Cycle Operations

This section describes how the control executes each milling fixed cycle. The following is assumed for each cycle:

- initial point level is the return level (G98 is active)
- incremental mode is active (G91 is active)
- the X and Y axes are the positioning axes
- the Z axis is the hole machining axis

The milling fixed cycles are modal, which means they remain active until a G-code that cancels the milling fixed cycle is programmed. A milling fixed cycle can, therefore, be repeated at different positions, without having to re-program all the parameters associated with a given operation.



CAUTION: The controlling spindle code determines which spindle and its related spindle M-codes (modal) will be active during milling cycles. When spindle is mentioned in relation to milling fixed cycles, we are referring to the controlling spindle. For more information on controlling spindles, refer to chapter 17.

Similarly, any parameters specified in the block with the G-code of the milling fixed cycle remain active until the cycle is cancelled, or until they are programmed again in a following block. L-words do not remain active and, instead, are active only for the block which contains the actual L-word.

G00, G01, G02, and G80 will cancel milling fixed cycle modes.

(G73): Deep Hole Peck Drilling Cycle with Dwell

- The format for the G73 cycle is as follows:

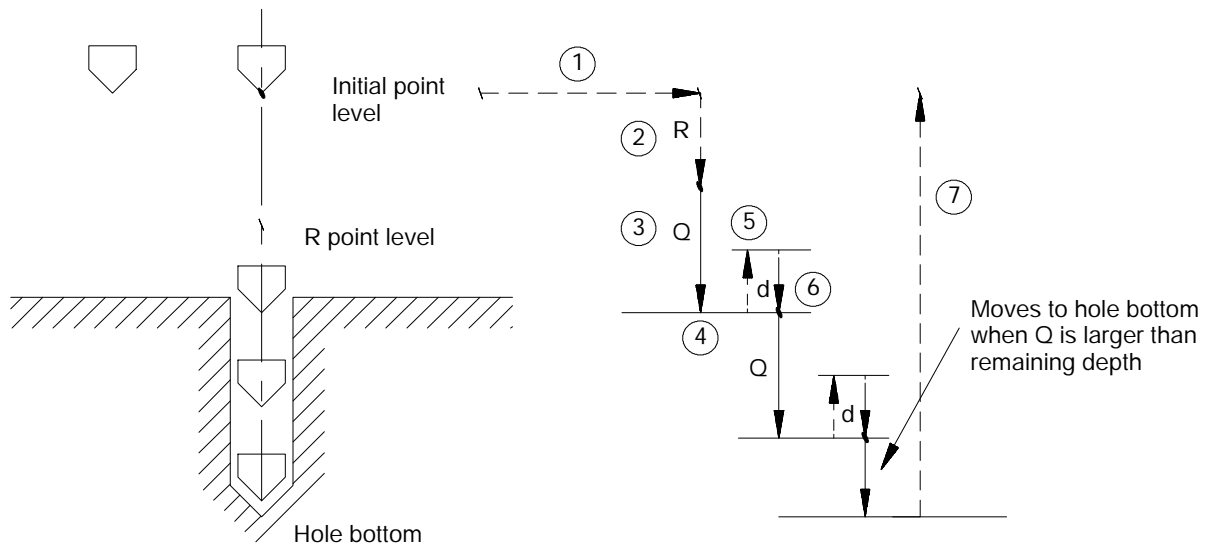
G73X_Y_Z_R_Q_P_F_L_;

Where :	Is :
X, Y	specifies the location of the hole position in the selected plane.
Z	defines the hole bottom.
R	defines the R point level.
Q	defines the infeed amount for each step into the hole.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.4
G73: Deep Hole Peck Drilling Cycle with Dwell



In the G73 peck drilling cycle, the control moves the axes in this manner:

1. The tool rapids to the initial point level above the hole location.
2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the drilling operation.
3. During the drilling operation, the control infeeds the drilling tool by an amount Q, as programmed in the drilling cycle.

4. If a value was programmed for the P parameter, the drilling tool will dwell after it reaches the bottom of the hole.
5. It then retracts by an amount d at a rapid feedrate. The amount d is specified by the system installer, or can be set by the operator as described in section 26.5. This intermittent feed simplifies chip disposal and lets a small retraction amount to be set in peck drilling.
6. After the drilling tool retracts an amount d, it then resumes drilling at the cutting feedrate to a depth d + Q.
This retraction and extension continues until the drilling tool reaches the depth of the hole as programmed with the Z-word in the drilling cycle block.
7. The drilling tool then retracts at a rapid feedrate to the initial point level as determined by G98.

When the single block function is active, the control stops axis motion and awaits “cycle start” after steps 1, 2 and 7.

(G74):Left-Hand Tapping Cycle

This cycle is used to cut left-handed threads.



CAUTION: The programmer or operator must set the direction of spindle rotation for tap-in. The control forces the proper spindle direction for the tap-out, but uses the programmed spindle direction for the tap-in.

The format for the G74 cycle is as follows:

G74X_Y_Z_R_P_F_L_;

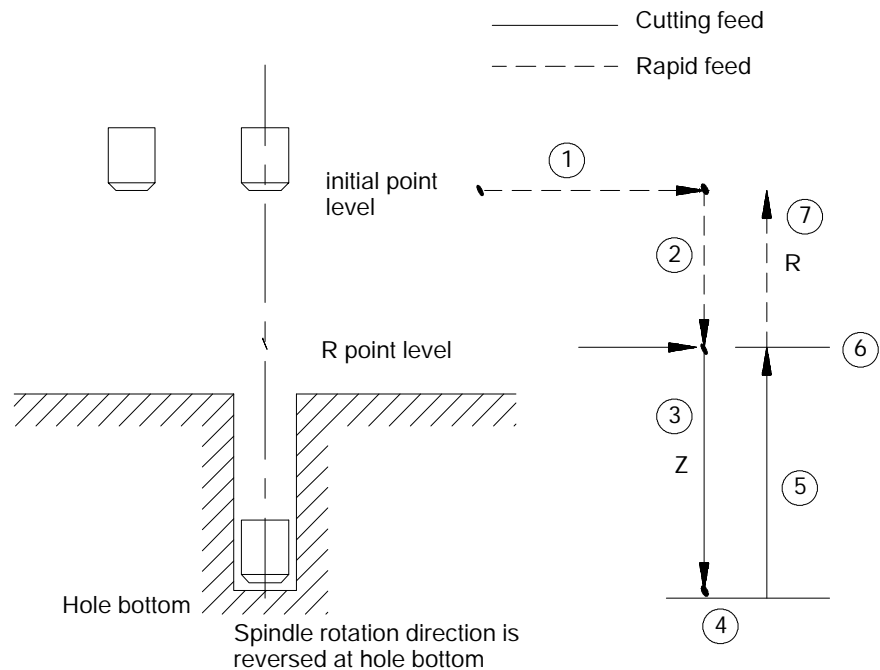
Where :	Is :
X, Y	specifies the location of the hole position in the selected plane.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the tapping feedrate. This should be programmed as close as possible to the rate in which the tap will be moving into the part (calculated from the tap thread pitch and the active spindle speed). Enter the feedrate in either IPM or IPR modes. No special spindle synchronization occurs with this cycle.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: When programming a G74 tapping cycle, consider this:

- The programmer or operator must start spindle rotation.
- Override usage- the control ignores the feedrate override switch and clamps override at 100 percent.
- During tapping the feedrate override switch, and the feedhold feature are both disabled. Cycle stop is not acknowledged until the end of the return operation.

Figure 26.5
G74: Left-Hand Tapping Cycle



In the G74 left-hand tapping cycle, the control moves the axes in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The threading tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the tapping operation.
3. During the tapping operation, the control infeeds the threading tool to the depth and at the feedrate programmed in the tapping cycle.

4. If a value was programmed for the P parameter, the threading tool dwells after it reaches the bottom of the hole, and after the spindle has been commanded to reverse.

The spindle reverses to the clockwise direction.

5. The threading tool retracts at the cutting feedrate to the R point.
6. If a value was programmed for the P parameter, the threading tool will dwell after it reaches the R point. (Dwells may be ignored if the system installer has chosen to do so in AMP.)

Then the spindle direction is reversed to counterclockwise.

7. With G98 active, the cutting tool will then accelerate to the rapid feedrate and retract to the initial point level.

When the single block function is active, the control stops axis motion and awaits “cycle start” after steps 1, 2 and 7.

If the operator activates a feedhold during steps 3, 4 or 5, axis motion stops after step 7. Axis motion will also stop during steps 1, 2 and 7. However, if feedhold is activated during step 7, axis motion will stop immediately.

Important: Your system installer can enable a tap retract feature for this cycle through PAL. Tap retract enables you to retract the tapping tool and resume the cycle, or completely abort the tapping operation. Refer to your system installers documentation for details.

(G74.1): Left-Hand
Solid-Tapping Cycle

Use this cycle to cut left-handed threads.

The format for the G74.1 cycle is:

```
G74.1X_Z_R_F_L_Q_D_S_;
```

Where :	Is :
X	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	represents the thread lead along the drilling axis (Z in this manual). It is mandatory and modal in any subsequent solid tapping cycle blocks until a new F-word is programmed. The control interprets the F-word as the thread lead in inches per revolution or millimeters per revolution, depending on the inch/metric mode active.
L	defines the number of times the drilling cycle is repeated.
Q	defines the angle at which to orient the spindle before starting the tap. It is modal in any subsequent solid tapping blocks until a new Q-word is programmed or the tapping cycle is cancelled by a G80. To retap a hole, a Q-word must have been programmed when the hole was originally tapped.
D	defines the return spindle speed, but cannot exceed the maximum tapping spindle speed set in AMP. This will adjust your Z-axis feedrate according to the thread lead defined in F.
S	defines spindle speed in rpm.

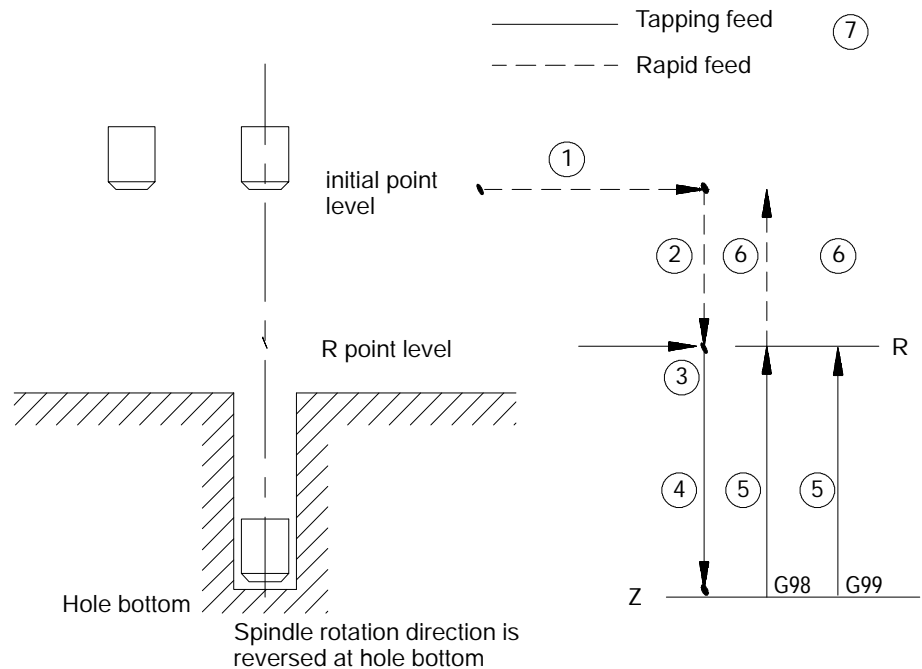
See section 26.3 for a detailed description of these parameters.

Important: When programming and executing a G74.1 solid tapping cycle, remember:

- the feedrate of the tapping phases is derived as
 $(\text{spindle } \{RPM\} * \text{F-lead } \{IPR\}) = \text{IPM}$
- the spindle speed that is active at the start of the cycle determines the effective Z feedrate
- the direction of spindle rotation for tap-in and tap-out phases will be automatically generated by the control
- spindle speed override has no effect on the solid tapping cycle; you can use feedrate override to adjust the tapping operation
- D cannot exceed the maximum tapping spindle speed set in AMP
- you cannot select tools via D-word while in solid tapping mode
- gear changes are locked out
- cycle stop is acknowledged throughout the cycle, but can be disabled by G63
- you can use active reset to abort the cycle after the cycle stop request has been acknowledged

- to re-tap a hole, a Q-word must have been programmed when the hole was originally tapped
- block retrace is possible during the tap-in portion of the cycle, but not during the tap-out

Figure 26.6
G74.1: Left-Hand Solid-Tapping Cycle



In the G74.1 left-hand solid-tapping cycle, the control moves the axes in this manner:

1. The tool rapids to the tapping position above the hole location.
2. The threading tool then rapids to the R point.
3. The control either orients or stops the spindle.

If a Q-word was programmed:	the control:
yes	orients the spindle
no	stops the spindle

4. Tap-in: The counterclockwise rotation of the spindle ramps up to the programmed S spindle speed and linear motion of the Z axis moves synchronously to reach the Z position.

5. **Tap-out:** The spindle and linear motion reverse to the clockwise direction and retract to the R point.

The tap-out speed is determined by $F * S$ unless you programmed D (tap-out rpm), in which case tap-out speed is $F * D$.

At the R point, spindle rotation has ramped to zero.

6. With G98 active, the cutting tool then accelerates to the rapid feedrate and retracts to the initial point level.

With G99 active, the cutting tool remains at R point; no movement occurs.

In single-block mode, the control stops axis motion after phases 1, 2, 3, and 6 of the cycle (Figure 27.M1).

Important: When it is active, S-Curve Acc/Dec (G47.1) will be applied to the rapid feedrate portions of the solid-tapping cycle.

(G76): Boring Cycle,
Spindle Shift

The format for the G76 cycle is as follows:

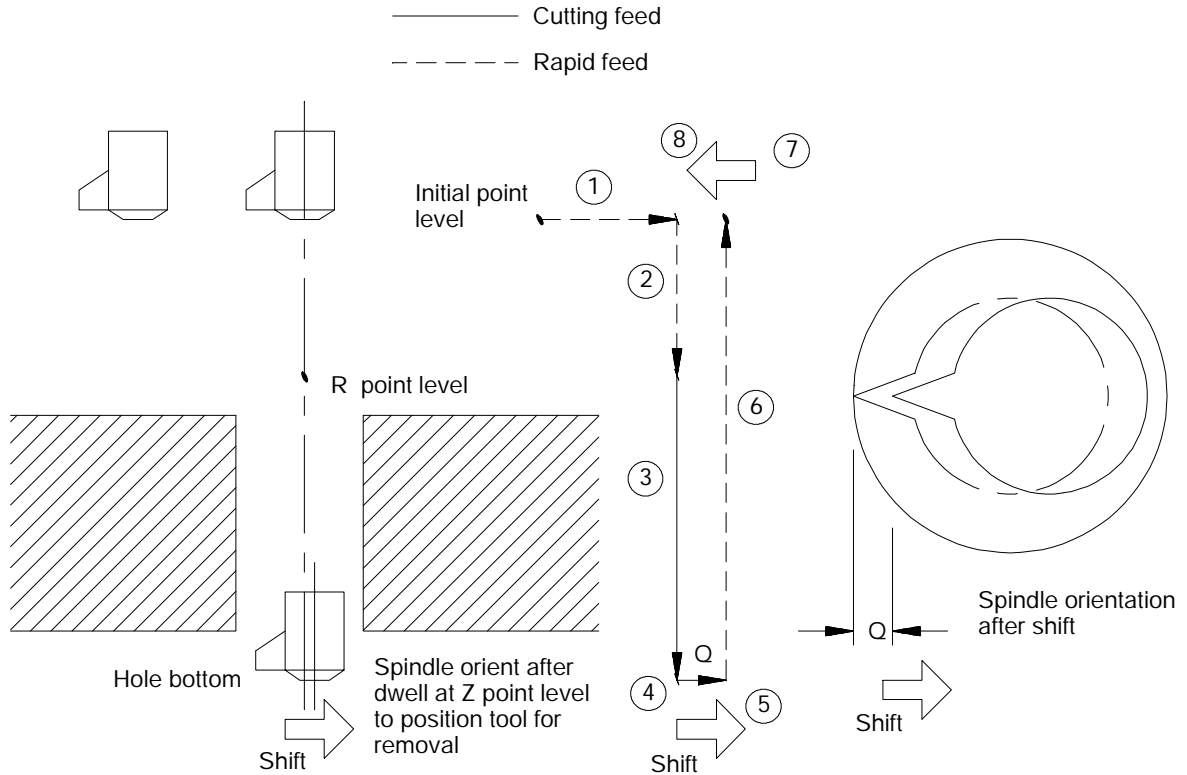
G76X__Y__Z__ { I__J__K__ } R__F__L__;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
Q or I, J, K	defines the tool shift amount and direction.
R	defines the R point level.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.7
G76: Boring Cycle, Spindle Shift



In the G76 boring cycle, the control moves the axes in this manner:

1. The tool rapids to the initial point level above the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool bores at the programmed feedrate to the pre-programmed depth of the hole (defined by the Z-word in the boring cycle block).
4. The control positions the spindle at the bottom of the hole in a particular orientation, determined by the system installer in AMP.
5. To prevent the boring tool from damaging the sides of the bored hole during retraction, the boring tool is shifted in either of two methods, which are explained on the following page and illustrated in Figure 26.7.

The shift direction is determined by two possible methods.

Method I

This shift method is a single axis shift. The direction and axis for the shift is set in AMP by the system installer or can be altered using the milling fixed cycle parameter table (see section 26.6).

- The direction of the axis is specified as + or -.
- The feedrate using this shift method is always rapid traverse.
- The Q-word shift amount is always interpreted as a positive value. A negative Q-word is not allowed.

Method II

The direction of the shift using this method is programmed in the boring cycle block. Program a shift amount for axes in the current plane only (determined by G17, G18, or G19) using the following words:

I__ programs a X axis move.
J__ programs a Y axis move.
K__ programs a Z axis move.

Follow the I-, J- and K-words (modal during milling fixed cycles) with incremental values in the block that programs the hole position.

When using Method II, remember:

- If both axes in the current plane are to be shifted, specify both words to move the axes.
 - The move generated will be a single linear move and will execute at rapid traverse.
6. The boring tool is then retracted at a rapid feedrate to the initial point level as determined by G98.
 7. After reaching the initial point level, the boring tool is shifted back (in a manner previously explained and illustrated) and the spindle is re-started in the counterclockwise direction again.

When the single block function is active, the control stops axis motion after steps 1, 2 and 8.

(G80): Cancel or End Fixed Cycles

The format for the G80 cancel or end fixed cycles is as follows:

G80;

Programming a G80 cancels the currently active milling fixed cycle mode. (G00, G01, G02, or G03 will also cancel any active milling fixed cycle.)

If milling fixed cycles are canceled with a G80, program execution returns to the mode which was in effect when the cycles were last turned on, for example, G00 - G03.

(G81): Drilling Cycle, No Dwell/Rapid Out

The format for the G81 cycle is as follows:

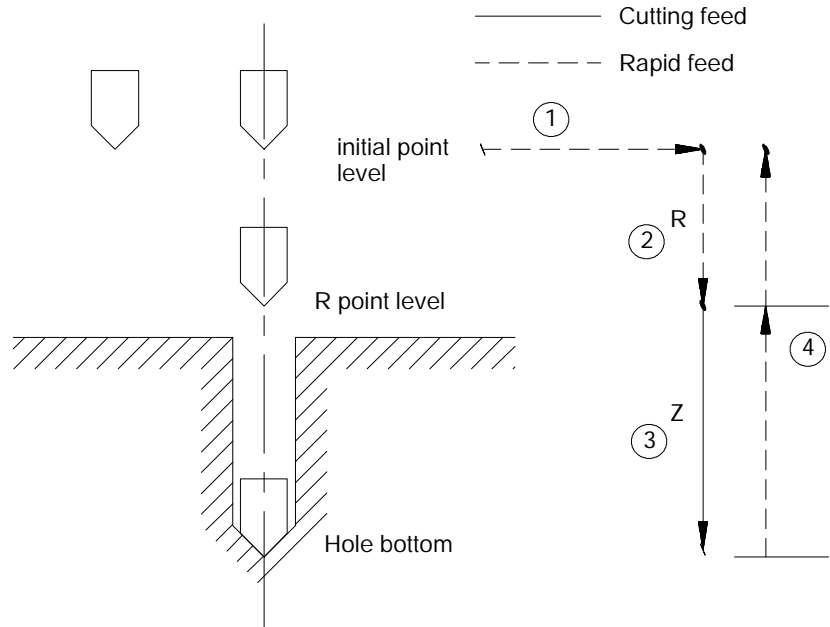
G81X_Y_Z_R_F_L_;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.8
G81: Drilling Cycle without Dwell



In the G81 drilling cycle, the control moves the axes in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the drilling operation.
3. The drilling tool continues to drill at the programmed feedrate until it reaches the depth of the hole as programmed with the Z-word.
4. The control retracts the drilling tool at a rapid feedrate to the initial point level as determined by G98.

When the single block function is active, the control stops axis motion after steps 1, 2 and 4.

(G82): Drill Cycle,
Dwell/Rapid Out

The format for the G82 cycle is as follows:

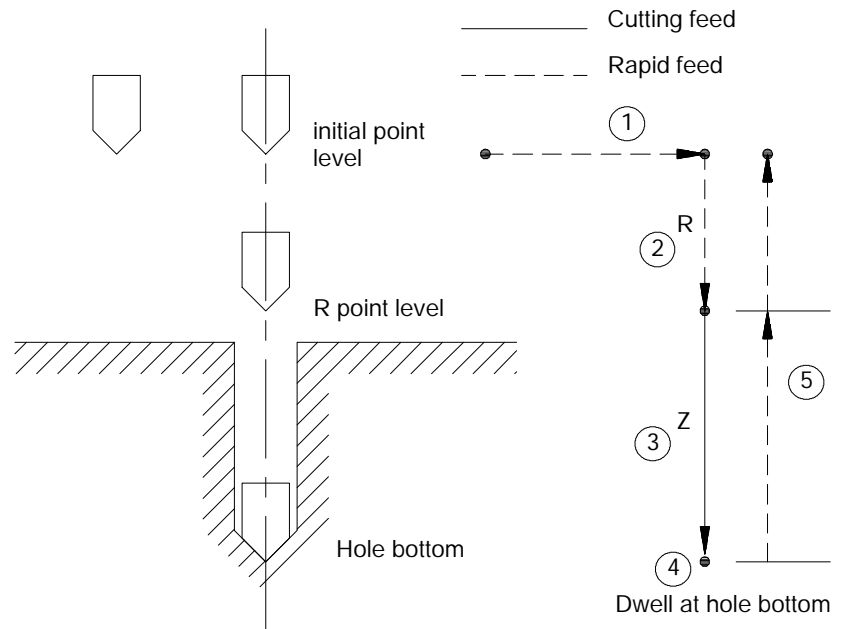
G82X_Y_Z_R_P_F_L_;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed explanation of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.9
G82: Drilling Cycle, Dwell/Rapid Out



In the G82 drilling cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level point above the hole location.
2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the drill operation.
3. The cutting tool drills at the programmed feedrate to the pre-programmed depth of the hole (defined by the Z-word in the boring cycle block).
4. If a value was programmed for the P parameter, the drilling tool will dwell after it reaches the bottom of the hole.
5. After the drilling tool reaches the hole bottom and the dwell is completed, the drilling tool is retracted at a rapid feedrate to the initial point level as determined by G98.

When the single block function is active, the control stops axis motion after steps 1, 2 and 5.

(G83): Deep Hole Drilling Cycle

The format for the G83 cycle is as follows:

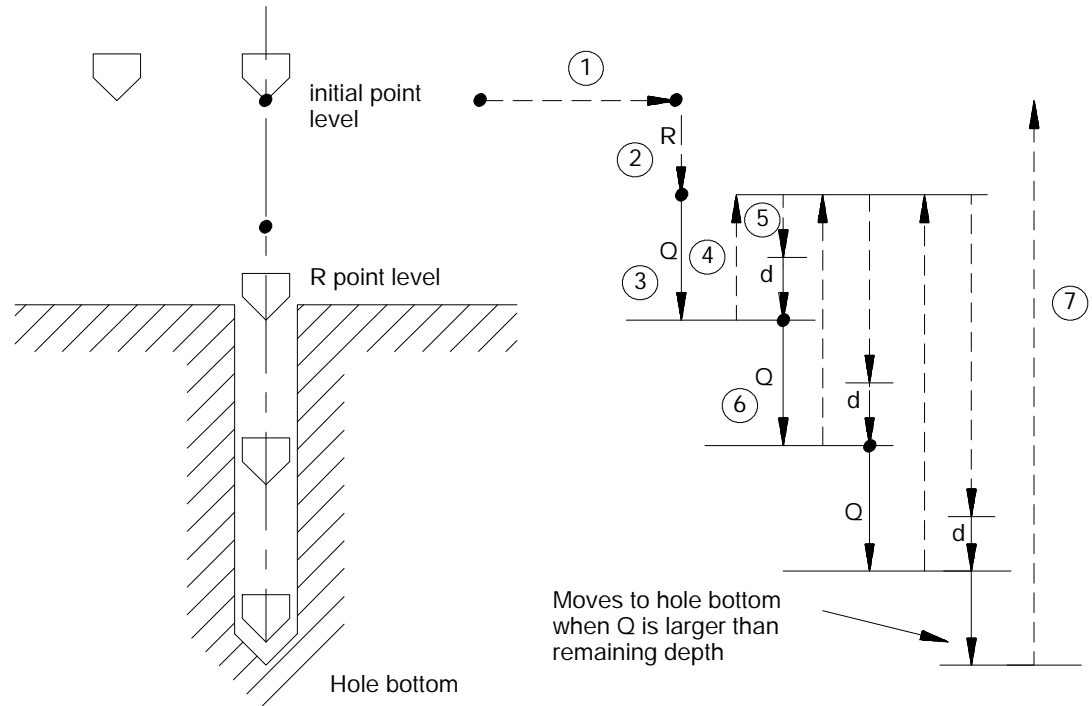
G83X_Y_Z_R_Q_F_L_;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
Q	..defines the infeed amount for each step into the hole.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.10
G83: Deep Hole Drilling Cycle



In the G83 drilling cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level above the hole location.
2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the deep hole drilling operation.
3. During the drilling operation, the control infeds the drilling tool by an amount Q , as programmed in the G83 block.
4. The drilling tool retracts at a rapid feedrate to the R point level.
5. The control feeds the drilling tool at rapid feedrate to a distance d above the level drilled in the previous infeed. The amount d is specified by the system installer, or can be set by the operator as described in section 26.5. This intermittent feed simplifies chip disposal and permits a very small retraction amount to be set in deep hole drilling.
6. The drilling tool slows to the cutting feedrate again and infeds an amount $Q + d$.

7. The cutting tool is then retracted at a rapid feedrate to the initial point level as determined by G98.

When the single block function is active, the control stops axis motion after steps 1, 2 and 7.

(G84): Right-Hand Tapping Cycle

This cycle is used to cut right-handed threads. The format for the G84 cycle is as follows:

G84X_Y_Z_R_P_F_L_;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the tapping feedrate. This should be programmed as close as possible to the rate in which the tap will be moving into the part (calculated from the tap thread pitch and the active spindle speed). Enter the feedrate in either IPM or IPR modes. No special spindle synchronization occurs with this cycle.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

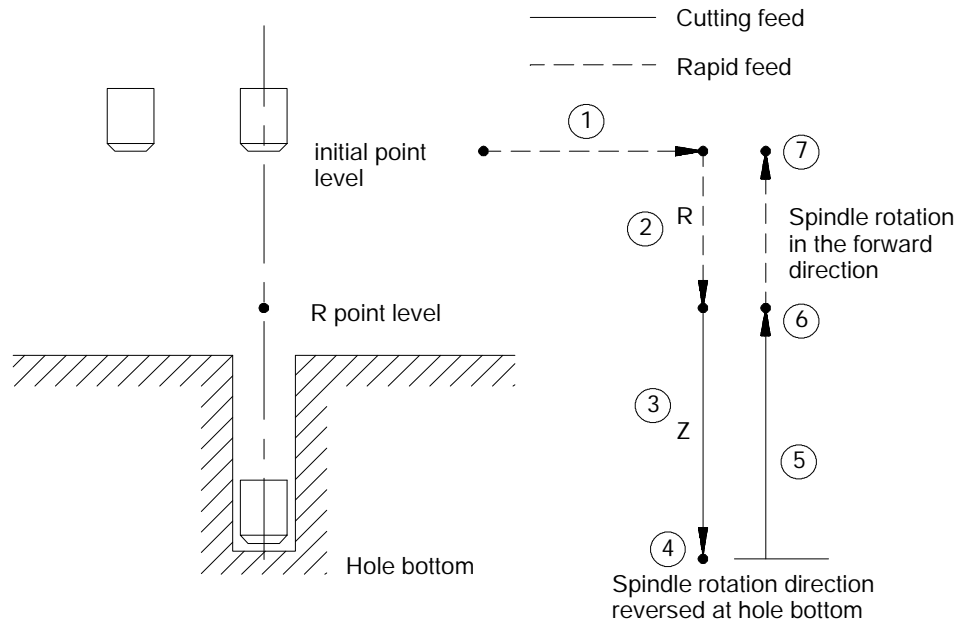


CAUTION: The programmer or operator must set the direction of spindle rotation for tap-in. The control forces the proper spindle direction for the tap-out, but uses the programmed spindle direction for the tap-in.

Important: When programming and executing a G84 tapping cycle, consider this:

- The programmer or operator must start spindle rotation.
- Override usage - the control ignores the feedrate override switch and clamps override at 100 percent.
- During tapping the feedrate override switch, and the feedhold feature are both disabled. Cycle stop is not acknowledged until the end of the return operation.

Figure 26.11
G84: Right-Hand Tapping Cycle



In the G84 right-hand tapping cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level above the hole location.
2. The threading tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the tapping operation.
3. During the tapping operation, the control infeeds the threading tool to the depth and at the feedrate programmed in the tapping cycle.
4. If a value was programmed for the P parameter, the threading tool will dwell after it reaches the bottom of the hole and after the spindle has been commanded to reverse.

The spindle reverses to the counterclockwise direction.

5. The threading tool retracts at the cutting feedrate to the R point.
6. If a value was programmed for the P parameter, the threading tool will dwell after it reaches the R point level. (Dwells may be ignored if the system installer has chosen to do so in AMP.)

Then the spindle direction is reversed to clockwise.

7. With G98 active, the cutting tool will then accelerate to the rapid feedrate and retract to the initial point level.

When the single block function is active, the control stops axis motion after steps 1, 2 and 6.

If the operator activates a feedhold during steps 3, 4 or 5, axis motion stops after step 7. Axis motion will also stop during steps 1, 2, and 7. However, if the operator activates a feedhold during step 7, axis motion will stop immediately.

Important: Your system installer can enable a tap retract feature for this cycle through PAL. Tap retract enables you to retract the tapping tool and resume the cycle, or completely abort the tapping operation. Refer to your system installers documentation for details.

(G84.1): Right-Hand
Solid-Tapping Cycle

Use this cycle to cut right-handed threads.

The format for the G84.1 cycle is:

G84.1 X_Z_R_F_L_Q_D_S_;

Where :	Is :
X	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	represents the thread lead along the drilling axis (Z in this manual). It is mandatory and modal in any subsequent solid tapping cycle blocks until a new F-word is programmed. The control interprets the F-word as the thread lead in inches per revolution or millimeters per revolution, depending on the inch/metric mode active.
L	defines the number of times the drilling cycle is repeated.
Q	defines the angle at which to orient the spindle before starting the tap. It is modal in any subsequent solid tapping blocks until a new Q-word is programmed or the tapping cycle is cancelled by a G80. To retap a hole, a Q-word must have been programmed when the hole was originally tapped.
D	defines the return spindle speed, but cannot exceed the maximum tapping spindle speed set in AMP. This will adjust your Z-axis feedrate according to the thread lead defined in F.
S	defines spindle speed in rpm.

See section 26.3 for a detailed description of these parameters.

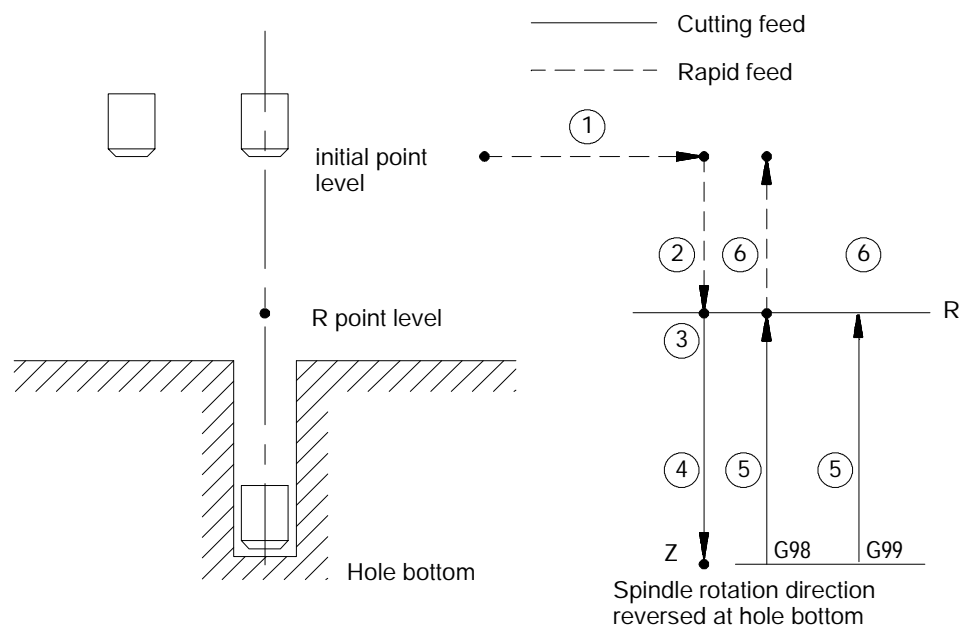
Important: When programming and executing a G84.1 solid tapping cycle, remember:

- the feedrate of the tapping phases is derived as

$$(\text{spindle \{RPM\}} * \text{F-lead \{IPR\}}) = \text{IPM}$$

- the spindle speed that is active at the start of the cycle determines the effective Z feedrate
- the direction of spindle rotation for tap-in and tap-out phases will be automatically generated by the control
- spindle speed override has no effect on the solid tapping cycle; you can use feedrate override to adjust the tapping operation
- D cannot exceed the maximum tapping spindle speed set in AMP
- you cannot select tools via D-word while in solid tapping mode
- gear changes are locked out
- cycle stop is acknowledged throughout the cycle, but can be disabled by G63
- you can use active reset to abort the cycle after the cycle stop request has been acknowledged
- to re-tap a hole, a Q-word must have been programmed when the hole was originally tapped
- block retrace is possible during the tap-in portion of the cycle, but not during the tap-out

Figure 27.M2
G84.1: Right-Hand Solid-Tapping Cycle



In the G84.1 right-hand solid-tapping cycle, the control moves the axes in this manner:

1. The tool rapids to the tapping position above the hole location.
2. The threading tool then rapids to the R point.
3. The control either orients or stops the spindle.

If a Q-word was programmed:	the control:
yes	orients the spindle
no	stops the spindle

4. Tap-in: The clockwise rotation of the spindle ramps up to the programmed S spindle speed and linear motion of the Z axis moves synchronously to reach the Z position.
5. Tap-out: The spindle and linear motion reverse to the counterclockwise direction and retract to the R point.

The tap-out speed is determined by $F * S$ unless you programmed D (tap-out rpm), in which case tap-out speed is $F * D$.

At the R point, spindle rotation has ramped to zero.

6. With G98 active, the cutting tool then accelerates to the rapid feedrate and retracts to the initial point level.

With G99 active, the cutting tool remains at R point; no movement occurs.

In single-block mode, the control stops axis motion after phases 1, 2, 3, and 6 of the cycle (Figure 27.M2).

Important: When it is active, S-Curve Acc/Dec (G47.1) will be applied to the rapid feedrate portions of the solid-tapping cycle.

(G85): Boring Cycle, No Dwell/Feed Out

The format for the G85 cycle is as follows:

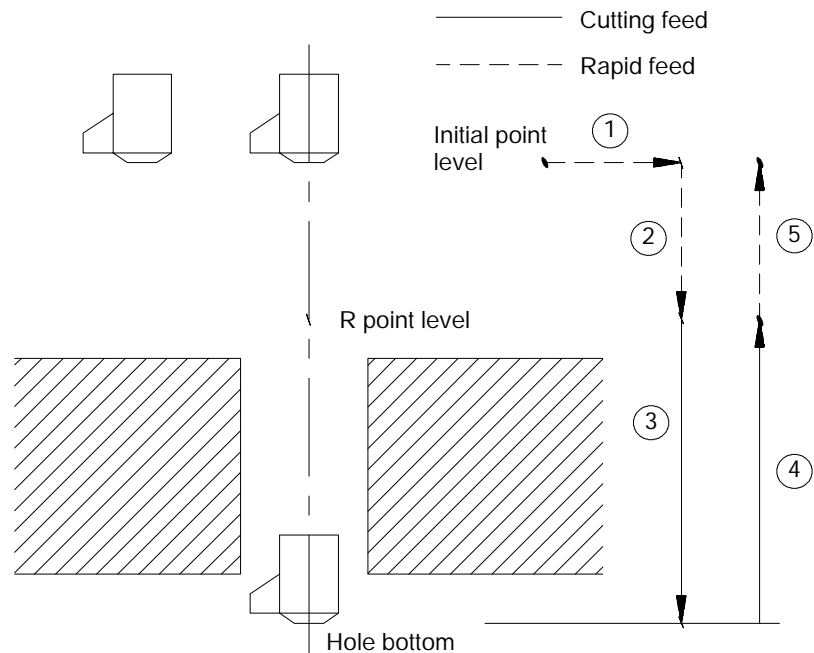
G85X_Y_Z_R_F_L_;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.12
G85: Boring Cycle (Without Dwell, Feed Out)



In the G85 right-hand tapping cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool continues to drill at the programmed feedrate until it reaches the depth of the hole as programmed with the Z-word.

4. The control retracts the boring tool at the cutting feedrate to the R point.
5. The control retracts the drilling tool at a rapid feedrate to the initial point level, as determined by G98.

When the single block function is active, the control stops axis motion after steps 1, 2 and 5.

(G86): Boring Cycle,
Spindle Stop/Rapid Out

The format for the G86 cycle is as follows:

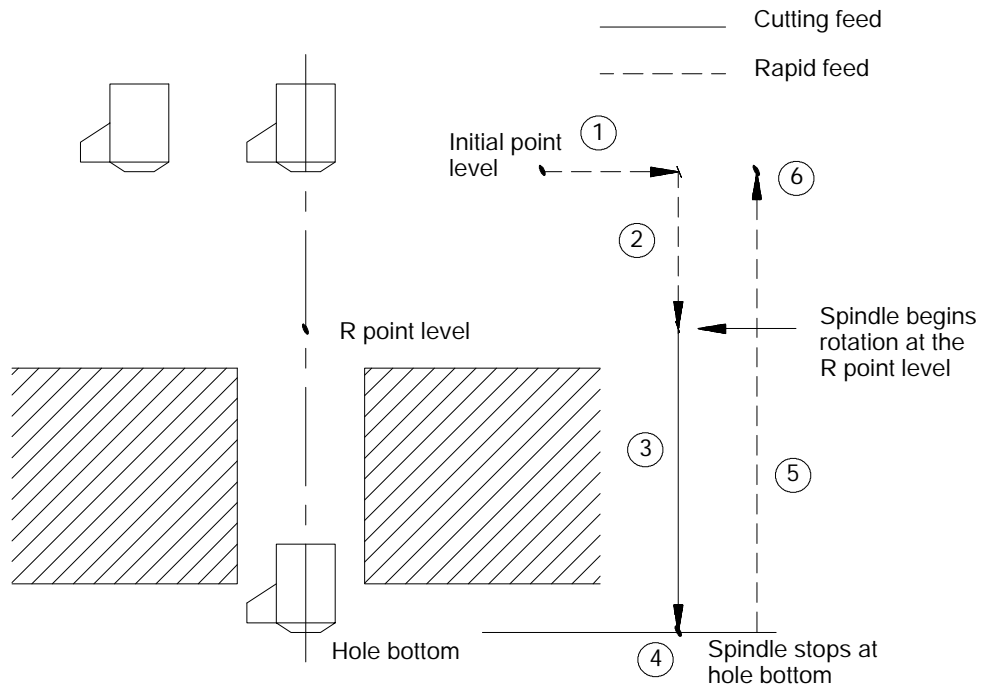
G86X_Y_Z_R_P_F_L_;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.13
G86: Boring Cycle, Spindle Stop/Rapid Out



In the G86 milling fixed cycle, the control moves the axis in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The cutting tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The cutting tool bores at the programmed feedrate until it reaches the depth of the hole as programmed with the Z-word.
4. If the user has entered a value for the P parameter, the cutting tool will dwell after it reaches the bottom of the hole.
5. The spindle stops rotating.
6. The boring tool is then retracted at a rapid feedrate to the initial point level, as determined by G98. Spindle rotation continues forward.

When the single block function is active, the control stops axis motion after steps 1, 2 and 6.

(G87): Back Boring Cycle

The format for the G87 back boring cycle is:

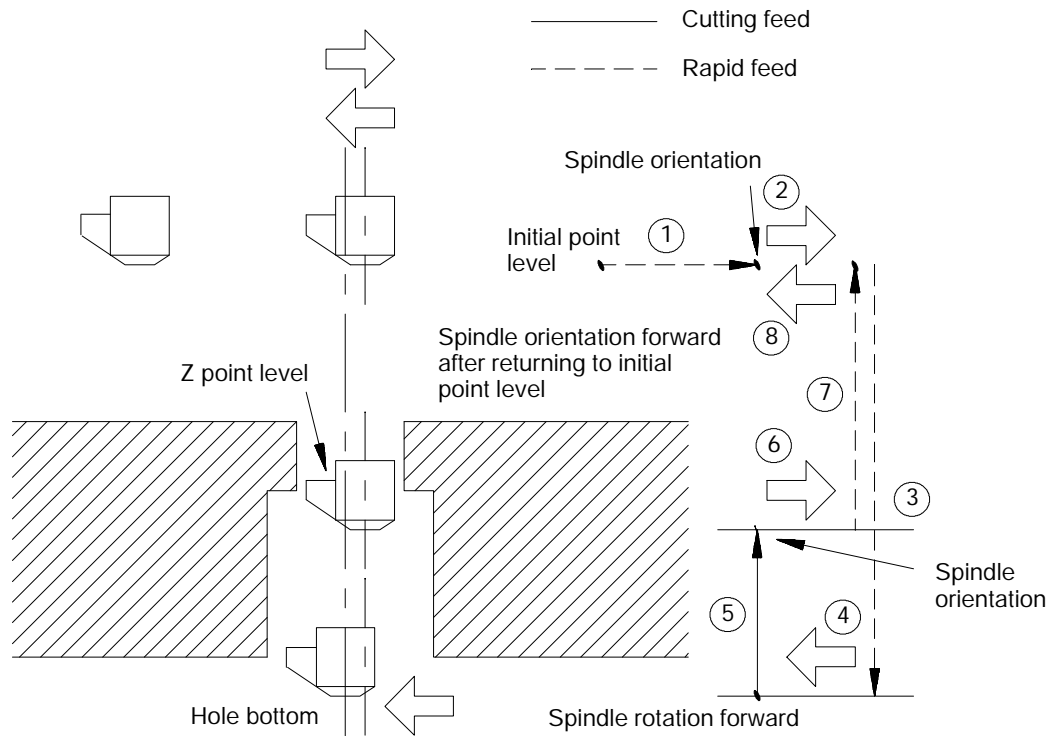
G87X_Y_Z_{I_J_K}R_F_L;
Q_

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the Z point level. The Z point level in this case is the top of the hole that is being cut by the back boring operation.
Q or I, J, K	defines the tool shift amount and direction.
R	defines the position beyond the hole bottom so the tool can safely shift.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: This cycle requires an existing hole through which the back boring tool can be safely lowered prior to the back boring operation.

Figure 26.14
G87: Back Boring Cycle



In the G87 back boring cycle, the control moves the axes in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. After the back boring tool is positioned, the control orients the tool to a position determined in AMP by the system installer.

The control shifts the boring tool by one of two possible methods, as discussed below. The shift method is determined in AMP by the system installer. Refer to the documentation supplied by your system installer for additional information.

Method I

This shift method is a single axis shift. The direction and axis for the shift is set in AMP by the system installer or can be altered using the milling fixed cycle parameter table (see section 26.5).

- The direction of the axis is specified as + or -.
- The feedrate using this shift method is the programmed feedrate.
- The Q-word shift amount is always interpreted as a positive value. A negative Q-word is not allowed.

Method II

The direction of the shift using this method is programmed in the boring cycle block. Program a shift amount for axes in the current plane only (determined by G17, G18, or G19) using the following words:

I__ programs an X axis move.
J__ programs a Y axis move.
K__ programs a Z axis move.

Follow the I-, J-, and K-words (modal during milling fixed cycles) with incremental values in the block that programs the hole position.

When using Method II, remember:

- If both axes in the current plane are to be shifted, specify both words to move the axes.
 - The move generated will be a single linear move and will execute at rapid traverse.
3. The back boring tool moves at a rapid feedrate through the existing hole to the depth designated by the R-word.
 4. Once the designated depth is reached, the back boring tool shifts the same distance but in the opposite direction as the previous shift (the shift made in step 2).

After this shift, the programmer or operator must start spindle rotation. The spindle must rotate in the clockwise direction.

5. The control retracts the back boring tool at the cutting feedrate to a level specified by the Z-word.
6. After reaching the Z depth, the spindle rotation stops so that the control can re-orient the back boring tool to the position specified in AMP.

The back boring tool is shifted a third time, in the same manner as in step 2, so that it is again “off-center” and can be removed through the existing hole.

7. The back boring tool moves at a rapid feedrate to the initial point level regardless of whether G98 or G99 are active.
8. The back boring tool is shifted a fourth time, in the same manner as in step 2, returning to the initial X, Y coordinates of the hole location.

(G88): Boring Cycle,
Spindle Stop/Manual Out

The format for the G88 cycle is:

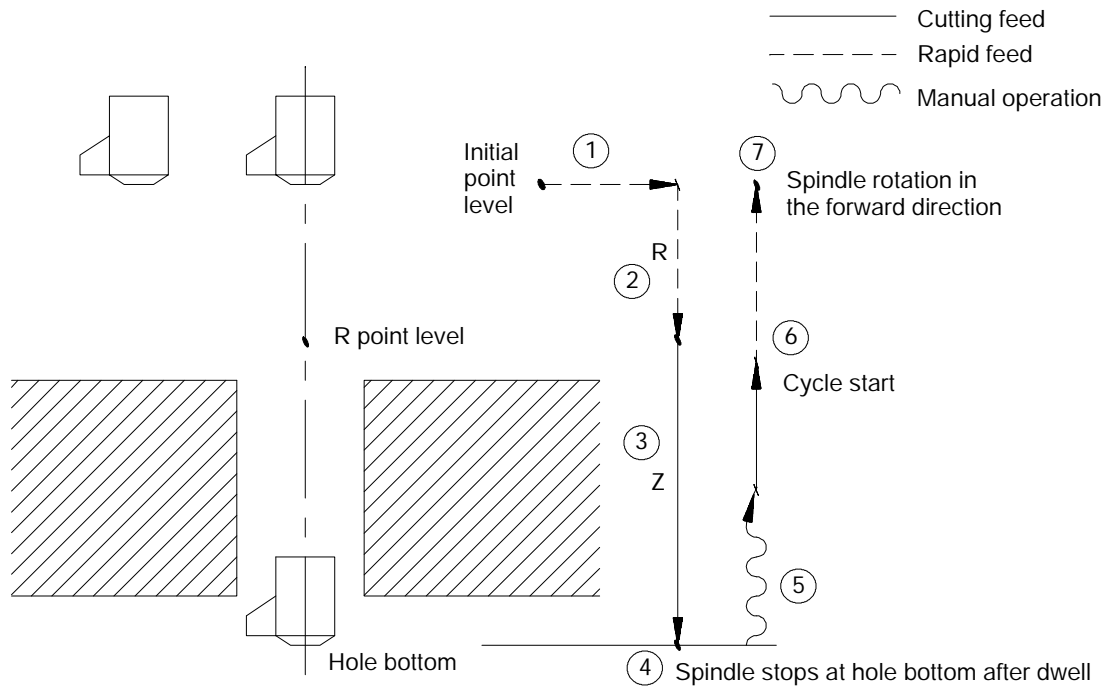
G88X__Y__Z__R__P__F__L__;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at the hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.15
G88: Boring Cycle, Spindle Stop/Manually Out



In the G88 boring cycle, the control moves the axis in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool bores at the programmed feedrate until it reaches the depth specified with the Z-word.
4. If the user has entered a value for the P parameter, the boring tool will dwell after it reaches the bottom of the hole.
5. After the tool reaches the Z depth, the spindle stops revolving. At this point, the operator must perform a manual retraction of the drilling axis as described in chapter 4. (Press <CYCLE START> to return the control to automatic mode.)
6. The boring tool is then retracted at a rapid feedrate to initial point level, as determined by G98.

7. At this point, the rotation of the spindle changes to the clockwise direction.

When the single block function is active, the control stops axis motion after steps 1, 2 and 5.

(G89): Boring Cycle,
Dwell/Feed Out

The operations in G89 are identical to as those of the G85 boring cycle with the exception that the control executes a dwell at hole bottom.

The format for the G89 cycle is as follows:

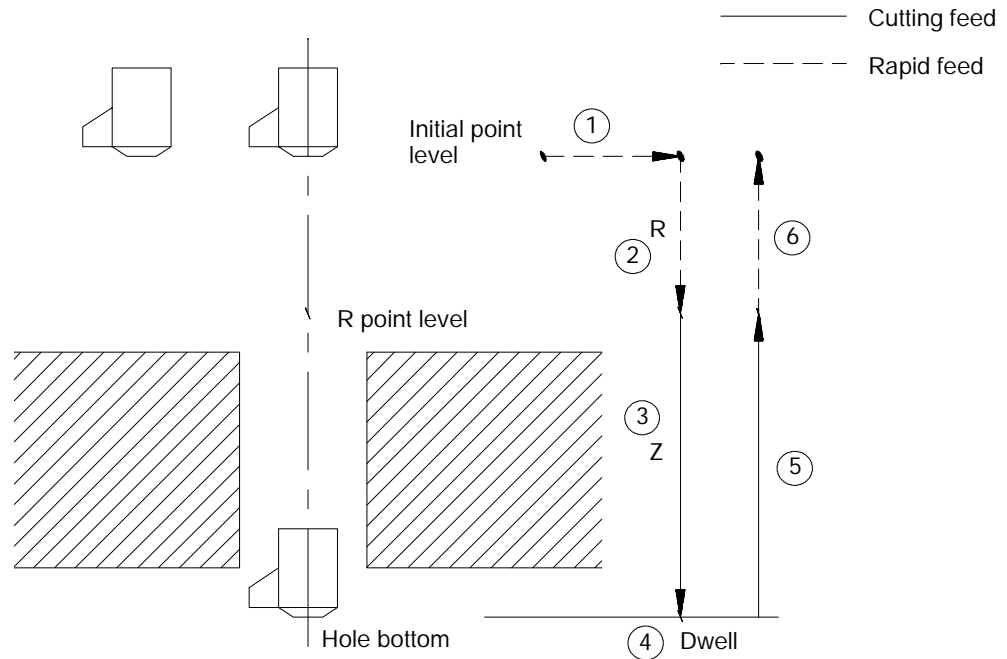
G89X_Y_Z_R_P_F_L_;

Where :	Is :
X, Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(See section 26.3 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 26.16
G89: Boring Cycle, Dwell/Feed Out



In the G89 boring cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level above the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool bores at the programmed feedrate until it reaches the depth of the hole specified by the Z-word.
4. If the user has entered a value for the P parameter, the boring tool will dwell after it reaches the bottom of the hole.
5. The control retracts the boring tool at the cutting feedrate to the R point level.
6. The boring tool accelerates to the rapid feedrate and retracts to the initial point level.

When the single block function is active, the control stops axis motion after steps 1, 2 and 5.

26.5 Altering Milling Fixed Cycle Operating Parameters

The system installer determines many parameter for the milling fixed cycles in AMP. The following 3 parameters are set in AMP but may be overridden by the operator using the Milling Cycle Parameter screen. When changed through this screen, the new values remain in effect until they are manually changed or AMP is downloaded with new values.

- **G73 Deep Hole Peck Drilling Cycle retract amount** - This parameter determines the value of “d”. “d” for this cycle is the distance above the last infeed step that the control retracts the tool from the part, normally to clear chips. See the section on G73 Deep Hole Peck Drilling cycle for details on this cycle’s operation.
- **Clearance Amount for Cycles** - This parameter also determines the value of “d”. The amount “d” for this cycle is the distance between the end of the tool and the plane of the uncut part. See the section on G83 Deep Hole Drilling for details on this cycles operation.
- **G76 / G87 Fine/Back Boring Cycles Shift Axis** - This parameter determines the axis that the shift amount programmed with “Q” will be on. Note that a shift in either axis, in either direction (positive or negative) for the currently active plane can be selected. This parameter is ignored if the shift direction is programmed in the block using I-, J-, or K-words.

To alter these 3 parameters, follow these steps:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {PRGRAM PARAM} softkey.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVI CE SETUP	MON- TOR	TI ME PARTS	→
↑	PTOM SI /OEM					→

- Press the {MILCYC PARAM} softkey. The Milling Cycle Parameter screen is displayed. Figure 26.17 shows a typical Milling Cycle Parameter screen.

(softkey level 3)

↑	ZONE LIMITS	F1 - F9	MILCYC PARAM	PRBCYC PARAM	→
---	----------------	---------	-----------------	-----------------	---

Figure 26.17
Milling Cycle Parameter Screen

MILLING CYCLE PARAMETERS

ENTER VALUE:

G73 DEEP HOLE PECK DRILLING CYCLE
RETRACT AMOUNT 1.000 [MM]

G83 DEEP HOLE DRILLING CYCLE
CLEARANCE AMOUNT 1.270 [MM]

G76/G87 FINE/BACK BORING CYCLE
G17 G18 G19
SHIFT DIRECTION +X +Z +Y

REPLCE VALUE	ADD TO VALUE	UPDATE & EXIT	QUIT
-----------------	-----------------	------------------	------

- Use the up and down cursor keys to select the parameter you intend to change. The CRT shows the selected parameter in reverse video.

Once a parameter is selected, you can either replace the parameter value, or add to the parameter value:

- To replace the current value of the parameter with a new value, key in the new value on the input line of the CRT and press the {REPLCE VALUE} softkey. The old value will be replaced with the new value just keyed in.
- To add an amount to the current value of the parameter, key in the amount to add to the current parameter value on the input line of the CRT and press the {ADD TO VALUE} softkey. The value just keyed in is then added to the old value for the selected parameter.

5. Replace the parameter value or add to it.

There are two ways to quit the Milling Cycle Parameter screen:

- To save the changes just made to the parameters and leave the Milling Cycle Parameter screen, press the {UPDATE & EXIT} softkey.
- To discard any changes just made to the parameters and leave the Milling Cycle Parameter screen, press the {QUIT} softkey.

(softkey level 3)

	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	
--	-----------------	-----------------	--	------------------	------	--

6. If you want to quit the Milling Cycle Parameter screen, save or discard any changes made.

26.6 Examples of Drilling Cycles

The following are example programs and an illustration of G83, deep hole drilling cycle. Example 26.2 is in incremental mode, Example 26.3 is in absolute. Figure 26.18 illustrates the result for both programs individually.

Example 26.2
Programming G83, Deep Hole Drilling Cycle in Incremental Mode

```

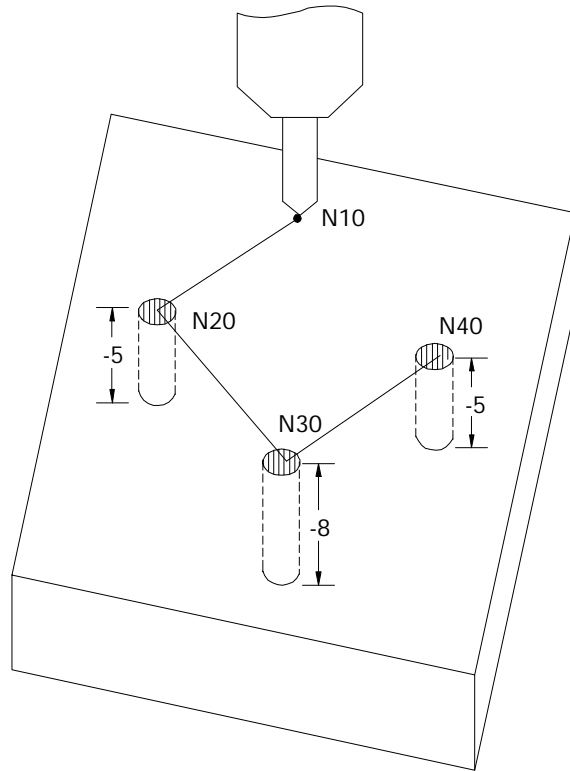
N10      G90 G00 X5 Y12;
N20      G91 G83 X-4 Y-2 Z-3 R-2 Q1.5;
N30      X4 Y-5 Z-5;
N40      X4 Y5 Z-3;
N50      M30;
    
```

Example 26.3
Programming G83, Deep Hole Drilling Cycle in Absolute Mode

```

N10      G90 G00 X5 Y12 Z0 G17 F200;
N20      G83 X1 Y10 Z-5 R-2 Q1.5;
N30      X5 Y5 Z-8;
N40      X9 Y10 Z-5;
N50      M30;
    
```

Figure 26.18
Result of Examples 27.2 and 27.3



END OF CHAPTER

Skip, Gauge, and Probing Cycles

27.1 Chapter Overview

This chapter describes the external skip, gauge, and probe functions available on the control. Use this table to find information:

Information on:	On page:
External Skip Functions (G31 codes)	27-2
Tool Gauging External Skip functions (G37 codes)	27-4
Hole Probing (G38)	27-8
Parallel Probing Cycle (G38.1)	27-12
Probing Parameters Table	27-15

27.2 External Skip, Gauge, and Probe Functions

Use skip, gauge and probe for various automatic measurements, and also for interfacing automatically to external mechanical devices.

These functions are available:

- **External Skip (G31-G31.4)** -- These functions move an axis on a linear path until the control receives an external signal through the PAL program. The block is aborted, the control records the axis coordinate as a paramacro parameter, and program execution skips on to the next block
- **Tool Gauging (G37 - G37.4)** -- These function identically to the external skip except that the axis coordinates (at the time the external signal is received) can be used to modify the tool offset table.
- **Hole Probing (G38)** -- Use this probing cycle to directly measure the diameter and center point location of a hole in a part.
- **Parallel Probing (G38.1)** -- Use this probing cycle to measure the amount that a part is out of parallel with a specified axis.

Important: The G04 dwell feature may also be enabled as an external skip or tool gauging command. For details about G04 see chapter 14.

The control provides several means of triggering an external skip, gauge, or probing block:

- Discrete inputs on the I/O ring
- Any one of the four available “High Speed inputs” (not available on 9/230 CNCs)
- A “Probe” input that directly latches the feedback counters.

These different inputs, each with different degrees of precision, may be used to signal the control to store the current axes positions. Refer to documentation prepared by your system installer for details on your specific machine.

These conditions must be satisfied when an external skip, gauge, or probe block is executed:

- Cutter compensation must be disabled (G40 mode) when the block is executed.
- The block that contains the external skip G code (excluding G04 as external skip) must be a linear block.



CAUTION: We do not recommend using a skip block from any fixed cycle block (such as drilling or pocket milling). If you do choose to execute a skip block in a fixed cycle mode, be aware that the block that is skipped when the trigger occurs can be a cycle generated block. If this is the case the cycle will continue normal execution skipping only the portion of the cycle that was executing when the trigger occurred. If the generated block skipped is a crucial portion of the cycle, damage to the part or machine tool can occur.

27.2 External Skip Functions (G31 codes)

Use external skip functions to terminate the execution of motion commands in a block when the control receives a signal through PAL. When the program block is terminated any remaining axis motion generated by the block that has not been performed remains unexecuted (other non-motion commands are still performed). The control continues normal program execution at the beginning of the next block following the skipped block.

The external skip function is controlled by G31, G31.1, G31.2, G31.3, and G31.4. The system installer determines what signal (such as a touch probe, manual switch, etc.) corresponds to each G31 code in PAL. The system installer can choose different signals to correspond to G31.2, G31.3, and G31.4. G31 and G31.1 are functionally the same, always using the same external signal and the same AMP defined feedrate. With proper PAL programming, a G04 dwell in seconds may also be used as an external skip function.

Format for any G31 external skip blocks is as follows:

G31 X__ Y__ Z__ F__;

Where :	Is :
G31	Any of the G codes in the G31 series or G04. Use the one that is configured to respond to the current external skip signal device that is being used.
X, Y, Z	The endpoint of the move if no external skip signal is received. These also determine the direction that the tool will travel in.
F	The external skip function feedrate. If no value is entered here, the external skip function will execute at either the currently active feedrate, or the feedrate defined for it in AMP (based on whether the AMP parameter Use AMP Skip Feedrate is set to "NO" or "YES"). A value entered here replaces the currently active feedrate and supersedes the AMP defined feedrate.

The G31 series of G codes always produce linear motion regardless of the current mode active at their execution. After their completion the control returns to the operating mode active before the external skip block was read (G00, G01, G02, G03).

Important: The move that immediately follows a G31 series external skip block cannot be a circular move.

The coordinates of the axes when the external skip signal is received are available as the paramacro system parameters #5061-#5066 (work coordinate system) and #5071-#5076 (machine coordinate system). These values will have been adjusted to compensate for the probe tip radius if a radius compensation value was entered.

For example, assume you have entered a probe tip radius of .01. It is triggered as axis 2 approaches in the positive direction at the axis 2 coordinate of 1.1200. The value available for paramacro parameter #5072 would be 1.1300

Probe tip radius is defined by the system installer in AMP. This value may also be changed through the paramacro system parameter #5096, or through the probe parameter table described in section 27.5.

Skip Function Application Examples

One typical application for these G-codes would be moving the part until it contacts a probe and then proceeding with a machining operation from that point. This would provide part feature consistency by insuring that the machining of all parts began from a fixed reference point (probe trigger point).

Note that for this application, the probe tip radius would not be significant, and should be entered as zero if the skip signal position paramacs are to be used.

Another typical application for these G-codes would be to mount the probe as if it were a tool. When the probe contacts the part and triggers, coordinate data would be available in the paramacs for use in the remainder of the part program.

Note the probe tip radius is significant for this application.

27.3 Tool Gauging External Skip Functions (G37 codes)

Tool gauging functions are similar to external skip functions. The key difference is that the tool gauging cycles use the actual tool position (when the external skip signal is received) to enter values in the tool offset table for the currently active offset.

Use tool gauging functions to terminate the execution of motion commands in a block and modify offset tables when the control receives a signal through PAL. When the program block is terminated any remaining axis motion generated by the block that has not been performed remains unexecuted (other non-motion commands are still performed). The current tool position is stored, and the control continues program execution at the beginning of the next block following the skipped block.

The gauging function is controlled by G37, G37.1, G37.2, G37.3, and G37.4. The system installer determines what signal (such as a touch probe, manual switch, etc.) corresponds to each G37 code in PAL. The system installer can choose different signals to correspond to G37, G37.1, G37.2, G37.3, and G37.4. G37 and G37.1 are functionally the same, always using the same external signal and the same AMP defined feedrate.

Format for any G37 skip blocks is as follows:

G37 Z__ F__;

Where :	Is :
G37	Corresponds to any of the G-codes in the G37 series. Use the one that is configured to respond to the current skip signal device that is being used.
X, Y, Z	The axis on which the offset measurement is to be taken is specified here as either X, Y, or Z. Only one axis may be specified in a G37 block. The numeric value following the axis name corresponds to the exact coordinate at which the skip signal is expected to occur. This value is a signed value (+ or -) and determines the initial direction of travel. If modifying the tool length offset, then the tool length axis should be programmed. If modifying the tool diameter, then any axis that is not a tool length axis may be programmed.
F	The tool gauging external skip function feedrate. If no value is entered here, the external skip function will execute at either the currently active feedrate, or the feedrate defined for it in AMP (based on whether the AMP parameter Use AMP Skip Feedrate is set to 'NO' or 'YES'). A value entered here replaces the currently active feedrate and supersedes the AMP defined feedrate.

The system installer determines (in AMP) a position tolerance for the G37 functions. This tolerance defines a legal range before and after the coordinate position programmed with the axis word in the G37 block.

If the skip signal is received before the tool enters or after the tool exits the position tolerance range, a PROBE ERROR will occur. This error appears on the screen as a warning but does not place the control in E-STOP. Instead the G37 block is aborted, and program execution proceeds to the next block. No modification of the tool offset table is performed.

The target offset value for these gauging operations is determined by the currently active offset number (active D word for tool diameter offset, active H word for tool length offset). Note that for length offset measurement, not only must the correct H word be active but the correct offset must be on (G43, or G44). If the tool diameter offset value is to be modified, then the following conditions must be true:

- a tool diameter offset (D word) must be active though diameter compensation (G41, G42) does not need to be on.
- for most tools to get an accurate measurement for tool diameter, the tool must be oriented so that the measured diameter is the outside edge of the tool (not a flute or other geometric feature that would affect the tool diameter).



CAUTION: If modifying a tool length offset, the offset value generated with this gauging operation is immediately loaded into the offset table. Since this offset must be the currently active offset, it becomes effective either immediately when the next block is executed, or delayed until the next block that contains motion on the tool length axis is executed (when an offset is activated is determined in AMP).

The G37 series of G-codes always produce linear motion regardless of the current mode active at their execution. After their completion, the control returns to the operating mode active before the skip block was read (G00, G01, G02, G03).

Important: The move that immediately follows a G37 series skip block cannot be a circular move.

The system installer determines in AMP if the new value is added to, or replaces the old value in the table. The system installer also determines in AMP what gauge cycles alter which tool offset tables, geometry, or wear.

If these probe parameters have been entered, the control automatically compensates for probe radius and length when calculating tool offset changes.

The coordinates of the axes when the external skip signal is received are available as the paramacro system parameters #5061-#5066 (work coordinate system) and #5071-#5076 (machine coordinate system). These values will have been adjusted to compensate for the probe tip radius and the probe length if radius and length compensation values were entered.

For example, assume you have entered a probe tip radius of .01. It is triggered as axis 2 approaches in the positive direction at the axis 2 coordinate of 1.1200. The value available for paramacro parameter #5072 would be 1.1300

Probe tip radius and probe length are defined by the system installer in AMP. These values may also be changed through the paramacro system parameters #5096 (for radius) and #5095 (for length), or through the probe parameter table described in section 27.5.

Refer to the paramacro chapter for details on paramacro parameters.

Tool Gauging Application Example

A typical application for these G-codes in determining tool offsets would execute as follows:

1. When the control executes the G37 block, the triggering device moves towards the tool using the axis specified in the block.
2. When the control receives the appropriate skip signal through PAL, axis motion stops.
3. The control records the position when the skip signal is received. It determines the difference by subtracting the position specified with the axis word in the G37 block from this position. The difference is then added to, or replaces the value in the appropriate geometry or wear table for the currently active tool offset number.

Figure 27.1
Typical Tool Gauging Configurations

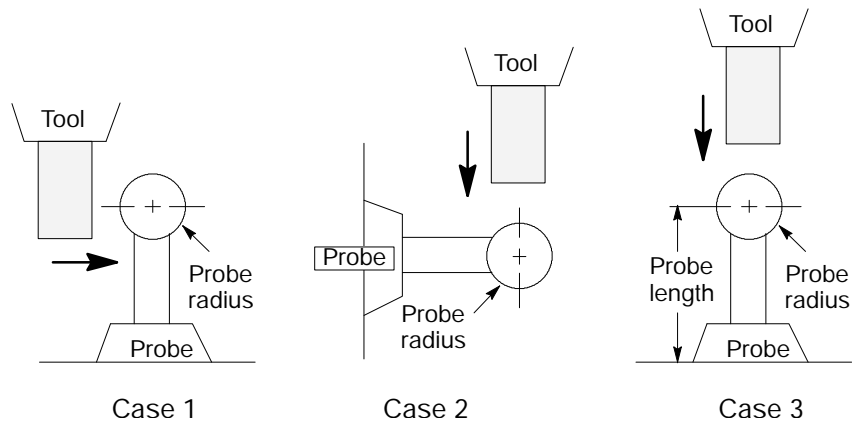


Figure 27.1 illustrates 3 typical tool gauging configurations. All 3 cases assume that the probe is at a known, fixed point on the machine.

Note that in Case 1 the tool radius is being gauged, while in Case 2 the tool length is being gauged. In both of these cases, only the probe tip radius is significant to the control in calculating the offset adjustment.

In Case 3, the tool length is being gauged, and both the probe radius and the probe length are significant to the control's offset adjustment calculations.

Important: The tool gauging configuration Case 3 depicted in Figure 27.1 is not recommended due to the risk of probe damage.

27.4 Hole Probing (G38)

The purpose of this cycle is to provide a means to measure the actual radius and/or locate the center of a hole in a part or gauge using a touch probe.

To use the G38 cycle, the currently active plane when the G38 is programmed must be the same plane that the hole to be measured is in (see chapter 13 for information on plane selection). For example, to measure a hole that is cut in the XY plane, the G38 code must be programmed with the XY plane active.

Format for the G38 code is as follows:

G38 H__ R__ D__ E__ F__;

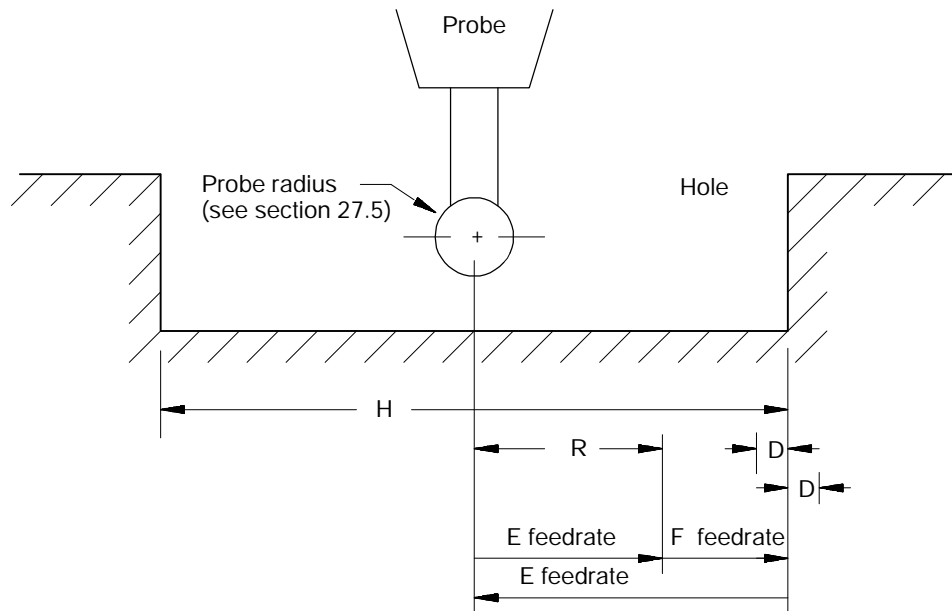
Where :	Is :
H	The estimated or expected diameter of the hole. This parameter is optional. If H is not programmed, the control will generate an H value that is equal to $2(R+D)$, where R is the approach distance and D is the tolerance band. Note that if $R+D$ is greater than one half H, the control will ignore the programmed H value and use $2(R+D)$ as the new expected diameter.
R	The incremental unsigned approach distance. Enter the distance from the start-point of the probing cycle to a point that it is desirable for the feedrate to be slowed. At this point, the feedrate will slow from the approach feedrate (E) to the probing feedrate (F). This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed in section 27.5.
D	The tolerance band distance. The value entered for D defines a band on each side of the expected diameter entered with the H parameter. Enter a value defining a tolerance distance on either side of the expected probe triggering point. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed in section 27.5.
E	The approach feedrate. Enter a value for this parameter that defines the feedrate at which the probe is to approach the position specified by the R parameter. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed in section 27.5.
F	The probe feedrate. Enter a value for this parameter that defines the feedrate at which the probe is to move after passing the point defined by the R parameter. The probe continues on at this feedrate until contact has been made with the diameter of the hole or until the tolerance band is exceeded. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed in section 27.5.

Parameters R, D, E, and F can be entered in three ways:

- The system installer may have entered them in AMP, in which case they will always be available and need not be programmed in the G38 block. Refer to the documentation provided by your system installer.
- They may be entered or changed through the probing parameters table described in section 27.5. If entered in the table, they need not be programmed in the G38 block. The table value will supersede any values entered in AMP.

- They may be programmed directly in the G38 block. Values entered for these parameters in the G38 block supercede both AMP values and probe parameter table values.

Figure 27.2
Parameters for G38 Hole Probing Cycle



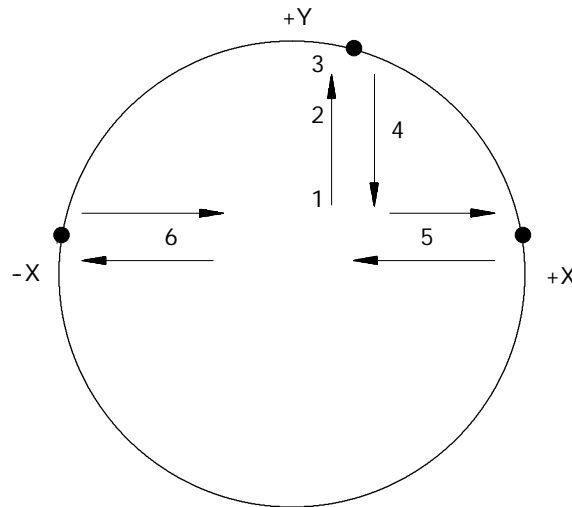
When the G38 code is executed, the probe must have previously been positioned with the probe in the hole so that it will contact the sides of the hole when axis motion takes place in the currently active plane. Ideally (to maximize accuracy) the probe should be positioned near the center of the hole.

The control executes the G38 cycle in the following manner:

1. When the G38 block is read during program execution, the control starts axis motion in the positive direction of the second axis defining the plane at the approach feedrate (E). If the plane is the XY plane, then this would be the Y axis.
2. The axis continues on at the approach feedrate (E) until the approach distance (R) is reached. Then the feedrate is reduced to the probe feedrate (F).

3. The axis continues towards the estimated diameter (H) until the probe signals that contact has been made. If the probe triggers before reaching the negative tolerance band (D), or does not trigger after passing through the positive tolerance band (D), a PROBE ERROR will occur. This error appears on the screen as a warning but does not place the control in E-STOP. Instead the G38 block is aborted, and program execution proceeds to the next block.
4. If the probe triggers within the tolerance band, the position is recorded, and the axis returns to the start-point of the probing operation at the approach feedrate (E).
5. The control repeats the preceding steps on the first axis in the current plane. If the plane is the XY plane, then this would be the X axis.
6. After successfully probing in the positive direction of the first axis in the current plane, the control will then probe in the negative direction of the first axis. In all, three points are measured for determining the circle's radius.

Figure 27.3
Typical Probe Path During G38 Hole Probing



The control calculates the actual radius and center position of the hole from the three data points just measured.

After the probing moves are completed and the hole center location has been calculated, the axes are positioned at the approach feedrate (E) to the exact hole center location.

Important: To accurately measure a hole radius and determine its center, the exact probe tip radius must be available to the control. This value is entered either through AMP, through paramacro system parameter #5096, or through the probe parameter table discussed in section 27.5.

Table 27.A shows the paramacro system parameters used to relay information from the probing operation to the programmer. Refer to the paramacro chapter for details on these system parameters.

Table 27.A
Paramacro System Parameters for G38 Hole Probing

System Parameter	Value this parameter holds
#5092	G38 hole center coordinate on first axis of active plane
#5093	G38 hole center coordinate on second axis of active plane
#5094	G38 radius of hole

An easy way to view these parameters is to copy them to a common parameter that may be accessed on a paramacro screen; for example, the program block:

```
#500=#5094;
```

would take the value of the cycles calculated hole radius and copy it to common parameter 500 that may be viewed in the macro table {COM 2A PARAM}. Note that the {COM 2A PARAM} parameters also allow a parameter description to be entered in the table, such as "HOLE RADIUS VALUE".

27.5
Parallel Probing Cycle
(G38.1)

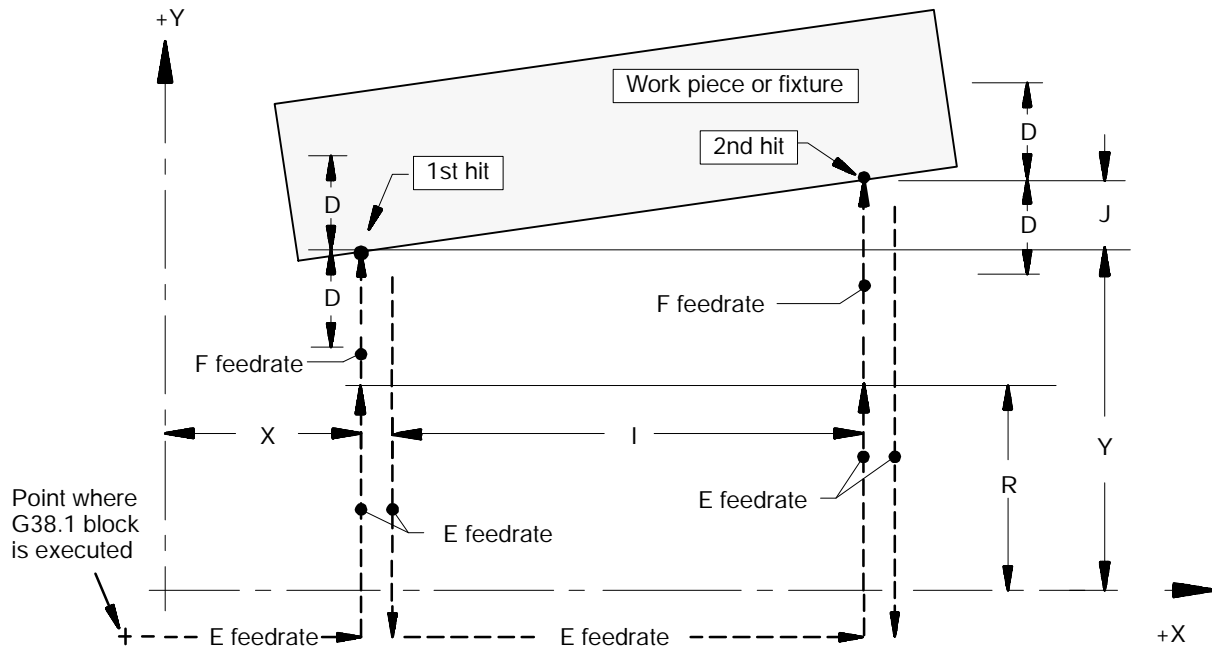
The purpose of this cycle is to provide a means to measure the amount that a part is out of parallel (or rotated) with a selected axis through the use of a touch probe. Note that the currently active plane (G17, G18, or G19) must be the same plane in which probe motion is to occur in and must be active before the probing cycle block is executed.

Format for the G38.1 code is as follows:

G38.1 X__ Y__ I__ J__ R__ D__ E__ F__;

Where :	Is :
X	Any valid axis name followed by the coordinate position of the first measuring point on that axis. May be an absolute or incremental, signed value. Being the first axis word in the G38.1 block indicates that this axis is the one from which measurements are to be taken. The G38.1 probing cycle will determine how much the part is out of parallel with this axis
Y	The name of any axis that is perpendicular to the first axis in the G38.1 block. May be an absolute or incremental, signed value. Parallelism will be measured by moving the probe along this axis to the edge of the part. The value entered with this parameter defines the expected position on this axis where the probe will hit the edge of the part.
I (integrand of first axis in G38.1 block)	The incremental signed distance between the first and second probe hits. This incremental distance is measured along the first axis programmed in the G38.1 block. In this manual I is the integrand word for the X axis. The integrand word for an axis is determined in AMP.
J (integrand of second axis in G38.1 block)	The estimated amount the part is out of parallel. J is an incremental, signed distance. J is added to the coordinate value entered with the second axis in the G38.1 block for the second probe hit only. The net result is to shift the tolerance band (programmed with the D word) by the amount J.
R	The incremental unsigned approach distance. This parameter determines the distance the second axis in the G38.1 block travels at the E feedrate when probing towards the part. After this distance is reached the probe slows to the F feedrate. This parameter is optional. If not entered, the control will default to the value entered in the probing table discussed in section 27.5.
D	The tolerance band distance. The value entered for D defines a band on both sides of the expected endpoint entered with the Y parameter. Enter a value for this parameter defining a tolerance distance on either side of the expected probe triggering point (Y above). This parameter is optional, but must have a positive value if programmed. If not programmed, the control will default to the value entered in the probing cycle parameter table discussed in section 27.5.
E	The approach feedrate. Enter a value for this parameter that defines the feedrate used to reach the approach distance (R). This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed in section 27.5.
F	The probe feedrate. Enter a value for this parameter that defines the feedrate at which the probe is to move after passing the point defined by the R parameter. The probe continues on at this feedrate until contact has been made with the edge of the part or the tolerance band is exceeded. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed in section 27.5.

Figure 27.4
Parameters and Motion Paths for G38.1 Probing Cycle



Parameters R, D, E, and F can be entered in 3 ways:

- The system installer may have entered them in AMP, in which case they will always be available and need not be programmed in the G38.1 block. Refer to the documentation provided by your system installer.
- They may be entered or changed through the probing parameters table described in section 27.5. If entered in the table, they need not be programmed in the G38.1 block. The table value will supersede any values entered in AMP.
- They may be programmed directly in the G38.1 block. Values entered for these parameters in the G38.1 block supersede both AMP values and probe parameter table values.

The control executes the G38.1 cycle in this manner:

1. When the G38.1 block is executed, the control initially moves only the first axis in the G38.1 block to the coordinate position entered with it. The approach feedrate (E) is used for this move.
2. The second axis in the G38.1 block is then moved to the coordinate defined by the approach distance parameter, R. Again, the approach feedrate (E) is used.
3. The feedrate is then reduced to the probe feedrate (F). The second axis continues to move into the tolerance band (D) until the probe triggers, signaling that contact has been made. If the probe triggers before reaching the negative tolerance band (D) or does not trigger after passing through the positive tolerance band (D), a PROBE ERROR will occur. This error appears on the screen as a warning but does not place the control in E-STOP. Instead the G38.1 block is aborted, and program execution proceeds to the next block.
4. If the probe triggers within the tolerance band, the position is recorded. The axis then reverses direction and retracts the distance traveled in steps 2 and 3 above at the approach feedrate (E).
5. The first axis in the G38.1 block moves the incremental distance entered with the I parameter. The approach feedrate (E) is used for this move.
6. Steps 2, 3, and 4 above are repeated. Note that if the J parameter is programmed, the value of J is added to the Y parameter for the second probe hit. This will shift the location of the tolerance band programmed with the D word.

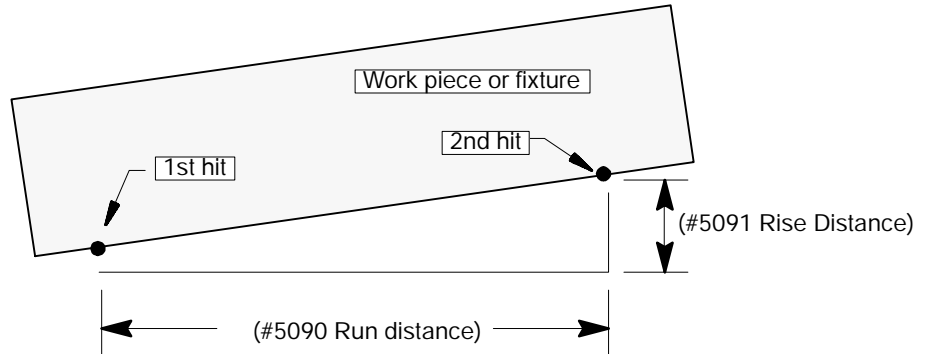
At this point, the control calculates the amount that the workpiece or fixture is out of parallel (with the first axis in the G38.1 block) using the two recorded positions. Note that the probe tip radius is not a factor in G38.1 parallel probing.

Table 27.B and Figure 27.5 show the values generated by this cycle and stored by the control as paramacro system parameters. These values may be viewed using the paramacro tables as discussed in chapter 28.

Table 27.B
Paramacro System Parameters for G38.1 Parallel Probing.

System Parameter	Value this parameter holds
#5090	G38.1 run measurement (always equal to I)
#5091	G38.1 rise measurement

Figure 27.5
G38.1 Parallel Probing Cycle Paramacro Parameter Values



27.6
Probing Parameters Table

Use this feature to access the Probe Parameters table and alter probe parameters affecting the operation of the G31, G37, and G38 codes . For details on the parameters available here, refer to the appropriate section of this chapter.

Access to this table may be restricted. Refer to chapter 2 regarding access control.

The parameters available for modification through this table may have already been defined in AMP. Refer to the documentation provided by your system installer.

Values entered here will supersede the corresponding AMP value. They will remain active even if the control is powered down.

To display or alter the values in the probing parameters table follow the steps below:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {PROGRAM PARAM} softkey.
(softkey level 2)

↑	PROGRAM PARAM	AMP	DEVI CE SETUP	MON- TOR	TIME PARTS	→
↑	PTOM SI/OEM					

3. Press the {PROBE PARAM} softkey to display the probing cycle parameter table.
(softkey level 3)

↑	ZONE LIMITS	F1 - F9	MI LCYC PARAM	PROBE PARAM		
---	----------------	---------	------------------	----------------	--	--

Figure 27.6
Probing Parameters Screen

PROBING PARAMETERS					
	APPROACH DISTANCE	R		0.000	[MM]
	TOLERANCE BAND	D		0.000	[MM]
	APPROACH FEEDRATE	E		0.000	[MPM]
	PROBE FEEDRATE	F		0.000	[MPM]
	PROBE LENGTH			0.000	[MM]
	PROBE RADIUS			0.000	[MM]
	REPLCE VALUE	ADD TO VALUE	INCH/ METRI C	UPDATE & EXIT	QUIT

4. Use the up, or down cursor keys to move the block cursor to the probing parameter to be changed. The selected parameter will be shown in reverse video.

5. You can change parameter values two ways:

- Press the {REPLCE VALUE} softkey then type in a new value for the selected parameter by using the keys on the operator panel. When you press the [TRANSMIT] key, the value typed in will replace the old value for that parameter.
- Press the {ADD TO VALUE} softkey then type in a value to add to the old value for the selected parameter by using the keys on the operator panel. When you press the [TRANSMIT] key, the value typed in will be added to the old value for that parameter.

(softkey level 4)

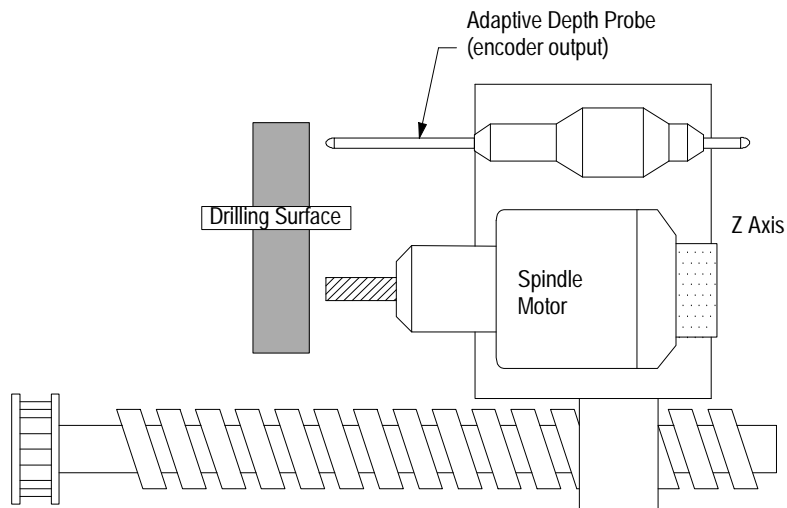
↑	REPLCE VALUE	ADD TO VALUE	SUB FM VALUE	UPDATE & EXIT	QUIT
---	-----------------	-----------------	-----------------	------------------	------

6. Exit the probing parameters screen in one of two ways:

If you want to:	Press this softkey:
save your changes and exit	{UPDATE & EXIT}
lose your changes and exit	{QUIT}

27.7
Adaptive Depth (G26)

Use the Adaptive Depth feature to enable an adaptive depth probe that monitors tool depth relative to the actual part surface. This feature is sometimes referred to as “cut to length” or “cut to depth”. This feature allows a more flexible part mounting system (small changes to part size or part mounting do not require reprogramming of the station) and a less accurate machine drive system (tool position is relative to the part surface rather than the machine home). The system installer can determine in AMP if the position loop, for moves that use the adaptive depth feature, is closed by the axis encoder or the adaptive depth probe.



The should know the following terms which are used throughout this discussion on adaptive depth:

Term:	Definition:
Adaptive Depth Probe	This is the mechanical device used to measure the location of an axis relative to some surface.
Adaptive Depth Axis	The controlling axis that is configured to be positioned by the adaptive depth probe. The adaptive depth probe is typically mechanically connected to the adaptive depth axis.
Trigger Tolerance for Probe	This is the amount of probe deflection necessary to indicate the adaptive depth probe has fired.
Maximum Probe Deflection	This is the AMPed travel limits for deflection of the adaptive depth probe. This does not include any probe deflection that occurs before the probe fired is indicated.
Probe Fired	This indicates the depth prove has recorded more feedback counts then the trigger tolerance and these have occurred beyond the probe zero point.
Probe Zero Point	This is a location established by touching off the adaptive depth probe to the part or established at power turn on. It typically aligns the tip of the adaptive depth probe with the drilling tool tip.
Adaptive Depth Feedback Source	This is the device that controls the positioning of the adaptive depth axis. Typically this is an encoder mounted to the motor shaft for normal moves, and the adaptive depth probe for moves that use a G26 block.

Format for an adaptive depth block is as follows:

$$G26 \begin{pmatrix} X_ \\ Y_ \\ Z_ \end{pmatrix} \begin{pmatrix} I_ \\ J_ \\ K_ \end{pmatrix};$$

Where:	Programs:
X, Y, or Z	Adaptive Depth Axis word. Use the axis word associated with the adaptive depth (the system installer selects this axis as the controlling axis in AMP). Program an axis destination that is sufficiently beyond where you expect the depth probe to contact the part surface. This position should be far enough beyond the probe contact point so as no deceleration occurs before the probe fires. This destination can be programmed as either an absolute or incremental value (G90 or G91 mode). Only one axis word can be programmed in a G26 block.
I, J, or K	Axis integrand. Use the axis integrand for the adaptive depth axis. Program an incremental distance equal to the amount of deflection you want the adaptive depth probe to depress. Since this integrand value is unsigned, the incremental deflection programmed here is in the same direction of motion as when the trip occurs. This should be the actual depth of the hole beyond the part surface. This axis integrand word is always an incremental value. The resolution to which you can program this word is dependant on the adaptive depth axis word format.. You may be able to program this depth with greater accuracy than the axis is normally capable of. The system installer also sets the maximum allowable adaptive depth probe deflection in AMP. This value is also the maximum value for the integrand.

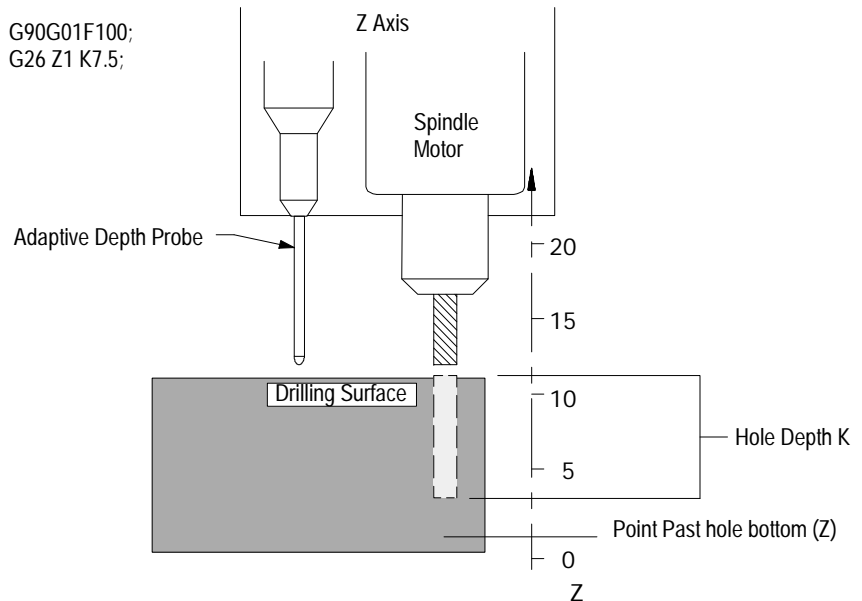


WARNING: The system installers maximum value for probe deflection is relative to the probe zero point. It is not an absolute value. Don't allow a large amount of probe deflection where you establish the zero point. The larger this deflection, the greater chance that you will exceed the maximum deflection your probe can withstand thus nullifying your machine tool builders maximum probe deflection.

The control will perform its normal axis deceleration as it approaches the final depth. When the final depth is reached the axis stops and the part program continues on from that point. Since the actual location of the endpoint of the move is not known until the probe is depressed, the control must re-setup any part program blocks that it previously read into block look ahead. If you program an adaptive depth probe distance (with the integrand word) that is very close to the AMPed probe trigger tolerance, axis speed as the control searches for a probe input may be reduced to allow enough time for the axis to decelerate once the probe has fired. For example if .2 inches of deflection is the probe trigger tolerance and you program an integrand of .21 inches, the control may need to limit the feedrate since only .01 inches are available for deceleration of the axis once the probe has fired.

You must program the G26 block in G01 mode. Programming G26 in other cutting modes (such as G00, G02, or G03) will generate an error. The G26 command is not modal and must be programmed in all blocks that use the adaptive depth feature. Using the G61 (inposition mode) during G26 blocks means that the G26 block will wait for the depth probe to reach the AMP'ed inposition.

Example 27.1
Adaptive Depth Drilling 7.5 mm Into Part Surface



The system installer determines how many counts of the adaptive depth probe constitutes contact with the part (a probe fired event AMPed as the probe trigger tolerance). Multiple counts are typically required because of the potential for probe deflection from acceleration/deceleration or machine vibration. This number of counts for the probe to fire is added back into the current axis position for an accurate part surface location. The position of the adaptive depth probe when it senses part contact is handled by the control as follows:

Probe Position when G26 Programmed:	Results in:
No probe deflection when adaptive depth axis starts moving.	Normal operation, control drills to programmed integrand depth.
Probe is already depressed some distance when adaptive depth axis starts moving.	<ul style="list-style-type: none"> • If current probe deflection is less than programmed integrand depth the control assumes it is executing either consecutive G26 blocks on that axis or it is continuing a previously aborted adaptive depth move. The tool continues on to integrand depth. • If current probe deflection is past the programmed integrand depth the tool retracts to the new integrand position relative to the part surface.
No probe deflection and the programmed integrand depth is smaller than the AMPed probe trigger tolerance.	The control will generate an error since the tool would exceed the integrand letter depth before the probe fires. You must either increase the drilling depth (the integrand amount in block) or your system installer must, using AMP, decrease the probe trigger tolerance.

The depth probe must fire before the programmed endpoint (G26 block axis word) is reached. If the axis reaches the programmed endpoint before the probe has fired, the control displays an error message and block execution stops. You must reset the program block to continue.

Once the probe is depressed from a G26 block:

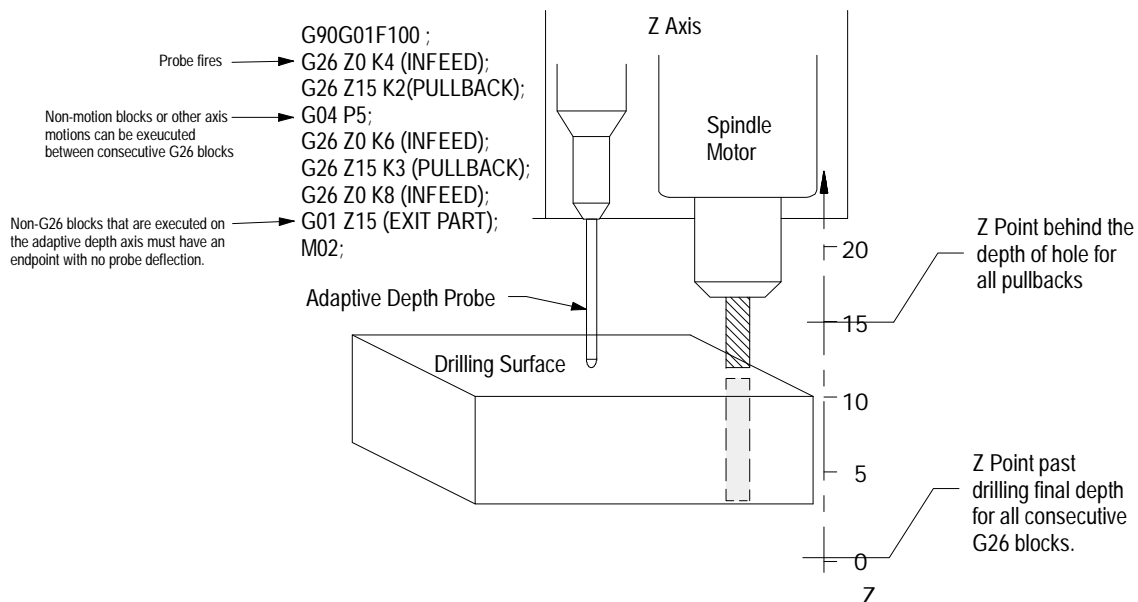
- additional G26 blocks that use the probe can be executed. Once the adaptive depth probe is fired and remains depressed, any G26 block can position the adaptive depth axis using the adaptive depth probe for position feedback. These G26 blocks must be consecutive except for:
 - non motion blocks
 - blocks that do not position the adaptive depth axis
 - non G26 blocks that completely move the adaptive depth probe off of the part (probe position less than zero at endpoint).

Once the probe is fired you must position the adaptive depth axis (assuming the probe is closing the feedback loop) using the integrand word in a G26 block. You must also still program an adaptive depth axis word (see Example 27.2). If you use a non-G26 block to position the adaptive depth axis, feedback is immediately switched from the adaptive depth probe back to the adaptive depth axis' normal feedback device. The endpoint of this non-G26 block must completely remove any probe deflection that occurs beyond the probes zero point or a warning is generated.

- as long as the probe remains depressed, other G26 blocks reference the zero point established the last time the probe tripped.
- the axis decelerates to a zero feedrate between consecutive G26 blocks (even if they are in the same direction). This is a result of the control performing a re-setup of part program blocks before executing any G26 block.

To program consecutive G26 blocks you must still program the G26 command along with the adaptive depth axis and its integrand. The adaptive depth axis block endpoint should be programmed in the same direction and well beyond the actual probe deflection at end point for any G26 block. The axis integrand (being the depth of cut) programs the new tool position. For example:

Example 27.2
Adaptive Depth Drilling with Consecutive G26 Blocks
(Peck Drilling Example)



“Probe Trips During Deceleration” Warnings

An axis deceleration can occur before the probe trips caused by the programmed endpoint of the G26 block being too close to the position at which the depth probe trips. In this situation, the control having failed to receive a probe fired indication, has begun to decelerate to the G26 programmed endpoint. During this deceleration the probe trips and feedback is switched over to the adaptive depth probe. The warning message “PROBE TRIP DURING DECL” is displayed. The axis then begins accelerating to the location defined by the integrand word. This can cause rough servo response as the feedback is switched from the axis encoder to the depth probe during a deceleration. This error can also be generated in subsequent G26 blocks after the adaptive depth probe is fired, when the adaptive depth axis word is programmed in a G26 block in the opposite direction that the integrand is requesting. See Example 27.2 for details on correctly programming the axis word.

Probe trips during deceleration can also occur as a result of the control thinking the G26 block is complete (axis within the in position band) however the probe trip occurs while the following error is collapsing. Solve this problem by moving the programmed endpoint of the adaptive depth axis further into the part or use a lower feedrate to reduce following error.

Unexpected Probe Trips

An error message is generated if the adaptive depth probe is not depressed at the start of the block and it suddenly starts generating counts during motion of a non G26 block in excess of the AMPed probe trigger tolerance amount. When this unexpected probe trip occurs the control will enter cycle stop. No axis deceleration ramp occurs for this condition. The total probe deflection will equal the AMPed probe trigger tolerance plus the existing following error for that axis at the point of the unexpected trip.

If the probe trips while the axis is performing a manual jog, the control will display an error message. The axis will continue the jog unless the maximum probe deflection is reached. The system installer determines the maximum probe deflection in AMP. Manual jogs always use feedback from the adaptive depth axis, not the adaptive depth probe.

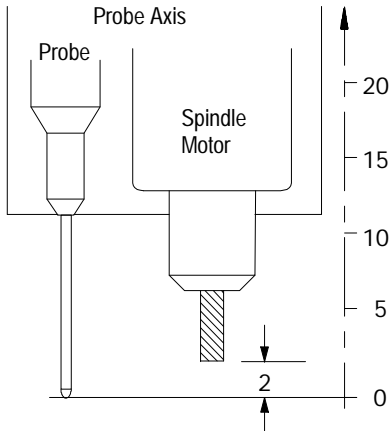
Zeroing the Adaptive Depth Probe

Use this feature to create a zero point for your adaptive depth probe. The adaptive depth probe only recognizes positive probe counts (those that occur with deflection greater than the probe zero point).

The adaptive depth probe position is zeroed automatically at power turn on. In the event that you must re-zero the probe the system installer can write PAL to enable you to zero the probe any time the adaptive depth axis is not in motion. Refer to your system installers documentation on using \$ADPCLR.

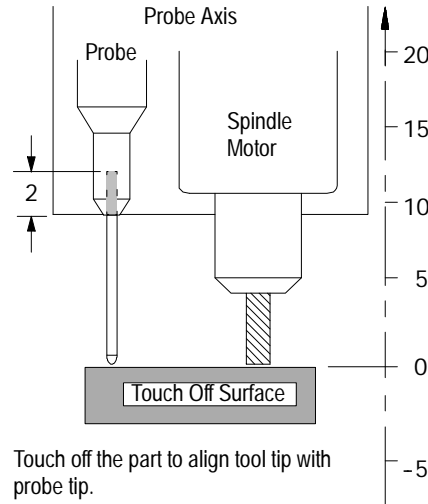


WARNING: We strongly recommend that you require (either through PAL or some other procedure) a manual probe zeroing operation be performed after every power up. This will help prevent problems that can occur if an invalid zero point is established at power up.



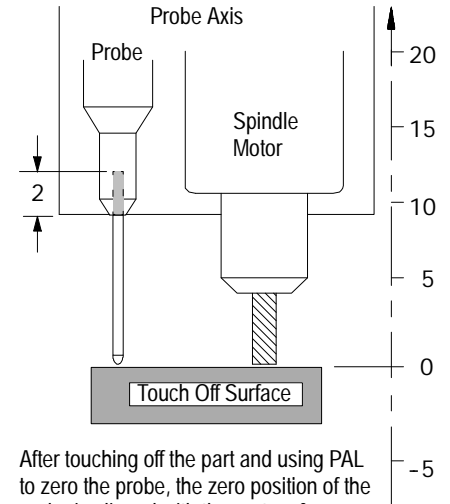
Probe zero position after power up is not aligned with tool tip. Typically no probe deflection exists at power up.

Axis monitor page for depth probe:
Following error = 0.0000
Probe position = 0.0000



Touch off the part to align tool tip with probe tip.

Axis monitor page for depth probe:
Following error = +0.2000
Probe position = -0.2000



After touching off the part and using PAL to zero the probe, the zero position of the probe is aligned with the part surface.

Axis monitor page for depth probe:
Following error = 0.0000
Probe position = 0.0000

Probe counts that occur when the probe deflection is less than zero will not trip the probe. Since the probe is not tripped in this range, the control will still use the axis feedback for positioning. Probe fired only occurs after the probe position register becomes positive (deflection reaches the zero point) plus the number of counts required for probe fired (probe trigger tolerance).

Feature Considerations

This feature:	Used with G26 adaptive depth:
Virtual C Spindle Cylindrical Interpolation Corner Radius and Chamfer Programming All Fixed Cycles (except some transfer line only cycles) QuickPath Plus program blocks	Is incompatible with the adaptive depth probe. An error is generated if one of these features is active or activated when a G26 block is executing.
Cutter compensation Part Rotation	Is incompatible with the adaptive depth probe only when the G26 block is used on an axis in the currently active plane. An error is generated if one of these features is active or activated when a G26 block is executing in the active plane.
Feedrate Clamping	Feedrate clamping occurs as in normal control operation. During a G26 block an additional feedrate clamp occurs that limits the following error of the adaptive depth axis from exceeding the programmed integrand amount.
Scaling	If you choose to use scaling during a G26 adaptive depth block keep in mind both the programmed endpoint and the axis integrand word are scaled.
Block Retrace	You can not perform a block retrace on any block that is executed with the adaptive depth probe fired. Additionally you can not completely retrace any block that will retrace to a position that results in the adaptive depth probe being fired. Attempting to retrace one of these blocks will stop motion once the probe fires and generates an error.
Interrupt Macros	You can execute an interrupt macro during a G26 block provided either: <ul style="list-style-type: none"> the return move does not generate any probe deflection beyond the zero point the blocks in the interrupt macro program do not deflect the probe beyond the zero point. If at least one of these constraints is not met, executing the return move of an interrupt macro will generate a unexpected probe trip error. You can not execute a G26 block within an interrupt macro.
Block Reset/Control Reset/E-STOP Reset	Performing one of these reset operations will abort any executing G26 block. If the probe remains depressed however, any additional G26 blocks use the current probe position relative to the part surface. You do not need to re-fire the probe. The probe remains tripped through the reset operation. Programming another G26 block uses the current probe deflection to calculate the endpoint from the executing G26 integrand word. If you require the adaptive depth probe to re-locate the part surface after a reset operation, you must first remove any probe deflection before executing the G26 block.
QuickCheck	When the G26 block is executed in QuickCheck, the control will not use the integrand word. Axis endpoints will appear to be at the programmed axis word location. The probe does not need to fire during QuickCheck.

This feature:	Used with G26 adaptive depth:
Dual and De-skew axes	Is incompatible with the adaptive depth probe. An error is generated when you attempt to run an adaptive depth cycle if one of these types of axes are configured as the adaptive depth axis.
Mid Start/ Program Restart	If motion to the block that is selected for Mid Start/Program Restart generates a move that fires the depth probe, a unexpected depth probe trip will occur. You can not use these features to start at any block that would normally have been executed with the depth probe tripped.
Polar Programming (G16)	You can not program a G26 block if the adaptive depth axis is in the current plane and the control is in polar programming mode.

If you would like to monitor your adaptive depth probe position, you can view the probe location on the adaptive depth probe axis monitor screen. The system installer can also make this information available through a PAL display page by monitoring adaptive depth axis information. Refer to your Integration and Maintenance Manual for details on using the monitor pages. Your system installer may have password protected the monitor softkey.

END OF CHAPTER

Paramacros

28.0 Chapter Overview

This chapter describes paramacros and how to program them. Use this table to find information:

Information on:	On page:
Paramacros	28-1
Parametric Expressions	28-2
Transfer of Control Commands	28-7
Parameter Assignments	28-12
Assigning Parameter Values	28-37
Macro Call Commands	28-45
Macro Output Commands	28-55

28.1 Paramacros

The paramacros feature is similar to a subprogram with many added features. Special features available with a paramacro are:

- Computable variables
- Computable word address fields in any block type
- Variable to and from PAL
- Access to certain modal system parameters for computations
- Arithmetic operators and expressions for computations
- Conditional Branching, Subroutines, and Subprogram calls based upon logical function results
- Parametric programs, Subroutines, and Subprograms
- Parametric Autocycles
- User-definable prompts to aid in program generation and execution
- Prompting of parameters for main program execution
- Calculator function under prompt edit

All of these features are valid in any block within a main program, subprogram, or paramacro program. Most are permitted in an MDI program unless otherwise stated; the only restriction being that no other program commands, except other paramacro commands, may exist in a block that contains paramacro commands. Macro and non-macro commands may not exist in the same program block.

28.2 Parametric Expressions

It may be necessary for mathematical expressions to be evaluated in a complex paramacro. This requires that some form of mathematical equation be written in a paramacro block. The following is a discussion of the operators and function commands available for use on the control. These operators and function commands are valid in any block within a program, subprogram, paramacro, or MDI program.

28.2.1 Basic Mathematical Operators

This subsection lists the mathematical operators that are available on the control. Use these operators to accomplish mathematical operations that are necessary to evaluate the basic mathematical equation such as addition, multiplication, etc. Table 28.A lists the operators and their meanings.

Table 28.A
Mathematical Operators

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
[]	Brackets
OR	Logical OR
XOR	Logical Exclusive OR
AND	Logical AND
MOD	Modulus

The control executes a mathematical operation in this order:

1. Any part of the expression between the brackets [] is evaluated first.
2. Multiplication, division, and MOD are evaluated second.
3. All other operations are evaluated third.

If the same level of evaluation is performed, the left-most operation takes priority.

Example 28.1
Mathematical Operations

Expression entered	Result
12/4*3	9
12/[4*3]	1
12+2/2	13
[12+2]/2	7
12-4+3	11
12-[4+3]	5

All logical operators have the format of:

A logical operator B

where A and B are numerical data or a parameters with a value assigned.

If B is negative in the above format, an error will occur.

If A is negative, the absolute value of A is used in the operation and the sign is attached to the final result.

Before evaluation, A and B are made integers by rounding and truncating.

Example 28.2
Logical Operation Examples

Expression Entered	Result
[16.2MOD3]	1.0
[-16.2MOD3]	-1.0
[-17.6MOD3]	0.0
[16.0MOD3]	1.0
[-5AND4]	-4.0
[4.4AND3.6]	4.0
[5AND-4]	ERROR
[83886079AND83886080]	67108864

28.2.2
Mathematical Function
Commands

This subsection lists the basic mathematical functions that are available on the control and their use. Use these functions to accomplish mathematical operations that are necessary to evaluate the trigonometric and other complex mathematical equation such as rounding off, square roots, logarithms, exponent, etc. Table 28.B lists the basic functions that are available and their meanings.

Table 28.B
Mathematical Functions

Function	Meaning
SIN	Sine (degrees)
COS	Cosine (degrees)
TAN	Tangent (degrees)
ATAN	Arc Tangent (degrees)
ASIN	Arc Sine (degrees)
ACOS	Arc Cosine (degrees)
SQRT	Square Root
ABS	Absolute Value
BIN	Conversion from BCD to Decimal
BCD	Conversion from Decimal to BCD
ROUND	Rounding Off (nearest whole number)
FIX	Truncation Down
FUP	Truncation Up
LN	Logarithms (base e)
EXP	Exponent

When programming these functions, the value on which that function is to be performed must be included in brackets: for example, SIN [10]. The exception to this is the arc tangent function. The format to ATAN requires the division of two values. For example, ATAN [10]/[2] is used to calculate the arc tangent of 5.

The functions in Table 28.B are executed from left to right in a program block. These functions are executed before the control executes any mathematical operators such as addition or subtraction. This order of execution can be changed only by enclosing operations in brackets []. Operations enclosed in brackets are executed first.

Example 28.3
Format for Functions

SIN[2]	This evaluates the sine of 2 degrees.
SQRT[14+2]	This evaluates the square root of 16.
SIN[SQRT[14+2]]	This evaluates the sine of the square root of 16.
LN[#2+4]	This evaluates the logarithm of the value of parameter #2 plus 4.

Example 28.4
Mathematical Function Examples

Expression Entered	Result
SIN[90]	1.0
SQRT[16]	4.0
ABS[-4]	4.0
BIN[855]	357
BCD[357]	855
ROUND[12.5]	13.0
ROUND[12.4]	12.0
FIX[12.7]	12.0
FUP[12.2]	13.0
FUP[12.0]	12.0
LN[9]	2.197225
EXP[2]	7.389056

Important: Precaution must be taken when performing calculations within the brackets [] following a mathematical function. The operations within the bracket are performed first, and then the function is performed on this resultant.

Example 28.5
Precaution for Order of Operation

N1#1=1.6;	Parameter #1 is set at 1.6
N2#2=2. 8;	Parameter #2 is set at 2.8
N3#3=ROUND[#1+#2] ;	Parameter #3 is set at 4.0

Note that the values composing parameter #3 are added together first and then rounded, not rounded and then added together.

28.2.3 Parametric Expressions as G- or M- Codes

You can use parametric expressions to specify G-codes or M-codes in a program block.

For example:

```
G#1 G#100 G#500 M#1 M#100 M#500;
```

```
G#520 G[#521-1] G[#522+10] M#520 M[#522+1] M[#522+10];
```

When using a parametric expression to specify a G- or M-code, remember:

- When specifying more than one G- or M-code in a block from the same modal group, the G- or M-code closest to the End-of-Block of that block is the one activated. All others in that modal group are ignored.
- Parametric expressions that generate G- or M-codes used to call a paramacro are invalid. If the result of the paramacro expression for a G-code is 65, 66, 66.1, or any AMP-defined G-code, the error “ILLEGAL G-CODE” appears. If the result of the paramacro expression for an M-code is any AMP-defined M-code, the control will not execute the macro but interpret the M-code as either a system defined M-code or a user defined M-code. No error is generated.
- To get the G- or M-code value, the system will truncate, after the tenths position, the result of the mathematical expression. The following example assumes #1=37.0:

This Block	Generates This G-Code
G#1	G37.0
G[#1+0.32]	G37.3
G[#1+0.49]	G37.4

Illegal Paramacro Commands

It is possible to call subprograms or paramacros within an MDI program, however, there are limitations to the allowable commands. The following lists examples of illegal MDI commands for these features:

- G66
- G66.1
- G67
- DO-END
- WHILE-DO-END
- GOTO
- IF-GOTO
- M99
- Amp-defined Modal G-code Macro Calls

Attempting to use any of the above as MDI commands, 9/PC generates an “ILLEGAL MACRO CMD VIA MDI” error message.

28.3 Transfer of Control Commands

Use transfer of control commands to alter the normal flow of program execution. Normally the control executes program blocks sequentially. By using control commands, the programmer can alter this normal flow of execution and transfer execution to a specific block or begin looping (executing the same set of blocks repetitively).

Important: Transfer of control commands call a block by its N number. If more than one N number exists in a block, the control uses only the left-most N number in that block. If the same N number is used for more than one block, the control will use the first block it encounters with the correct N number (the control searches in the forward direction first, then starts at the top of the program).

Two types of transfer of control commands are available:

- **Conditional** -- The execution of a jump or loop is dependant on whether a mathematical condition is true.
- **Non-Conditional** -- The execution of a jump or loop is always performed when that block is executed.

28.3.1 Conditional Operators

This section describes conditional operators that are available for paramacro programming. A conditional operator causes a comparison between two values and yields a result of true or false. Use conditional operators in “IF” or “WHILE” commands as described in sections 28.3.2 and 28.3.3 .

Use the true or false condition to determine whether the “IF” or “WHILE” blocks are executed. Table 28.C lists the conditional operators available for paramacro programming:

Table 28.C
Conditional Operators

Operator	Condition Tested
EQ	Equal
NE	Not Equal
GT	Greater Than
LT	Less Than
GE	Greater Than or Equal
LE	Less Than or Equal

Program a condition between the [and] brackets in this format:

[A EQ B]

where A and B represent some numerical value. The values for A and B can be in the form of some mathematical equation or in the form of a paramacro parameter.

Example 28.6
Evaluation of Conditional Expressions

Expression	Evaluation
[6.03 EQ 6.0301]	FALSE
[6.03 NE 6.0301]	TRUE
[2.5 GT 2.5]	FALSE
[2.5 LT 2.51]	TRUE
[2.51 GE 2.5]	TRUE
[2.5 LE 2.5]	TRUE
[[2.5-3] LE 1]	TRUE
[#1 GT #2]	This depends on the value of the parameters #1 and #2

For details on the use of conditional expressions, refer to page 28-9 on “IF” statements and page 28-10 on “WHILE” statements. For details on the use of paramacro parameters, refer to page 28-12.

28.3.2 GOTO and IF-GOTO Commands

Unconditional GOTO

Any time the control executes a GOTO block, the unconditional GOTO command automatically transfers control.

Use this format for the GOTO command:

GOTO n;

Where :	Is:
n	Execution is transferred to the block with the sequence number specified as n any time that the GOTO block is executed.

Example 28.7
Unconditional GOTO

```
N1. . . ;
N2. . . ;
N3GOTO5;
N4. . . ;
N5. . . ;
N6. . . ;
/N7GOTO1;
```

In Example 28.7, execution continues sequentially until block N3 is read; then execution transfers to block N5 and again resumes sequential execution to block N6. If optional block skip 1 is off, block N7 will transfer execution back to block N1.

Conditional IF-GOTO

The conditional IF-GOTO command is dependent on whether a mathematical condition is true. If this condition is true, execution will transfer to the block specified.

Use this format for the IF-GOTO command:

```
IF [(condition)] GOTO n;
```

Where :	Is :
(condition)	some mathematical condition. This condition is tested by the control to determine if it is true or false.
n	if the condition is tested as true, execution is transferred to the block specified as n

If the condition is tested as false, execution falls through the block and the GOTO is not executed. Program execution continues in a normal fashion.

Example 28.8
Conditional IF

```
N1. . . ;
N2IF[#3EQ-1.5]GOTO5;
N3. . . ;
N4. . . ;
N5. . . ;
N6IF[#4LT3]GOTO1;
N7. . . ;
```

When block N2 is read, parameter #3 is compared to the value -1.5. If the comparison is true, then blocks N3 and N4 are skipped, and execution continues on from block N5. If the comparison is false, then execution continues to block N3. When block N6 is read, parameter #4 is compared to the value 3. If the comparison is true, then execution is transferred to block N1; if it is false, execution continues to block N7.

28.3.3 DO-END and WHILE-DO-END Commands

Unconditional DO-END

The unconditional DO-END command is rarely used. The lack of a condition here causes the control to loop indefinitely until reset or <CYCLE STOP> is pressed, or until some other transfer of control command forces execution out of the loop.

The format for the unconditional DO-END command is as follows:

```
DO m;
:
:
:
END m;
```

Where :	Is :
m	a loop identifier used to relate a DO block with an END block. The value of m must be the same for the DO as it is for the corresponding END. This value can be either 1, 2, or 3.

All blocks between the DO and the END command are executed indefinitely or until execution is stopped by some external operation such as by pressing <E-STOP> or <CYCLE STOP>, or when a block delete is performed if programmed.

Conditional WHILE-DO-END

The conditional WHILE-DO-END command is dependant on whether a mathematical condition is true. If this condition is false, execution will transfer to the block immediately following the END statement block.

Use this format for the WHILE-DO-END command:

```
WHILE [ (condition) ] DO m;
;
;
;
END m;
```

Where :	Is :
(condition)	some mathematical condition. This condition is tested by the control to determine if it is true or false.
m	an identifier used by the control to relate a DO block with an END block. The value of m must be the same for the DO as it is for the corresponding END. This value can be either 1, 2, or 3.

All blocks between the DO and the END command are executed until the condition is tested as false. This set of blocks is referred to as a WHILE-DO-END program segment.

When the condition for the WHILE-DO block is tested as false, execution is then transferred to the block immediately following the END statement block.

Example 28.9
WHILE-DO-END Program Segment

```
N1 #1=1;
N2WHILE[#1LT10]DO1
N3#1=[#1+1];
N4. . . ;
N5. . . ;
N6END1;
N7. . . ;
```

In Example 28.9, blocks N2 through N6 are executed 9 times. At that time, the condition in block N2 becomes false, and program execution is transferred to block N7.

Nesting is possible with a WHILE-DO-END command. Nesting is defined here as one WHILE-DO-END program segment executing within another WHILE-DO-END program segment. WHILE-DO-END nesting is limited to 3 independent segments at one time.

Example 28.10
Nested WHILE DO Commands

```
N1#1=1;
N2WHILE[#1LT10]DO1;
N3#1=[#1+1];
N4WHILE[#1EQ2]DO2;
N5. . . ;
N6END2;
N7END1;
N8. . . ;
```

In Example 28.10, blocks N2 through N7 are repeated until the condition in block N2 becomes false. Within DO loop 1, DO loop 2 will be repeated until the condition in block N4 becomes false.

28.4 Parameter Assignments

The following subsections describe assigning different paramacro parameter values and how these parameters are used in a paramacro. Use parameters for paramacros to replace a numeric value. They can be said to be used as a variable.

There are 5 types of parameters that may be called for use in a paramacro:

- local - independent set of variables assigned to each nested macro
- common - variables available to all programs
- system - variables that indicate specific system condition
- PAL - provide variables shared between part and PAL programs
- dual process - provide variables shared between processes

The following subsections describe these different types of parameters independently. This in no way means that they are not interchangeable in the same macro program. Mixing the different types of parameters in the same paramacro is acceptable.

28.4.1 Local Parameter Assignments

Local parameters are #1 - #33. There are 5 sets of local parameters. The first set is reserved for use in the main program and any subprogram called by that main program with an M98. The remaining four sets are for each nested level of macro (4 levels of nesting maximum).

Assigned parameter values are specific to the individual macro nesting levels. Local parameters are assigned as described on page 28-37.

Local parameters are used in a specific macro to perform calculations and axis motions. After their initial assignment, these parameters can be modified within any macro at the same nesting level. For example macro O11111 called from a main program has 33 local parameter values to work with (#1 to #33). All macros called from the main program, and nested at the same level, use the same local parameters with the same values unless they are initialized in that macro.

For example macro O11111 called from a main program assigns a value to #1 = 1 and the macro returns execution to the main program with an M99. Later in the same main program (before executing an M99, M02, or M30) macro O11111 is called from the main program again. The value assigned to #1 (=1) remains from the previous macro that executed at that nesting level.

Important: Any local variables you intend to use in a macro we recommend you initialize them before you start using them unless you require values passed from a macro at the same nesting level. In our example above where macro O11111 assigns #1=1. The value of #1 is carried to any macro that is nested at the same nesting level. If for example after macro O11111 returns control to the main program a different macro O22222 is called, the same set of local variables is assigned to O11111 and O22222 because they are both nested at level 1. Confusion could be prevented if before macro O22222 uses #1 it initializes that variable using #1 = 0. All local variables are reset to zero when the control executes an end of program block (M02 or M30) or a program restart code (M99 in a main program).

Considerations for Local Parameters

When assigning values to local parameters, remember:

- All local variable assignments are reset to zero any time the control reads an M02 or M30 in a part program or an M99 in a main program.
- All local variable assignments are reset to zero any time that power is turned on, the control is reset, or an E-STOP reset operation is executed.
- If more than one I,J, or K set is programmed in an argument, use Table 28.H (B) on page 28-38 for the parameter assignment.
- Values displayed in paramacro tables for local parameters are only valid if the active program/subprogram uses that local paramacro. If the active program/subprogram does not use that local paramacro, the table may display a value of zero.

Example 28.11
Assigning Using More Than One I, J, K Set

G65P1001K1I 2J3J4J5;

The above block sets the following parameters:

parameter #6 = 1

parameter #7 = 2

parameter #8 = 3

parameter #11 = 4

parameter #14 = 5

If the same parameter is assigned more than one value in an argument, only the right-most value is stored for the parameter.

Example 28.12
Assigning the Same Parameter Twice

G65P1001R3. 1A2R- 0. 5

The above block sets the following parameters:

parameter #1 = 2.0 As set by the A-word

parameter #18 = -0.5 As set by the last R-word.

The 1st value of 3.1, assigned to parameter #18 by the R-word, is replaced by the 2nd value set by the second R-word.

Example 28.13
Assigning The Same Parameter Twice Using I, J, and K

G65P1001R2I 3. 4D5I - 0. 6

The above blocks set the following parameters:

parameter #18 = 2 As set by the R-word.

parameter #4 = 3.4 As set by the 1st I-word.

parameter #7 = -0.6 As set by the 2nd I-word.

The 1st value of 5, assigned to parameter #7 by the D-word, is replaced by the 2nd value set by the second I-word.

28.4.2 Common Parameters

The common parameters refer to parameter numbers 100 to 199 and 500 to 999 for all 9/Series controls except for the 9/240, which allows 100 to 199 and 500 to 699. The common parameters are assigned through the use of a common parameter table as described on page 28-41.

Common parameters are global in nature. This means that the same set of parameters can be called by any program, macro, subprogram, or MDI program.

Common parameters are divided in to two types: saved or unsaved.

- Saved common parameters refers to the common parameters that retain their value even after power to the control is lost. Saved common parameters are parameter numbers 500 - 999 for all 9/Series controls except the 9/240, which allows only 500 to 699.
- Unsaved common parameters refers to the common parameters that do not retain their value after power to the control is lost. When power to the control is turned back on, these parameters reset their value to zero. Unsaved common parameters are numbers 100 - 199.

The PAL programmer can use some of these parameters to check parametric values with the Paramacro Range Check feature. For more information refer to the description of \$PRMQTY and \$PRMERR in your 9/Series CNC PAL Reference Manual.

28.4.3 System Parameters

System parameters may be used by any part program, including paramacros and subprograms. All of these parameters may be used as data or may be changed by assignment (read and write) unless indicated differently in Table 28.D.

These system parameters are generated by the control and can be modified by operation or programming. They correspond to different control conditions such as current operating modes, offsets, etc.

Table 28.D lists the system parameters.

Table 28.D
System Parameters

Parameter #	System Parameter	Page
2001 to 2999	Tool Offset Tables	28-18
3000	² Program Stop With Message (PAL)	28-19
3001	System Timer (PAL)	28-19
3002	System Clock	28-20
3003	² Block Execution Control 1	28-20
3004	² Block Execution Control 2	28-21
3006	² Program Stop With Message	28-21
3007	¹ Mirror Image	28-22
4001 to 4120	¹ Modal Information	28-22
5001 to 5012	¹ Coordinates of End Point	28-23
5021 to 5032	¹ Coordinates of Commanded Position	28-24
5041 to 5052	¹ Machine Coordinate Position	28-24
5061 to 5069 or 5541 to 5552	¹ Skip Signal Position (Work Coordinate)	28-25
5071 to 5079 or 5561 to 5572	¹ Skip Signal Position (Machine Coordinates)	28-25
5081 to 5089 or 5581 to 5592	¹ Active Tool Length Offsets	28-26
5090 to 5094	¹ Probing Cycle Positions	28-27
5095 to 5096	Probe Stylus Length and Radius	28-28
5101 to 5112	¹ Current Following Error	28-28
5201 to 5212	External Offset Amount	28-28
5221 to 5232	G54 Work Coordinate Table Value	28-29
5241 to 5252	G55 Work Coordinate Table Value	
5261 to 5272	G56 Work Coordinate Table Value	
5281 to 5292	G57 Work Coordinate Table Value	
5301 to 5312	G58 Work Coordinate Table Value	
5321 to 5332	G59 Work Coordinate Table Value	
5341 to 5352	G59.1 Work Coordinate Table Value	
5361 to 5372	G59.2 Work Coordinate Table Value	
5381 to 5392	G59.3 Work Coordinate Table Value	
5630	¹ S-Curve Time per Block	28-31
5661 to 5642	¹ Acceleration Ramps for Linear Acc/Dec Mode	28-31
5651 to 5662	¹ Deceleration Ramps for Linear Acc/Dec Mode	28-31

5671 to 5682	¹ Acceleration Ramps for S-Curve Acc/Dec Mode	28-32
5691 to 5702	¹ Deceleration Ramps for S-Curve Acc/Dec Mode	28-32
5711 to 5722	¹ Jerk	28-33

Table 28.D
System Parameters (continued)

Parameter #	System Parameter	Page
5731 to 5743	Home Marker Distance	28-33
5751 to 5763	Home Marker Tolerance	28-34

¹ These parameters may only have their value received (read-only)

² These parameters may only have their value changed (write-only)

#2001 to 2999
Tool Offset Tables

These parameters may be changed or simply read through programming. The values for these parameters are received or entered into the tool offset tables for geometry and wear (discussed in chapter 3). Table 28.E gives the parameter numbers associated with each table value.

Table 28.E
Tool Offset Table Parameters

	Offset Number	Parameter # for Geometry Table	Parameter # for Wear Table
Tool Length	1 to 99	#2001 to 2099	#2201 to 2299
Tool Radius	1 to 99	#2401 to 2499	#2601 to 2699

#3000

Program Stop With Message (PAL)

Use this parameter to cause a cycle stop operation and display a message on line 1 of the CRT. Any block that assigns any non-zero value to parameter 3000 will result in a cycle stop. The actual value assigned to parameter 3000 is not used. Parameter 3000 is a write-only parameter.

When the control executes this block, a cycle stop is performed and the message "SEE PART PROGRAM FOR MACRO STOP MESSAGE" is displayed on line 1 of the CRT. This is intended to point out to the operator an important comment in the program block that assigns a value to parameter 3000 (see chapter 10 on comment blocks).

For example, programming

```
#3000=. 1 (TOOL NUMBER 6 IS WORN);
```

causes program execution to stop at the beginning of this block and displays a message telling the operator to read the comment in the block. A block reset must be performed before a cycle start will resume normal program execution.

When this block is executed, it also sets the paramacro alarm PAL flag (\$MCALRM) true. Refer to the system installer's documentation for details on the effect of this PAL flag.

#3001

System Timer (PAL)

This parameter is referred to as the timer parameter. It is a read-write parameter. Every 20ms a value of 20 is added to the value of parameter 3001. The value of this parameter is also stored by a PAL flag (\$PM20MS) and may be modified or set by the system installers PAL program. Refer to the system installer's documentation for details on the use of this timer. The maximum value of this parameter is 32768ms. Any value greater than 32768 will cause this parameter to "rollover" to zero and restart counting again. The value of this parameter is reset to zero every time power is lost.

#3002
System Clock

This parameter is referred to as a clock parameter and references an hour counter. It is a read-write parameter with negative value assignments being illegal. The maximum value for this parameter is 1 year (8760 hours). The parameter value is maintained when power is lost. It is incremented by .000005556 every 20 ms.

#3003
Block Execution Control 1

Use this parameter to control whether the control ignores single-block mode and to control when M-codes are executed in a block. The value of this parameter ranges from 0 to 3, and it is a write-only parameter.

These results occur when parameter 3003 is set to the corresponding values:

Value:	Single-block mode:	M-codes are executed:
0	can be activated	at the beginning of the program blocks execution
1	requests are ignored	
2	can be activated	after the complete execution of the other commands in the block
3	requests are ignored	

#3004

Block Execution Control 2

This parameter determines whether a cycle stop request will be recognized, whether the feedrate override switch is active, and whether exact stop mode is available (G61 mode). The range of this parameter is from 0 to 7 and it is a write-only parameter.

Table 28.F shows the results of the different values for parameter number 3004. If they are ignored, the control will not allow the feature to be used. If they are recognized, the feature can be activated in the normal manner.

Table 28.F
Parameter 3004 Values

Value of Parameter	Cycle Stop	Feedrate Override	Exact Stop Mode
0	Recognized	Recognized	Recognized
1	Ignored	Recognized	Recognized
2	Recognized	Ignored	Recognized
3	Ignored	Ignored	Recognized
4	Recognized	Recognized	Ignored
5	Ignored	Recognized	Ignored
6	Recognized	Ignored	Ignored
7	Ignored	Ignored	Ignored

#3006

Program Stop With Message

Use this parameter to cause a cycle stop operation and display a message on line 1 of the CRT. Any block that assigns a new value to the parameter 3006 will result in a cycle stop. Any decimal value may be assigned to this parameter the value of which is not used.

When the control executes this block, a cycle stop is performed and the message "SEE (MESSAGE) IN PART PROGRAM BLOCK" is displayed on line 1 of the CRT. This is intended to point out to the operator an important comment in a program block (see chapter 10 on comment blocks). This parameter is a write-only.

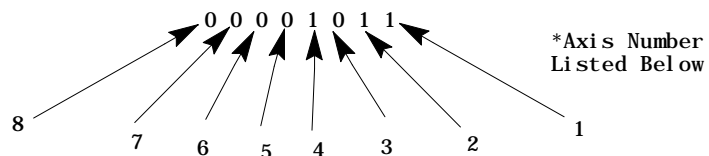
For example, programming:

```
#3006=. 1 (Install Tool Number 6);
```

will cause program execution to stop at the beginning of this block and the message display the message telling the operator to read the comment in the block.

```
#3007  
Mirror Image
```

This parameter is a read-only. It generates an integer that represents, in binary, what axes are mirrored. For example, if the value of this parameter was 3, the binary equivalent for this is 00000011. The first digit of this binary equivalent (reading right to left) corresponds to axis 1, the second is axis 2, the third is axis 3, etc., up to the number of axes configured in your system. When a zero is in the binary location for an axis, it indicates that the axis is not mirrored. If a 1 is in that axis location, that axis is mirrored. For example, if the parameter #3007 is the integer 11 (binary 00001011), it indicates axes 1, 2, and 4 are mirrored and axes 3, 5, and 6 are not mirrored.



Where:

0 indicates axis is not mirrored

1 indicates axis is mirrored

This parameter reflects both the programmed and front-panel (external mirror) status of mirroring on the axes.

```
#4001 to 4120  
Modal Information
```

These are read-only parameters. They indicate the value of a modal program word. Table 28.G shows the modal program word that applies to the given parameter number.

Table 28.G
Modal Data Parameters

Parameter Number	Modal Data Value
#4001 to 4021	These correspond to the different G-code Groups 1-21 (see chapter 10) and show what G-code from group is currently active.
4108	Current E-word value
4109	Current F-word value
4113	Most recently programmed M-code
4114	Most recently programmed N-word
4115	Current program number O-word
4119	Current S-word value
4120	Current T-word value

For example, if currently programming in G02 mode at a feedrate of 100, the parameters would be as follows:

G02 is a group 1 G-code, so its value of 02 is set to parameter number 4001.

The feedrate programmed with an F-word gives parameter number 4109 a value of 100.

#5001 to 5012 Coordinates of End Point

These parameters are read-only. They correspond to the coordinates of the end point (destination) of a programmed move. These are the coordinates in the work coordinate system.

5001	Axis 1 coordinate position	5007	Axis 7 coordinate position
5002	Axis 2 coordinate position	5008	Axis 8 coordinate position
5003	Axis 3 coordinate position	5009	Axis 9 coordinate position
5004	Axis 4 coordinate position	5010	Axis 10 coordinate position
5005	Axis 5 coordinate position	5011	Axis 11 coordinate position
5006	Axis 6 coordinate position	5012	Axis 12 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5021 to 5032
Coordinates of Commanded Position

These parameters are read-only. They correspond to the current coordinates of the cutting tool. These are the coordinates in the work coordinate system.

5021	Axis 1 coordinate position	5027	Axis 7 coordinate position
5022	Axis 2 coordinate position	5028	Axis 8 coordinate position
5023	Axis 3 coordinate position	5029	Axis 9 coordinate position
5024	Axis 4 coordinate position	5030	Axis 10 coordinate position
5025	Axis 5 coordinate position	5031	Axis 11 coordinate position
5026	Axis 6 coordinate position	5032	Axis 12 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5041 to 5052
Machine Coordinate Position

These parameters are read-only. They correspond to the coordinates of the cutting tool in the machine (absolute) coordinate system.

5041	Axis 1 coordinate position	5047	Axis 7 coordinate position
5042	Axis 2 coordinate position	5048	Axis 8 coordinate position
5043	Axis 3 coordinate position	5049	Axis 9 coordinate position
5044	Axis 4 coordinate position	5050	Axis 10 coordinate position
5045	Axis 5 coordinate position	5051	Axis 11 coordinate position
5046	Axis 6 coordinate position	5052	Axis 12 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

Position data for the absolute position of an adaptive depth probe is invalid.

#5061 to 5069 or #5541 to 5552

Skip Signal Position Work Coordinate Position

These parameters are read-only. They correspond to the coordinates of the cutting tool when a skip signal is received to PAL from a probe or other device such as a switch. These are the coordinates in the work coordinate system.

5061	Axis 1 coordinate position	5067	Axis 7 coordinate position
5062	Axis 2 coordinate position	5068	Axis 8 coordinate position
5063	Axis 3 coordinate position	5069	Axis 9 coordinate position
5064	Axis 4 coordinate position		
5065	Axis 5 coordinate position		
5066	Axis 6 coordinate position		

Or if your system has more than 9 axes:

5541	Axis 1 coordinate position	5547	Axis 7 coordinate position
5542	Axis 2 coordinate position	5548	Axis 8 coordinate position
5543	Axis 3 coordinate position	5549	Axis 9 coordinate position
5544	Axis 4 coordinate position	5550	Axis 10 coordinate position
5545	Axis 5 coordinate position	5551	Axis 11 coordinate position
5546	Axis 6 coordinate position	5552	Axis 12 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5071 to 5079 or #5561 to 5562

Skip Signal Position Machine Coordinate System

These parameters are read-only. They correspond to the coordinates of the cutting tool when a skip signal is received to PAL from a probe or other device such as a switch. These are the coordinates in the machine (absolute) coordinate system.

5071	Axis 1 coordinate position	5077	Axis 7 coordinate position
5072	Axis 2 coordinate position	5078	Axis 8 coordinate position
5073	Axis 3 coordinate position	5079	Axis 9 coordinate position
5074	Axis 4 coordinate position		
5075	Axis 5 coordinate position		
5076	Axis 6 coordinate position		

Or if your system has more than 9 axes:

5561	Axis 1 coordinate position	5567	Axis 7 coordinate position
5562	Axis 2 coordinate position	5568	Axis 8 coordinate position
5563	Axis 3 coordinate position	5569	Axis 9 coordinate position
5564	Axis 4 coordinate position	5570	Axis 10 coordinate position
5565	Axis 5 coordinate position	5571	Axis 11 coordinate position
5566	Axis 6 coordinate position	5572	Axis 12 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5081 to 5079 or #5581 to 5592 Active Tool Length Offsets

These are read-only parameters. They correspond to the currently active tool length offsets (see chapter 20).

5081	Current axis 1 tool length offset.	5087	Current axis 7 tool length offset.
5082	Current axis 2 tool length offset.	5088	Current axis 8 tool length offset.
5083	Current axis 3 tool length offset.	5089	Current axis 9 tool length offset.
5084	Current axis 4 tool length offset.		
5085	Current axis 5 tool length offset.		
5086	Current axis 6 tool length offset.		

Or if your system has more than 9 axes:

5581	Current axis 1 tool length offset.	5587	Current axis 7 tool length offset.
5582	Current axis 2 tool length offset.	5588	Current axis 8 tool length offset.
5583	Current axis 3 tool length offset.	5589	Current axis 9 tool length offset.
5584	Current axis 4 tool length offset.	5590	Current axis 10 tool length offset.
5585	Current axis 5 tool length offset.	5591	Current axis 11 tool length offset.
5586	Current axis 6 tool length offset.	5592	Current axis 12 tool length offset.

#5090 to 5094
Probing Cycle Positions

These parameters are read-only. They correspond to values (in the machine coordinate system) generated by the last successful probing cycle. These cycles are programmed using G-codes G38 (circle diameter and center measurement) and G38.1 (parallel probing cycle).

5090	G38.1 rise measurement
5091	G38.1 run measurement
5092	G38 Hole Center Coordinate Location on First hit axis
5093	G38 Hole Center Coordinate Location on Second and Third hit axis
5094	G38 Radius of Hole

Figure 28.1
Paramacro Parameters for G38.1 Parallel Probing

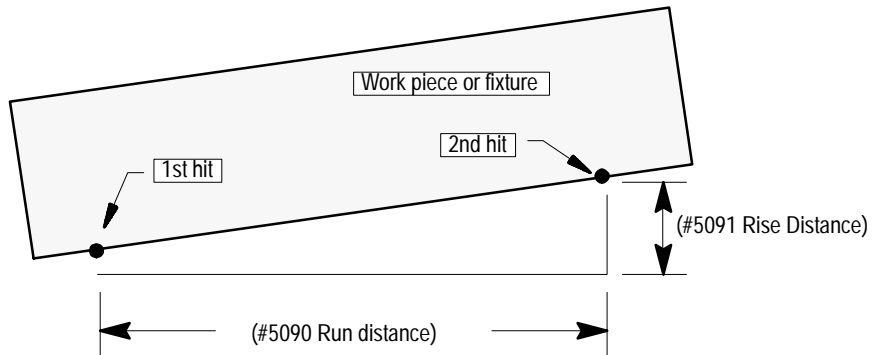
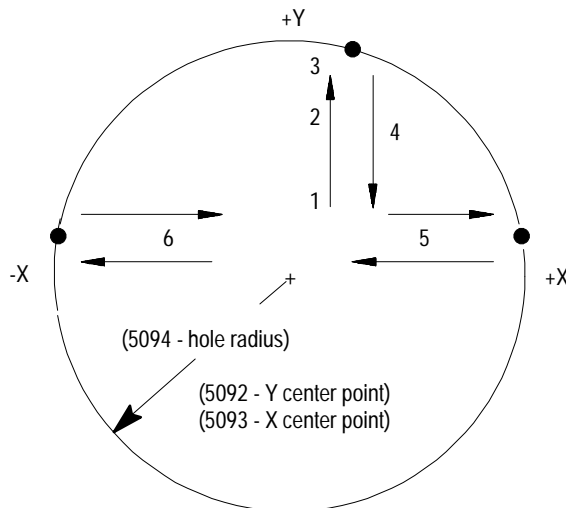


Figure 28.2
Paramacro Parameters for G38 Circle Diameter and Center Measurement



#5095 to 5096
Probe stylus Length and Radius

These parameters correspond to the values set in the probing cycle parameter table discussed in chapter 27. When values are assigned to these parameters, the current values in the probe table is replaced.

5095	Probe stylus Length
5096	Probe stylus Radius

For details on probe radius and length parameters, see chapter 27 on tool gauging.

#5101 to 5112
Current Following Error

These parameters are read-only. They correspond to the current following error for an axis.

5101	Axis 1 following error	5107	Axis 7 following error
5102	Axis 2 following error	5108	Axis 8 following error
5103	Axis 3 following error	5109	Axis 9 following error
5104	Axis 4 following error	5110	Axis 10 following error
5105	Axis 5 following error	5111	Axis 11 following error
5106	Axis 6 following error	5112	Axis 12 following error

The system installer determines in AMP the name (or word) that is used to define the axis. The following error of a system constantly changes. You can use this parameter to take a “snapshot” of the following error, but the value that is read may not the current following error of the system.

#5201 to 5212
External Offset Amount

These parameters are read or write. They correspond to the current value set in the work coordinate table for the external offset (see chapter 3). This allows the reading of data from the tables and also the setting of data into the table by assigning values to the parameters.

5201	Axis 1 offset amount	5207	Axis 7 offset amount
5202	Axis 2 offset amount	5208	Axis 8 offset amount
5203	Axis 3 offset amount	5209	Axis 9 offset amount
5204	Axis 4 offset amount	5210	Axis 10 offset amount
5205	Axis 5 offset amount	5211	Axis 11 offset amount
5206	Axis 6 offset amount	5212	Axis 12 offset amount

The system installer determines in AMP the name (or word) that is used to define the axis. Changes made to the external offset using this paramacro variable go into effect only after the axis has been re-homed, or power to the control has been cycled.

#5221 to 5392

Work Coordinate Table Value

These parameters are read or write. They correspond to the current value set in the work coordinate table for the G54-G59 work coordinate systems (see chapter 3). You can read data from the tables and set data into the table by assigning values to the parameters. The axis names are set in AMP.

5221	G54 Axis 1 Coordinate
5222	G54 Axis 2 Coordinate
5223	G54 Axis 3 Coordinate
5224	G54 Axis 4 Coordinate
5225	G54 Axis 5 Coordinate
5226	G54 Axis 6 Coordinate
5227	G54 Axis 7 Coordinate
5228	G54 Axis 8 Coordinate
5229	G54 Axis 9 Coordinate
5230	G54 Axis 10 Coordinate
5231	G54 Axis 11 Coordinate
5232	G54 Axis 12 Coordinate

5321	G59 Axis 1 Coordinate
5322	G59 Axis 2 Coordinate
5323	G59 Axis 3 Coordinate
5324	G59 Axis 4 Coordinate
5325	G59 Axis 5 Coordinate
5326	G59 Axis 6 Coordinate
5327	G59 Axis 7 Coordinate
5328	G59 Axis 8 Coordinate
5329	G59 Axis 9 Coordinate
5330	G59 Axis 10 Coordinate
5331	G59 Axis 11 Coordinate
5332	G59 Axis 12 Coordinate

5241	G55 Axis 1 Coordinate
5242	G55 Axis 2 Coordinate
5243	G55 Axis 3 Coordinate
5244	G55 Axis 4 Coordinate
5245	G55 Axis 5 Coordinate
5246	G55 Axis 6 Coordinate
5247	G55 Axis 7 Coordinate
5248	G55 Axis 8 Coordinate
5249	G55 Axis 9 Coordinate
5250	G55 Axis 10 Coordinate
5251	G55 Axis 11 Coordinate
5252	G55 Axis 12 Coordinate

5341	G59.1 Axis 1 Coordinate
5342	G59.1 Axis 2 Coordinate
5343	G59.1 Axis 3 Coordinate
5344	G59.1 Axis 4 Coordinate
5345	G59.1 Axis 5 Coordinate
5346	G59.1 Axis 6 Coordinate
5347	G59.1 Axis 7 Coordinate
5348	G59.1 Axis 8 Coordinate
5349	G59.1 Axis 9 Coordinate
5350	G59.1 Axis 10 Coordinate
5351	G59.1 Axis 11 Coordinate
5352	G59.1 Axis 12 Coordinate

#5221 to 5392

Work Coordinate Table Value (continued)

5261	G56 Axis 1 Coordinate
5262	G56 Axis 2 Coordinate
5263	G56 Axis 3 Coordinate
5264	G56 Axis 4 Coordinate
5265	G56 Axis 5 Coordinate
5266	G56 Axis 6 Coordinate
5267	G56 Axis 7 Coordinate
5268	G56 Axis 8 Coordinate
5269	G56 Axis 9 Coordinate
5270	G56 Axis 10 Coordinate
5271	G56 Axis 11 Coordinate
5272	G56 Axis 12 Coordinate

5361	G59.2 Axis 1 Coordinate
5362	G59.2 Axis 2 Coordinate
5363	G59.2 Axis 3 Coordinate
5364	G59.2 Axis 4 Coordinate
5365	G59.2 Axis 5 Coordinate
5366	G59.2 Axis 6 Coordinate
5367	G59.2 Axis 7 Coordinate
5368	G59.2 Axis 8 Coordinate
5369	G59.2 Axis 9 Coordinate
5370	G59.2 Axis 10 Coordinate
5371	G59.2 Axis 11 Coordinate
5372	G59.2 Axis 12 Coordinate

5281	G57 Axis 1 Coordinate
5282	G57 Axis 2 Coordinate
5283	G57 Axis 3 Coordinate
5284	G57 Axis 4 Coordinate
5285	G57 Axis 5 Coordinate
5286	G57 Axis 6 Coordinate
5287	G57 Axis 7 Coordinate
5288	G57 Axis 8 Coordinate
5289	G57 Axis 9 Coordinate
5290	G57 Axis 10 Coordinate
5291	G57 Axis 11 Coordinate
5292	G57 Axis 12 Coordinate

5381	G59.3 Axis 1 Coordinate
5382	G59.3 Axis 2 Coordinate
5383	G59.3 Axis 3 Coordinate
5384	G59.3 Axis 4 Coordinate
5385	G59.3 Axis 5 Coordinate
5386	G59.3 Axis 6 Coordinate
5387	G59.3 Axis 7 Coordinate
5388	G59.3 Axis 8 Coordinate
5389	G59.3 Axis 9 Coordinate
5390	G59.3 Axis 10 Coordinate
5391	G59.3 Axis 11 Coordinate
5392	G59.3 Axis 12 Coordinate

5301	G58 Axis 1 Coordinate
5302	G58 Axis 2 Coordinate
5303	G58 Axis 3 Coordinate
5304	G58 Axis 4 Coordinate
5305	G58 Axis 5 Coordinate
5306	G58 Axis 6 Coordinate
5307	G58 Axis 7 Coordinate
5308	G58 Axis 8 Coordinate
5309	G58 Axis 9 Coordinate
5310	G58 Axis 10 Coordinate
5311	G58 Axis 11 Coordinate
5312	G58 Axis 12 Coordinate

The system installer determines in AMP the name (or word) that is used to define the axis.

#5630

S-Curve Time per Block

This parameter is read only. The value represents the amount of time (seconds converted to system scans) for a part program block's S-Curve filter where S-Curve Acc/Dec is applied during G47.1 mode. When it is multiplied by the scan time, the product equals the amount of time required by the acceleration.

This parameter is only calculated for blocks that have programmed motion with S-Curve Acc/Dec.

#5631 to 5642

Acceleration Ramps for Linear Acc/Dec Mode

These parameters are read only. They correspond to the active acceleration ramps in Linear Acc/Dec mode. You can set these parameters by programming a G48.1 in your part program block. Control Reset, Program End (M02/M03), or G48 will reset these values to their default AMP values. For more information about programming G48.x codes, refer to chapter 18 in your 9/Series CNC Operation and Programming Manual.

5631	Axis 1 acceleration ramp	5637	Axis 7 acceleration ramp
5632	Axis 2 acceleration ramp	5638	Axis 8 acceleration ramp
5633	Axis 3 acceleration ramp	5639	Axis 9 acceleration ramp
5634	Axis 4 acceleration ramp	5640	Axis 10 acceleration ramp
5635	Axis 5 acceleration ramp	5641	Axis 11 acceleration ramp
5636	Axis 6 acceleration ramp	5642	Axis 12 acceleration ramp

#5651 to 5662

Deceleration Ramps for Linear Acc/Dec Mode

These parameters are read only. They correspond to the active deceleration ramps in Linear Acc/Dec mode. You can set these parameters by programming a G48.2 in your part program block. Control Reset, Program End (M02/M03), or G48 will reset these values to their default AMP values. For more information about programming G48.x codes, refer to chapter 18 in your 9/Series CNC Operation and Programming Manual.

5651	Axis 1 deceleration ramp	5657	Axis 7 deceleration ramp
5652	Axis 2 deceleration ramp	5658	Axis 8 deceleration ramp
5653	Axis 3 deceleration ramp	5659	Axis 9 deceleration ramp
5654	Axis 4 deceleration ramp	5660	Axis 10 deceleration ramp
5655	Axis 5 deceleration ramp	5661	Axis 11 deceleration ramp
5656	Axis 6 deceleration ramp	5662	Axis 12 deceleration ramp

#5671 to 5682

Acceleration Ramps for S-Curve Acc/Dec Mode

These parameters are read only. They correspond to the active acceleration ramps in S-Curve Acc/Dec mode. You can set these parameters by programming a G48.3 in your part program block. Control Reset, Program End (M02/M03), or G48 will reset these values to their default AMP values. For more information about programming G48.x codes, refer to chapter 18 in your 9/Series CNC Operation and Programming Manual.

5671	Axis 1 acceleration ramp	5677	Axis 7 acceleration ramp
5672	Axis 2 acceleration ramp	5678	Axis 8 acceleration ramp
5673	Axis 3 acceleration ramp	5679	Axis 9 acceleration ramp
5674	Axis 4 acceleration ramp	5680	Axis 10 acceleration ramp
5675	Axis 5 acceleration ramp	5681	Axis 11 acceleration ramp
5676	Axis 6 acceleration ramp	5682	Axis 12 acceleration ramp

#5691 to 5702

Deceleration Ramps for S-Curve Acc/Dec Mode

These parameters are read only. They correspond to the active deceleration ramps in S-Curve Acc/Dec mode. You can set these parameters by programming a G48.4 in your part program block. Control Reset, Program End (M02/M03), or G48 will reset these values to their default AMP values. For more information about programming G48.x codes, refer to chapter 18 in your 9/Series CNC Operation and Programming Manual.

5691	Axis 1 deceleration ramp	5697	Axis 7 deceleration ramp
5692	Axis 2 deceleration ramp	5698	Axis 8 deceleration ramp
5693	Axis 3 deceleration ramp	5699	Axis 9 deceleration ramp
5694	Axis 4 deceleration ramp	5700	Axis 10 deceleration ramp
5695	Axis 5 deceleration ramp	5701	Axis 11 deceleration ramp
5696	Axis 6 deceleration ramp	5702	Axis 12 deceleration ramp

#5711 to 5722

Jerk

These parameters are read only. They are only applicable to the current jerk values when S-Curve Acc/Dec mode is active. You can set these parameters by programming a G48.5 in your part program block. Control Reset, Program End (M02/M03), or G48 will reset these values to their default AMP values. For more information about programming G48.x codes, refer to chapter 18 in your 9/Series CNC Operation and Programming Manual.

5711	Axis 1 jerk	5717	Axis 7 jerk
5712	Axis 2 jerk	5718	Axis 8 jerk
5713	Axis 3 jerk	5719	Axis 9 jerk
5714	Axis 4 jerk	5720	Axis 10 jerk
5715	Axis 5 jerk	5721	Axis 11 jerk
5716	Axis 6 jerk	5722	Axis 12 jerk

#5731 to 5743

Home Marker Distance

These parameters are read only. They correspond to the current home marker distance. These parameters will contain the distance to marker calculated when the axis stopped after the home switch went false during the last homing operation.

5731	Axis 1 home marker distance	5737	Axis 7 home marker distance
5732	Axis 2 home marker distance	5738	Axis 8 home marker distance
5733	Axis 3 home marker distance	5739	Axis 9 home marker distance
5734	Axis 4 home marker distance	5740	Axis 10 home marker distance
5735	Axis 5 home marker distance	5741	Axis 11 home marker distance
5736	Axis 6 home marker distance	5742	Axis 12 home marker distance

#5751 to 5763
Home Marker Tolerance

These parameters are read only. They correspond to the current home marker tolerance. These parameters will contain the tolerance value at power turn on and will represent 3/8 of an electrical cycle of the feedback device converted to current axis programming units (inch/metric, degrees or revolutions).

5751	Axis 1 home marker tolerance	5757	Axis 7 home marker tolerance
5752	Axis 2 home marker tolerance	5758	Axis 8 home marker tolerance
5753	Axis 3 home marker tolerance	5759	Axis 9 home marker tolerance
5754	Axis 4 home marker tolerance	5760	Axis 10 home marker tolerance
5755	Axis 5 home marker tolerance	5761	Axis 11 home marker tolerance
5756	Axis 6 home marker tolerance	5762	Axis 12 home marker tolerance

28.4.4
PAL Parameters

Paramacro parameters are provided by on the control to allow a means of communicating values between the PAL program and the part program. This is done by assigning values to specific paramacro parameters or PAL flags. They are:

- **Input parameters:** Use these parameters to transfer data from PAL to the part program.
- **Output parameters:** Use these parameters to transfer data from the part program to PAL. Some applications may, however, use the output flags to transfer data from PAL to the part program as needed.

Input Flags:

There are 4-integer or 3-integer and 32-bit pattern input parameters available. The part program may only read the values assigned to these parameters; it may not write values to them. The paramacro input parameters available to the part programmer are:

- #1000 - #1031 and #1040 - #1071

These paramacro PAL parameters are used to display the binary equivalent of the integer assigned to #1032. #1000 is the first bit, #1001 is the second bit, #1002 is the third bit, and so forth up to parameter #1031 (which is the 32nd bit).

The second set of parameters, #1040 - #1071, functions the same way.

- #1032 - #1035 and #1072 - #1075

The control always interprets parameter #1032, #1033, #1034, and #1035 as integer values regardless of how they are assigned in PAL (as an integer or on a per bit basis). #1032 is the only parameter that may also be interpreted by the control on a per-bit basis using parameters #1000 - #1031. PAL may always interpret these values on either a per-bit basis or as integer values.

The second set of parameters, #1072 - #1075, functions the same way.

See the system installer's documentation for a detailed description of the use and operation of these input flags.

Output Flags:

Output flags function almost identically to input flag with one key difference. Where input flags may only be read by the part program, output flags may be both read and written to by the part program. Typically these are used only to output information to the PAL program from the part program; however, if the available number of input flags is not sufficient for a given application, the Output flags may also be used to send information to the part program from PAL.

Output flags should not be used as Input flags unless absolutely necessary. This is because the operator/programmer has the ability to inadvertently write data to the Output flags, whereas the Input flags cannot be written to from the control.

Output flags are broken into four 32-bit words. The part programmer can only assign or read the values of to these flags as integers with the exception of parameter #1132 which may be assigned as an integer or as a bit pattern. The paramacro output input parameters available to the part programmer are:

- #1100 - #1131 and #1140 - #1171

When the values of these parameters are assigned in the part program, they should be assigned values of 1 or 0 (as bit patterns). If any integer value (other than zero) is assigned to these parameters, it will be interpreted by PAL as a 1. These paramacro PAL parameters are used to pass the binary equivalent of the integer assigned to #1132. #1100 is the first bit, #1101 is the second bit, #1102 is the third bit, and so forth up to parameter #1131 (which is the 32nd bit). When a value is assigned to #1132, the values assigned to #1100 - #1131 are overwritten with the binary equivalent of #1132.

The second set of parameters, #1140 - #1171, functions the same way.

- #1132 - #1135 and #1172 - #1175

The control always interprets these parameters as integer values. #1132 is the only parameter that may also be interpreted by the part program on a per-bit basis using parameters #1100 #1131.

The second set of parameters, #1172 - #1175, functions the same way.

See the system installer's documentation for a detailed description of the use and operation of these input flags.

28.4.5 Shared Dual-Process Parameters (#7100 - 7199)

These parameters are available only on dual-process systems (see chapter 30 for details). These parameters provide a link through which common data can be passed between the different processes of the machine. The shared dual-process parameters are paramacro parameter numbers:

#7100 to #7199

These parameters are global. This means that the same set of parameters can be called by any program, macro, subprogram, or MDI program in any process.

All shared dual-process parameters are saved at power-down. This means that they retain their value even after power to the control is lost.

Synchronization Problems with Shared Dual-Process Parameters

The programmer must concern himself with timing when changing dual process paramacro parameters that are used in more than one process. When you change a shared dual-process paramacro parameter that is to be used in an additional, currently executing process, you should always use a synchronization M-code (with setup). This will help prevent timing errors that can develop when different values than expected are used due to setup block considerations. Refer to page 30-8 for details on using synchronization M-codes with shared dual-process parameters.

28.5 Assigning Parameter Values

There are 3 methods for assigning parameters. They can be assigned by:

- using arguments (only available for local parameters)
- direct assignments
- using tables (view or set common parameters, view local parameters)

Assigning Parameters Using Arguments

Arguments may be used only to assign local parameter values. System, Common, and PAL variables may not be assigned using arguments. Usually parameters assigned using an argument are variables for a macro. They are usually specific to the part currently being cut (for example, the length and diameter of a shaft in a macro that turns a shaft).

The control provides 5 sets of local parameters. The first set of local parameters (those that apply to the main program and any subprogram call) may not be assigned using arguments. The second through fifth sets may be assigned by their association to given words in an argument statement located in a paramacro calling block. Table 28.H gives a listing of arguments and their corresponding parameter numbers.

These arguments assign values to the local parameters associated with the paramacro called in the same block.

Table 28.H
Argument Assignments

(A)

Word Address	Parameter Assigned
A	#1
B	#2
C	#3
D	#7
E	#8
F	#9
H	#11
I*	#4
J*	#5
K*	#6
M	#13
Q	#17
R	#18
S	#19
T	#20
U	#21
V	#22
W	#23
X	#24
Y	#25
Z	#26

(B)

I, J, K Set #	Word Address	Parameter Assigned
1	I	#4
	J	#5
	K	#6
2	I	#7
	J	#8
	K	#9
3	I	#10
	J	#11
	K	#12
4	I	#13
	J	#14
	K	#15
5	I	#16
	J	#17
	K	#18
6	I	#19
	J	#20
	K	#21
7	I	#22
	J	#23
	K	#24
8	I	#25
	J	#26
	K	#27
9	I	#28
	J	#29
	K	#30
10	I	#31
	J	#32
	K	#33

* If more than one I, J, or K set is programmed in a block, use Table 28.H (B) for the parameter assignment.

To enter a value for a parameter # using an argument, enter the word corresponding to the desired parameter number in a block that calls a paramacro (for legal argument locations, see specific formats for calling the macro) followed by the value to assign that parameter. For example:

```
G65P1001A1.1 B19;
```

assigns the value of:

1.1 to local parameter #1 in paramacro 1001

19 to local parameter #2 in paramacro 1001

You can specify arguments as any valid parametric expression. For example:

```
G246A#100B[#500+10.0]C[SIN[#101]];
```

Direct Assignment Through Programming

This assignment method applies to Local, Common, System, and PAL parameters. You can perform direct assignment in Main, Macro, or MDI programs. Direct assignment is done by setting the parameter equal to some value in an equation using the “=” operator. For example, to assign a value of 2 to parameter number 100, simply enter the following program block:

```
#100=2;
```

The value to the left of the equals sign must contain the # sign followed by a legal parameter number. This parameter number may also take on the form of:

```
#parameter expression = parameter expression
```

Example 28.14 Calling Parameter Numbers

```
#6=1;
#144=1;
#[SIN[#6]]=1;
#[148/2]=1;
#[#6]=1;
```

All of the above can be used as legal parameter numbers. Any time that a different parameter is used between the [] symbols, the current value of that parameter is used for evaluation. For example:

```
#1=4;
#1=#1+2;
```

The net result of the above two blocks would be the assignment of a value of 6 to parameter #1.

Example 28.15
Assigning Parameters:

```
#100=1+1;  
#100=5-3;  
#100=#3;  
#100=#7+1;  
#100=#100+1;
```

You can also assign multiple paramacro parameters in a single block. In a multiple assignment block, each assignment is separated by a comma. For example:

```
#1=10, #100=ROUND[#2+#3], #500=10.0*5;
```

If you use multiple assignments in the same block, remember:

- You can enter as many assignments as can be typed into one block (127 characters maximum).
- For local and common parameters, block execution is from left to right. For example:

```
#1 = 10, #2=#1+2;
```

When executed, #1 is 10 and #2 is 12

- Once the first paramacro parameter assignment is made in a block, only assignment syntax is allowed in that block. You cannot program other information in that block, including programming a G-code. For example:

```
#1 = 19.0, G1X10;
```

will get the error message, "PARAMETER ASSIGNMENT SYNTAX ERROR"

- Only assign the same parameter a value once in each block. For example:

```
#1=5, #2=4, #1=6;
```

would cause the error message "PARAMETER ASSIGNMENT SYNTAX ERROR" to appear, since #1 is assigned a value twice in the same block (#1=5 and #1=6).

Direct Assignment Through Tables

Use this feature to view or set common parameters and view local parameters. Assignment through tables is generally used to edit common parameters.

To edit the values of the common parameters or view the local parameters, follow these steps.

1. Press the {MACRO PARAM} softkey.
(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUI CK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

2. Press the appropriate softkey for the table to be viewed. The macro parameters are separated in to 4 tables:
 - {LOCAL PARAM} softkey - Use this softkey to view the local parameters of the currently active program (unless the block look ahead has scanned an upcoming paramacro call). This table includes parameters numbered 1 - 33. Only one of the five available sets can be viewed on this screen at any one time. The local parameters reset to zero at the end of program command (M02 or M30).
 - {COM-1 PARAM} softkey - Use this softkey to view or edit the common parameters numbered 100 - 199 (available to any program, subprogram, or paramacro program). These common parameters lose their value and are reset to zero when power to the control is turned off.
 - {COM-2A PARAM} softkey - Use this softkey to view or edit the common parameters (available to any program, subprogram, or paramacro program) numbered 500 - 519. These common parameters are in backed up memory. This means that their value is retained even when power to the control is cycled off. In addition to being in backed up memory these parameters allow an alphanumeric name to be assigned to them. This alphanumeric name is only for the purpose of easy identification. It may not be used to call a common parameter in a program.
 - {COM-2B PARAM} softkey - Use this softkey to view or edit the common parameters. These common parameters, 520 to 699, are battery-backed up on the 9/240. On the 9/230, 9/260, and 9/290, parameters 520 to 999 are backed up by the supercapacitor. This means that their value is retained even when power to the control is cycled off. They differ from common 2A parameters in that they do not allow an additional name to be stored in the table with their values.

If viewing the local parameter table, do not continue to step 3. If editing one of the common parameter tables, move on to step 3.

(softkey level 3)

↑	LOCAL PARAM	COM- 1 PARAM	COM- 2A PARAM	COM- 2B PARAM		
---	----------------	-----------------	------------------	------------------	--	--

3. Select a parameter to change by moving the cursor to the desired parameter number. Note that the selected parameter is shown in reverse video. Move the cursor by an entire page by pressing the up or down cursor key while holding down the [SHIFT] key.

Pressing the {SEARCH NUMBER} softkey allows a rapid search for the desired parameter number. To use this feature to position the cursor, press the {SEARCH NUMBER} softkey. Key in the desired parameter number and press the [TRANSMIT] key. The entered parameter number is shown in reverse video.

4. Select and complete the appropriate step to alter the common parameter values. The 3 options include:
 - To replace the current value of the selected parameter, press the {REPLCE VALUE} softkey. Key in the new value for the parameter and press the [TRANSMIT] key. The old value is replaced with the value just keyed in.
 - To zero the current of a selected parameter, press the {ZERO VALUE} softkey. The message "SELECT VALUE TO ZERO AND PRESS TRANSMIT" appears. Select the parameter which is to be set to zero, and press the [TRANSMIT] key. The current value of the selected parameter is replaced with zero.
 - To zero all of the parameter values that are found in this parameter table, press the {0 ALL VALUES} softkey. The prompt "OK TO ZERO ALL VALUES? (Y/N):" will be displayed. Press the Y character followed by the [TRANSMIT] key to zero all parameter values in the table. Press the N character followed by the [TRANSMIT] key to abort the operation.
 - Note that while viewing one of the parameter screens while a program is executing, any changes to a parameter value on that screen that are made by the program will not be displayed. By pressing the {REFRESH SCREEN} softkey, any parameters that have been changed by the program will be updated to their current values.
5. If the {COM- 2A PARAM} softkey has been pressed (in step 2), additional softkeys will be available to alter the parameter name.

Select and complete the appropriate step to alter the common parameter names. The 3 options include:

- To edit an existing parameter name or enter a parameter name for the first time for a local parameter, press the {REPLCE NAME} softkey. Key in a parameter name for the parameter. A name may be up to 8 characters long and include any alphanumeric character with the exception of a few of the special symbols. After the name is keyed in, press the [TRANSMIT] key. The new parameter name is displayed next to the value of that parameter.
 - To clear a parameter name so that no name is displayed next to the parameter on the screen, press the {CLEAR NAME} softkey. The message “SELECT NAME TO CLEAR AND PRESS TRANSMIT” appears. Select the name to clear and press the [TRANSMIT] KEY. The currently selected parameter name will be deleted.
 - To clear all of the parameter names that are found on the {COMMON 2A} screen for all of the parameters, press the {CLEAR ALL NM} softkey. The prompt “OK TO CLEAR ALL NAMES? (Y/N): ” appears. Press Y followed by the [TRANSMIT] key if it is okay to delete all parameter names. Press N followed by the [TRANSMIT] key if you want to abort the delete-all-name operation.
- Note that the parameter name is used only for display purposes. It has no real function other than to permanently label a parameter value. The parameter name is retained as is the parameter value for these parameters even after power is turned off. The softkeys used to edit the parameter name operate this way:

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ZERO VALUE	REPLCE NAME	CLEAR NAME	→
↑	CLEAR ALL NM			O ALL VALUE	REFRSH SCREEN	→

Addressing Assigned Parameters

Once you assign a parameter you can address it in a program:

Example 28.16
Addressing Assigned Parameters

```
#100=5;
#105=8;
G01X#100+5 ;   Axis moves to 10.
G01x[#100+5]   Axis moves to 8
```

You can also indirectly address parameters with other parameters

Example 28.17
Indirectly Addressing Parameters

```
#100=101
#101=2.345
G01 X#[#100]; X axis moves to the
                contents of #100 which
                is #101. #101 has the
                value of 2.345.
```

Backing Up Parameter Values

You can back up the contents of COM1, COM2A, or COM2B individually, or all of these simultaneously, by using the BACKUP softkeys.

(softkey level 2)

↑	LOCAL PARAM	COM-1 PARAM	COM-2A PARAM	COM-2B PARAM		→
↑	BACKUP COM1	BACKUP COM2A	BACKUP COM2B		BACKUP ALL	→

To back up parameters:

Press this softkey:

#100 - 199	{ BACKUP COM1 }
#500 - 519	{ BACKUP COM2A }
#520 - 999	{ BACKUP COM2B }
all of the above	{ BACKUP ALL }

1. Press the appropriate BACKUP softkey.

The system prompts you for a file name.

2. Enter a name for the backup file and press [TRANSMIT] .

The system verifies the file name and backs up the selected parameters into a part program. You can restore these parameters by selecting and executing that part program.

Important: If part program calculations cause an overflow value, then the generated backup file contains an M00 and the parameter number followed by the word “OVERFLOW” as a comment.

28.6 Macro Call Commands

When a paramacro is called, execution of the currently active part program is halted, and execution is transferred to the macro program. Call paramacros in the following ways:

- Programming G65 in a part program
- Programming G66 or G66.1 in a part program
- Setting the proper AMP data can call a paramacro with the programming of specific G-, T-, S-, M-, and B-codes

You can use a paramacro call to call any program that has a program name of up to 5 numeric digits following the letter O (see chapter 10 on program names). This program must also contain an M99 end of subprogram or macro code somewhere in the program before an M02 or M30 is read. This M99 code causes control to return to the main program or restarts the paramacro if it is to be executed more than one time.

Important: The M99 code may be programmed anywhere in a paramacro program block provided no axis words are programmed to the left of the M99. Any information (other than axis words) programmed to the left of M99 is executed as part of the paramacro. Any information (including axis words) programmed in the block to the right of the M99 command is ignored.

M99X10;	X10 is ignored
X10M99;	Error is generated
M03M99;	M03 is executed

After the control has executed the macro the specified number of times (as specified by the L-word), execution is returned to the block following the paramacro call in the calling program.



CAUTION: Any edits that are made to a subprogram, or to a paramacro program (as discussed in chapter 5) that has already been called for automatic execution, are ignored until the calling program is disabled and reactivated. Subprograms and paramacros are called for automatic execution the instant that the calling program is selected as active (as discussed in chapter 7).

28.6.1 Non-Modal Paramacro Call (G65)

Use this format for calling a paramacro using the G65 command:

G65 P_ L_ A_ B_;

Where :	Is :
P	Indicates the program number of the called macro. P ranges from 1 - 99999.
L	Programs the number of times the macro will be executed. L ranges from 1 - 9999, and may be expressed as any valid parametric expression. If not specified, the control uses a default value of 1.
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 28.H). Arguments may be specified as any valid parametric expression.

The G65 command is non-modal. The macro is executed only at the time the control executes the G65 block. The control calls the macro specified by the G65 block as programmed by the P-word.

The control executes this macro until the control reaches an M99 macro return code. The macro then returns to the next unexecuted sequential block in the calling program unless the macro has not been repeated the number of times as determined by the L-word. If this is the case, the macro re-executes.

You can define the L-word or any optional argument statements in a G65 block by using any valid parametric expression. For example:

G65 P1002 L[#1+1] A[12*6] B[SIN[#101]];

28.6.2 Modal Paramacro Call (G66)

Use this format for calling a paramacro using the G66 command:

G66 P_ L_ A_ B_;

Where :	Is :
P	Indicates the program number of the called macro. P ranges from 1 - 99999.
L	Programs the number of times the macro will be executed after each motion block that follows the G66. L ranges from 1 - 9999, and may be expressed as any valid parametric expression. If not specified, the control uses a default value of 1.
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 28.H). Arguments may be specified as any valid parametric expression.

The G66 command is modal and remains in effect until canceled with a G67 block. The macro programmed by the P-word in the G66 block is not executed when the G66 block is read. The control delays macro execution to any block following the G66 command that contains a motion command.

When the control encounters a motion block (even if this block is contained in a different macro) following the G66 block, it executes the motions called for by that block first. After that block has been executed, the control then calls the macro specified by the G66 block.

The control executes this macro until the control reaches an M99 macro return code. The macro then returns to the next unexecuted sequential block in the calling program unless the macro has not been repeated the number of times as determined by the L-word. If this is the case, the macro re-executes.

Each time that a specific macro is called by a motion command, it is executed the number of times programmed with the L-word. All local variables remain at their current value throughout the program unless replaced, the control is reset, E-STOP is reset, or the control encounters an M02 or an M30 code in a program.

An L-word programmed with a G66 macro call cannot be replaced without reprogramming the entire G66 block with the new L-word. An L-word is active each time the macro is called by the main program and causes the macro to be executed the number of times programmed with L.

You can define the L-word or any optional argument statements in a G66 block by using any valid parametric expression. For example:

```
G66 P1002 L[#1+1] A[12*6] B[SIN[#101]];
```

Unlike non-modal macro calls, the G66 macro call repeats automatically after any axis move until cancelled by a G67 block. This also applies to nested macros. When the control begins execution of the nested macro 1002 in the program below, each axis move in the nested macro also calls for the execution of the macro 1001.

Example 28.18
Modal Macro Call

```
N0100G66P1001;
N0200G65P1002;
```

In Example 28.18, after the complete execution of the macro 1002, the macro 1001 is called. Any motion blocks in macro 1002 will cause macro 1001 to be executed.

Example 28.19
Modal Macro Operation

```
(MAIN);

O1000;
N010G90;
N020G66P1001L2A1. 1;      Parameter #1 is set at 1.1 in macro 1001.
N030X1;                   X Axis is moved 1 unit and then macro 1001 is called and
                           executed 2 times.
N040Z. 25                 Z Axis is moved .25 units and then macro 1001 is called and
                           executed 2 times.
N050G66P1002A2;          Parameter #1 is set at 2. in macro 1002.
N060X1. ;                X axis is moved 1 unit then macro 1002 is called and
                           executed once.
N070G67;                 Macro 1002 is canceled.
N090G67;                 Macro 1001 is canceled.
N100M30;
```

```
(MACRO);

O1001;
N200Z#1;                  Z Axis moves an amount equal to the current value for
                           parameter #1
N210#1=1. 7              Parameter #1 for macro 1001 is set at 1.7.
N220M99;                  Macro end.
```

```
(MACRO);

O1002;
N300Z#1;                  Z Axis moves an amount equal to the current value set
                           parameter #1 (in this case always 2 units). Macro 1001 is
                           called and executed twice.
N310M99;                  Macro end.
```

28.6.3 Modal Paramacro Call (G66.1)

Important: When the control executes block N040, the original value as set in block N020 for parameter number 1 is ignored, and the most current value (1.7) is used. The first time macro 1001 is executed, Z moves 1.1 units. The second time macro 1001 is executed, Z moves 1.7 units.

Use this format for calling a paramacro using the G66.1 command:

G66.1 P_ L_ A_ B_;

Where :	Is :
P	Indicates the program number of the called macro. P ranges from 1 - 99999.
L	Programs the number of times the macro will be executed. L ranges from 1 - 9999, and may be expressed as any valid parametric expression. If not specified, the control uses a default value of 1.
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 28.H). Arguments may be specified as any valid parametric expression.

The G66.1 command is modal and is executed in the same manner as the G66 with these exceptions:

- The macro programmed by the P-word in the G66 block is not executed when the G66 block is read, whereas the macro programmed by the G66.1 is executed when G66.1 is read.
- The macro is executed in any and all blocks following the G66.1, not just after motion blocks, except for paramacro command blocks such as assignment, goto, etc.
- Axis motion cannot be generated by normal program blocks. Axis motion can be generated only in the program called by G66.1.
- The following words, when programmed after the G66.1 block, are used as argument assignments:

N: when programmed after a word other than N or O, is used as assignment #14.

G: The last G-code programmed in a block will be used as an argument statement for parameter #10. All other G-codes are interpreted as normal.

L: Assigns value to parameter #12

P: Assigns value to parameter #16

All other argument assignments are interpreted as listed in Table 28.H.

The L-word or any optional argument statements following a G66.1 can contain any valid mathematical expression. For example:

```
G66. 1 P1002 L[#1+1] A[12*6] B[SIN[#101]];
```

Example 28.20
G66.1 Macro Operation

N0100G90G17G00;	
N0110G66. 1P9400;	Macro 9400 is executed.
N0120G91G18G01;	G91 and G18 become effective, 01 is assigned to parameter #10, macro 9400 is executed.
N0130G03X1. ;	03 is assigned to parameter #10, 1. is assigned to parameter #24, macro 9400 is executed.
N0135;	Macro 9400 is executed.
N0140G67;	Macro 9400 is deactivated.
N0150M30;	program end.

Any time the macro is called (while executing the G66.1), the L-word programming the number of repetitions is in effect. Any attempt to reprogram an L-word outside of a G66.1 block is interpreted as an argument assignment for parameter #12.

Important: When nesting a macro (any macro including G66.1) within a G66.1 macro, the outer G66.1 macro is executed after each individual block of the nested macro, except for paramacro command blocks such as assignment, goto, etc.

Example 28.21
Nesting a Modal Macro

```
N0100G66. 1P1001;
N0200G65P1002;
```

After the execution of each individual block within the macro 1002, the macro 1001 is called.

You can define the L-word or any optional argument statements in a G66.1 block as any valid parametric expression. For example:

```
G66. 1 P1002 L[#1+1] A[12*6] B[SIN[#101]];
```

28.6.4 AMP-defined G-Code Macro Call

Use this format for calling an AMP-defined macro:

G_ A_ B_;

Where :	Is :
G_	Programs an AMP-defined G-code command (from G1 to G255.9).
A- Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 28.H). Arguments may be specified as any valid parametric expression.

An AMP-defined G-code macro is a G-code that is specified in AMP by the system installer. When one of these AMP-defined G-codes is executed in a part program, execution is transferred to the macro with the program number associated to that G-code.

G-code values for paramacro calls may range from 1 to 255.9. The system installer may define a maximum of 25 AMP-defined G-codes to call specific paramacro programs. The paramacro program name called with the AMP-defined G-code is a program number from 1 to 8999 or 9010 to 9019. Refer to the system installer's documentation for details.

Important: The system installer may disable the use of AMP-defined G and M-code macro calls when in MDI mode. Refer to the system installer's documentation to determine if this feature is functional in MDI.

AMP-defined G-code macros can be executed as either modal or non-modal macros as selected in AMP. If selected as modal, they can be execute using either G66 modality or G66.1 modality. This modality type for AMP defined G-codes is also determined by the system installer in AMP.

Any optional argument statements following an AMP-defined G-code may contain any valid parametric expression. For example:

G255A[12*6]B[SIN[#101]];

In a part program, if more than one digit is entered after the decimal point, the value will be truncated. For example, 231.18 is 231.1, and 231.14 is 231.1.

Important: Certain AMP-defined G-code Macro calls cannot be called by any other AMP-defined macro call.

28.6.5
AMP-Defined M-Code Macro
Call

Use this format for calling an AMP-defined M-code macro:

M255 A_B_

Where :	Is :
M255	Programs an AMP-defined M-code command.
A- Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 28.H). Arguments may be specified as any valid parametric expression.

These macros are executed only as non-modal macro.

The term AMP-defined M-code macro comes from the fact that the M-code that calls a specific macro program is specified in AMP by the system installer. The system installer may define M-codes that will call paramacro programs with program names ranging from 9001 to 9009. Refer to the system installer's documentation to determine what M-codes are used to call what paramacro program name.

When one of these AMP assigned M-codes is specified in a part program, execution is transferred to the macro associated to that specific M-code.

M-code values for paramacro calls may range from -1 to 999. The system installer may define a maximum of 9 AMP-defined M-codes to call specific paramacro programs.

Important: Note the system installer may optionally disable the use of AMP-defined G- and M-code macro calls when in MDI mode. See the system installer's documentation to determine if this feature is functional in MDI.

28.6.6
AMP-Defined T-, S-, and
B- Code Macro Call

Use this format for calling an AMP-defined T-, S-, or B-code macro:

T t ;
S s ;

or

B b ;

Where :	Is equal to the value assigned to parameter:
t	#149
s	#147
b	#146

Important: Programming arguments are not allowed with the AMP-defined T-, S-, or B-code macro calls.

These macros are executed only as non-modal macro.

The execution of the T-, S-, or B-code macro calls is the same as M-code macro calls with the following exceptions:

- the parameter # referenced when called
- the macro program called

T calls macro 9000
S calls macro 9029
B calls macro 9028

In order for the T-, S-, or B-words to call up a macro program, these prerequisites must be met:

1. The value following the word must be equal to the value stored for the specified parameter #.

For example:

T14;

The value of 14 must have been previously stored as the value for the parameter #149.

2. An AMP flag for that specific word must be turned on by the system installer to allow that word to call a macro.
3. The value for an AMP-defined T-, S-, or B-code command has the same format and range as an ordinary T-, S-, or B-code.

28.6.7 Nesting Macros

Nesting occurs when one program calls another program. A subprogram called by a main program is an example of nesting. (The “nested” program is the program called.)

Nesting applies to macros as well. When the main program calls a macro, the macro is said to be on nesting level 1. If this macro in turn calls another macro, this second macro is said to be in nesting level 2. Macros may be nested up to a maximum of 4 levels. However, if the maximum number of nested paramacros (4) is combined with up to 4 subprograms that end with M98, a maximum of 8 levels of nesting can be programmed.

What is not counted as an additional nested level? When a lower nested macro with a modal feature forces a higher nested macro to call it, the number of nested levels does not increase. Nor does it increase when a subprogram is called using M98.

Precautions must be taken when attempting to nest AMP assigned macro calls since many combinations of these calls may not be valid. The system installer determines in AMP the functionality of the AMP-defined macro call when nested. These two options are available (see the system installer’s documentation to determine which applies to your system):

- **Works as a macro call** - When “works as a macro call” is selected, G-, M-, T-, S-, or B-code macro calls that are nested and called by other G-, M-, T-, S-, or B-code macro calls allow nesting as shown in Table 28.I.

Table 28.I
Works as a Macro Call

CALLING PROGRAM	TYPE OF MACRO NESTED ¹			
	G65, G66, or G66.1	AMP-G	AMP-M	AMP-T S or B
G65, G66 or G66.1	Yes	Yes	Yes	Yes
AMP G-code	Yes	No	Yes	Yes
AMP M-code	Yes	Yes	No	No
AMP-T-, S-, or B-code	Yes	yes	No	No

¹ What Yes/No means:

Yes -- the macro type across the top row may be called from the macro type down the left column.

No -- the macro type across the top row may not be called from the macro type down the left column. When this nesting is attempted, the control will execute any other operation that would normally be performed by that G-, M-, T-, S-, or B-code (as defined by the system as a standard code, PAL, or some other AMP feature) and the paramacro call normally made by that code is not performed.

- **Works as the system-defined code** - When “works as the system defined code” is selected, G-, M-, T-, S-, or B-code macro calls that are nested and called by other G-, M-, T-, S-, or B-code macro calls allow nesting as shown in Table 28.J.

Table 28.J
Works as the System-defined Code

CALLING PROGRAM	TYPE OF MACRO NESTED ¹			
	G65, G66, or G66.1	AMP-G	AMP-M	AMP-T S or B
G65, G66 or G66.1	Yes	Yes	Yes	Yes
AMP G-code	Yes	No	No	No
AMP M-code	Yes	No	No	No
AMP-T-, S-, or B-code	Yes	No	No	No

¹ What Yes/No means:

Yes -- the macro type across the top row may be called from the macro type down the left column.

No -- the macro type across the top row may not be called from the macro type down the left column. When this nesting is attempted, the control will execute any other operation that would normally be performed by that G-, M-, T-, S-, or B-code (as defined by the system as a standard code, PAL, or some other AMP feature) and the paramacro call normally made by that code is not performed.

Important: If the nesting is invalid (No in one of the above tables), the control will execute the programmed code as some other function (as defined by the control as a standard code, PAL, or some other AMP feature) and the macro call is not made. If no other function is found that uses that G-, M-, T-, S-, or B-code, the control generates an error.

The rule to follow for Table 28.J is that an AMP-assigned macro may not call an AMP-assigned macro.

For example, if the calling program is an AMP-assigned M-code macro, then G65, G66 and G66.1 macro calls are allowed; but no other types of macro calls are allowed, including an M-code macro.

28.7 Macro Output Commands

Paramacros have the ability to force the control to output selected parameter values to a device attached to the RS-232 port. This is done by using one of these commands in any macro or MDI program block:

POPEN
PCLOS
BPRNT
DPRNT

Upon execution of these commands in a program block, the control outputs the appropriate data (in the form of part program blocks such that they may be output to an RS-232 compatible device).

POPEN

This command affects a connection to the output device by sending a DC2 control code and a percent character “%” to the RS-232 interface. This command must be specified prior to outputting any data. After this command, the control outputs any following program blocks including the parameter values that are used in them.

The format for a POPEN block is:

POPEN;

PCLOSE

This command disconnects the control from the output device by sending a DC4 control code and a percent character “%” to the RS-232 interface. This command must be specified after all data have been output. This command only executes provided that the corresponding POPEN command has previously been executed.

The format for a PCLOSE block is:

PCLOS;

BPRNT

This command initiates the outputting of a variable number of parameter values in binary format. An end of block character is output at the completion of outputting all of the specified values. This command will not be executed if the POPEN command has not been issued.

The format for the BPRNT block is:

BPRNT [s #p[d]...];

Where :	Is :
s	is any alpha-numeric string of characters, including all letters, +, -, *, and / symbols. Note that the * is output as a space character. This string is optional and does not need to be programmed in the block.
#p	This is any valid parameter number. Note that the parameter number must have the # sign before it.
d	This indicates the number of digits after the decimal point to be output.

There may be as many S and #P in a block as desired provided that the length of the block does not exceed the maximum block size.

Example 28.22
Sample of a BPRNT Block

```
BPRNT[INSTALL*TOOL*NUMBER*1#123[4]*PRESS*CYCLE*STOP**#234[2]];
```

Example 28.22 would yield an output equal to the character strings with the * symbols being converted to spaces and the parameter values for parameters #123 and #234. The value of the parameter is output in binary as a 32-bit string with the most significant bit output first. Negative values are output in 2's complement.

Example 28.23
BPRNT Program Example

```
#123=0. 40936;
#124=- 1638. 4;
#10=12. 34;
POPEN;
BPRNT[ _____ ];
BPRNT[ COMMENT*HERE*X#123[ 3]**Y#124[ 3]**Z#10[ 0] ];
BPRNT[ _____ ];
PCLOS;
M30;
```

The output from Example 28.23 would be:

```
COMMENT HERE X0. 409 Y1638. 400 Z12.
```

If the output went to a punched paper tape, it would be formatted in ISO code.

DPRNT

This command initiates the outputting of a variable number of parameter values in decimal format. An end-of-block character is output at the completion of outputting all of the specified values. This command will not be executed if the POPEN command has not been issued.

The format for the DPRNT block is:

```
DPRNT [ s #p[i d]... ];
```

Where :	Is :
s	Is any alpha-numeric string of characters, including all letters, +, -, *, and / symbols. Note that the * is output as a space character. This string is optional and does not need to be programmed in the block.
#p	This is any valid parameter number. Note that the parameter number must have the # sign before it.
i	This indicates the number of digits before the decimal point to be output.
d	This indicates the number of digits after the decimal point to be output.

Important: The sum of i + d cannot exceed 8.

There may be as many S and #P in a block as desired provided that the length of the block does not exceed the maximum block size.

Example 28.24
Sample of a DPRNT Block

```
DPRNT[ I NSTALL*TOOL*#123[ 53] *PRESS*CYCLE*STOP**#234[ 20] ];
```

Example 28.24 would yield an output equal to the character strings with the * symbols being converted to spaces and the parameter values for parameters #123 and #234. The value of the parameter is output as a string of decimal digits. A decimal point is not output if there are no decimal digits. The most significant digit is output first.

The output may be either ASCII/ISO or EIA format as determined in AMP by the system installer. Leading zero suppression, and/or a leading plus sign can also be set by AMP parameters.

Example 28.25
DPRNT Program Example

```
#12=123. 45678;  
#4=- 98. 7;  
#30=234. 567  
POPEN;  
DPRNT[ _____ ]  
DPRNT[ COMMENT*HERE*X#12[ 53] *Y#4[ 53] *T#30[ 20] ];  
DPRNT[ _____ ]  
PCLOS;  
M30;
```

The output from Example 28.25 would be:

```
COMMENT HERE X23. 456 Y-98. 700 T35
```

END OF CHAPTER

Program Interrupt

29.0 Chapter Overview

This chapter describes the program interrupt feature. This feature lets you execute a subprogram or paramacro program while some other program is executing. This subprogram or paramacro is executed when PAL receives an interrupt signal (usually through the use of some switch triggered by the operator or one of the axes). The interrupt program can be executed even mid-block during a programs execution.

Use this table to find the information in this chapter:

Information on:	On page:
Enabling and Disabling Interrupts (M96/M97)	29-1
Interrupt Request Considerations	29-4
Interrupt Types	29-5
The Interrupt Program	29-8

29.1 Enabling and Disabling Interrupts (M96/M97)

Interrupts are enabled or disabled on the control by two modal M codes. These M codes are determined in AMP by the system installer. This manual assumes the following values for these M codes (these are the default values in AMP):

M96 **Enables Program Interrupts**

M97 **Disables Program Interrupts**

When program interrupts are enabled (M96) the correct interrupt signal to PAL will cause an interrupt program to be executed. When program interrupts are disabled (M97) an interrupt signal is ignored.

The format for these M codes is:

M96L__P__;

M97L__;

Where :	Selects:
L	the type of interrupt and the signal that will call the interrupt. L ranges from 0 - 3.
P	the interrupt program. P is followed by a 5 digit non-decimal program name.

An error is generated if anything other than an N word, a P or L word, a block delete /, or a comment character is programmed in the M96 or M97 block.

An interrupt M code M96 or M97 may also be programmed within a interrupt program. If this is the case the interrupt does not become enabled/disabled until the interrupt currently being executed is completed and execution is returned to the main program.

Selecting the Type of Interrupt

Two types of interrupt programs may be enabled or disabled with these M-codes. These two types are described in section 29.3 . You can use up to 4 signals to PAL (switches) to call interrupt programs. The system installer determines what switch corresponds to which type of interrupt in PAL. Refer to documentation prepared by the system installer for details on the types available and switches used to control them for a specific machine application.

The M code determines which type and which switch signal is enabled or disabled by programming an L-word with the M code. There are 4 L-words:

L0	Interrupt type 1	Switch 0 triggers this.
L1	Interrupt type 2	Switch 1 triggers this.
L2	Interrupt type 2	Switch 2 triggers this.
L3	Interrupt type 2	Switch 3 triggers this.

Program these L-words in each M96 or M97 block. Not programming the L-word causes the control to assume an L-word of L0 has been programmed. Refer to documentation prepared by the system installer for the location and operation of the 4 switches.

If more than one L-word is programmed in a block, the right most L-word is the one that is used for that block. Other L-words in that block are ignored.

Selecting an Interrupt Program

Any legal subprogram or paramacro may be selected as an interrupt program (refer to the section in chapter 10 on subprograms or chapter 28 for paramacros). For a program to be used as an interrupt program it must have a program name of 5 numeric characters following an O address (see the section on program names in chapter 10). This interrupt program must contain an M99 block as the last block in the interrupt program. This M99 block has all the same restrictions as an M99 block for a subprogram as discussed in chapter 10.

The subprogram or paramacro program is assigned to a particular type of interrupt by programming a P word in the M block that enables the interrupt (M96 in this manual). When selecting a program with a P word note that only the numeric value of the program name is entered, the letter O is omitted. For example, programming:

```
M96LOP11111;
```

would enable the program O11111 as a type 1 interrupt and allow it to be executed when switch 0 sends a signal to PAL. If the program called with the P word does not exist, the control will generate an error when the switch that activates the program is activated. Note the P word is not programmed in the disable M code block (M97 in this manual).

Example 29.1 Enabling and Disabling the Interrupt Features

N1M96LOP11111;	Enables program O11111 as a type 1 interrupt and allows it to be executed when the interrupt signal from switch 0 is received.
N2M96L1P12345;	Enables program O12345 as a type 2 interrupt and allows it to be executed when the interrupt signal from switch 1 is received.
N3M96L3P11111;	Enables program O11111 as a type 2 interrupt and allows it to be executed when the interrupt signal from switch 3 is received. Note that this is the same program as selected for type 0 interrupts.
N4M97L3;	Disables any interrupt program that is called by switch 3. Any signal to execute an interrupt from switch 3 is ignored after this block is executed unless reactivated with a M96L3 block.
N5M96LOP22222;	Alters the program that is called for the interrupt with switch 0. The new program called when the interrupt signal is received is O22222.
N6M97L0;	Disables interrupt switch 0.
N7M97L1;	Disables interrupt switch 1.

Important: All program interrupts that are enabled in a part program are automatically disabled by the control when either an end of program (M02 or M30) block is read, a new program is selected as active, or a control reset is performed.

29.2 Interrupt Request Considerations

When using system interrupts, take into consideration:

- The system installer can determine in AMP if a signal to execute an interrupt program is delayed until the end of a currently executing block, or if the interrupt is executed immediately when the signal is received.
- The system installer can determine in AMP whether an interrupt program request is recognized when an interrupt switch is turned on, or only when the switch makes the transition from off to on. This is to help prevent the accidental execution of an interrupt program if a switch is inadvertently left on when a program begins execution.
- Interrupt programs should normally be disabled during thread cutting. The execution of an interrupt program during a threading pass may cause undesired results.
- The system installer has the option of writing PAL to allow the use of 4 interrupt signals (4 switches). One of these signals may call a type 1 interrupt; the other three call a type 2 interrupt.
- Before an interrupt signal is recognized by the control it must first have been enabled by programming a M96 followed by the correct L word for that signal.
- Interrupt programs may only be executed when the control is in the automatic mode. Interrupt requests that occur during MDI or manual modes are ignored.
- Cutter compensation is temporarily cancelled during the execution of an interrupt program. It is reactivated upon completion of the interrupt program. No entry or exit move from compensation is performed.
- Interrupts that are requested when the control is in E-STOP are ignored regardless of whether the interrupt is enabled or not.
- Interrupts can only be executed when the control is in the <CYCLE START> state. If a request for an interrupt is made when the control is in <CYCLE STOP> or cycle suspend the interrupt request is still recognized. The interrupt program will be executed when a <CYCLE START> state becomes active again.

- If an interrupt occurs during a block retrace, the interrupt will be performed. The block retrace however will be aborted at that point and no further retrace will be allowed. Block retrace will, however, still be able to return any moves that have already been retraced before the interrupt occurred.
- During the execution of a milling cycle, if the interrupt is a delay type (executed at the completion of the currently executing block), the control will execute the interrupt after all motions generated by that block are completed. If the interrupt type is immediate (executed as soon as the interrupt signal is received), the control interrupts the currently executing path.

29.3 Interrupt Types

There are two types of interrupts, type 1 and type 2. These are selected by the L-word in the M96 block. L0 activates type 1 interrupts and L1, L2, and L3 activate type 2 interrupts. Type 1 and type 2 interrupts are shown in Figure 29.1 and Figure 29.2.

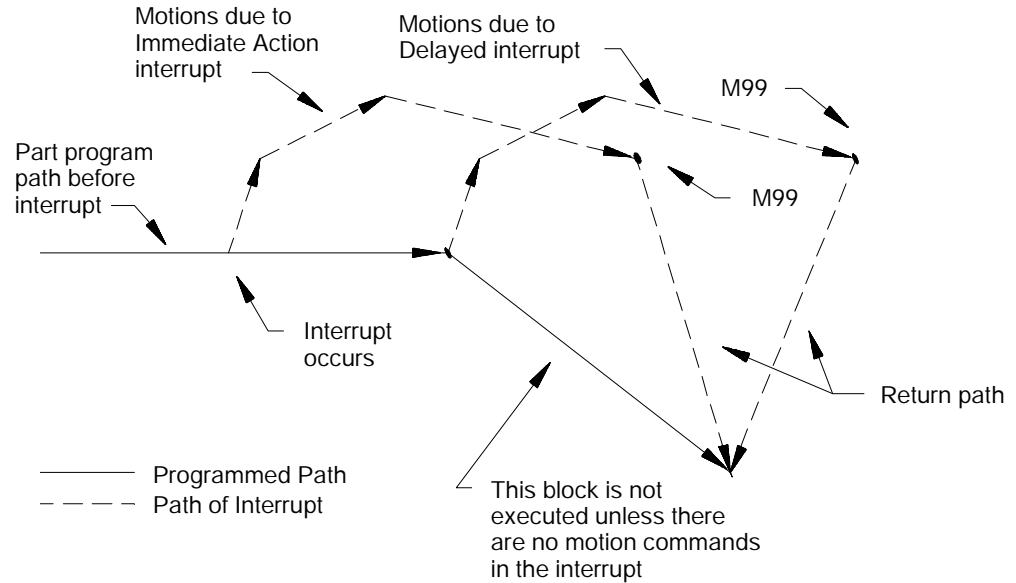
The key difference between a type 1 and a type 2 interrupt is the tool path that is taken when the return from interrupt is made as programmed with an M99 in the interrupt program.

Type 1 Interrupts

If no axis motion is generated by the interrupt program then the control executes the interrupt program and then continues executing the part program as normal regardless of the location that the interrupt program was executed.

If axis motion is generated by the interrupt program then the control returns the tool to the endpoint of the next fully unexecuted block and continues executing the part program from this point.

Figure 29.1
Type 1 Interrupt

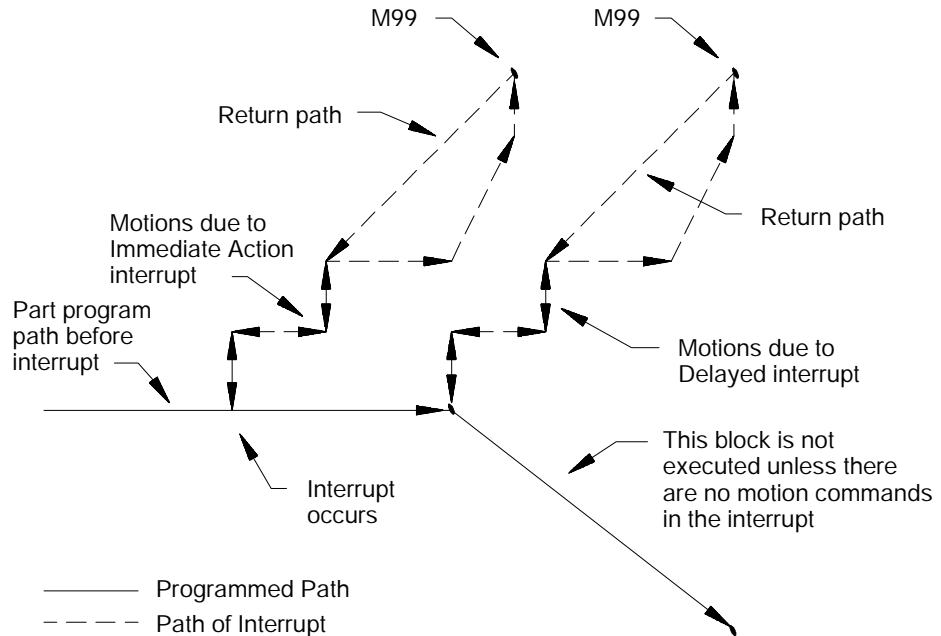


CAUTION: If the interrupt is executed in the middle of a blocks execution, and there is axis motion in the interrupt program than the control will never reach the end point of the interrupted block, execution will transfer to the end point of the following block. This applies to type 1 interrupts only.

Type 2 Interrupts

The control returns the tool to the point in the program where it was when the interrupt was performed by using type 2 interrupts. Normally the first 4 linear moves (G00 or G01) in the interrupt program are remembered (this may be altered by programming a specific M code as discussed later). If a non-linear (G02 or G03) circular move is performed as one of the first 4 blocks of the interrupt, the control only remembers the moves up to that block. If that block is the first block in the interrupt then the start point of that block is all that is remembered. These remembered blocks are retraced after the M99 code is read by the control in the interrupt program. This returns the cutting tool to the location in the program where program execution was interrupted.

Figure 29.2
Type 2 Interrupt



It is possible to alter the number of blocks that the control re-executes in reverse when returning to the start position of the interrupt. The number of return blocks is normally 4; however, it may be altered by programming one of these M codes:

M900 - zero blocks retraced

M901 - one block retraced

M902 - two blocks retraced

M903 - three blocks retraced

M904 - four blocks retraced

These M codes may be programmed in any block in the main program that is before the interrupt program is executed. They may also be programmed within an interrupt though the M code will not go into effect until execution is returned to the main program. If not programmed, the default is 4 blocks. The default is automatically reset at the end of program commands (M02 or M30).

The number of retrace blocks as set with this M code is the same for all active or inactive interrupts. If an interrupt is enabled after this M code is programmed it will take on the number of retrace blocks as programmed with this M code.

When the return from interrupt is executed (M99 in the interrupt) the control will generate a linear move to the end point of the last remembered move for retrace. Then the moves are retraced returning the tool back to the start point of the interrupt. Note this may not be the same location in the main program if a different tool offset has been activated.

29.4 The Interrupt Program

When you intend to use a program as an interrupt program, remember:

- Any modal data (G codes, feedrates, spindle speeds, coordinate system offsets, etc.) contained in the main program are carried into the interrupt program. Any changes made to this modal data within the interrupt will be effective only in the interrupt program; changes are not carried back into the main program when the interrupt is completed. This does not include tool or tool offset data that is changed in the interrupt. Any tool or tool offset changes will be carried back into the suspended main part program.
- The system installer can determine if an interrupt program is to be called as a paramacro when executed, or a subprogram when executed.
- If it is to be called as a paramacro, remember that this assigns a new set of local parameters for the interrupt. If it is to be called as a subprogram, the same set of local parameters that applied to the interrupted program apply to the subprogram.
- If an interrupt is chosen as a macro program, it may not be a macro that requires the assignment of local variables in the calling block (can not require an argument).
- Macro type interrupts are always called as the G65 non-modal type. G66 and G66.1 modal types may not be called. Refer to the chapter on paramacros for details on the G65 type macros.
- The interrupt program must contain an M99 block. Any axis motion commands that are to the left of the M99 code in the block will result as an error. Other programming commands to the left of the M99 code in the block will be executed. Any characters to the right of the M99 code in the block are ignored.

- If using a type 2 interrupt (L1, L2, or L3), remember that the control remembers up to the first 4 blocks in the program and uses these to retrace its moves back to the starting point of the interrupt program. The control remembers up to 4 of the first moves or until a circular block is executed. For details, see section 29.3 on interrupt types.
- The interrupt program may contain a milling cycle in the interrupt.
- These G codes are illegal in an interrupt program:
 - cutter compensation G codes G40, G41, and G42
 - coordinate system offsets G52, G92, and G92.1
- Any inherent modality from the main program (such as a milling cycle, or an active modal paramacro) will be temporarily canceled during the execution of a interrupt program.
- Only one interrupt may be executing at any given time. All 4 may be active at once, but only one may be executing. This means that an interrupt may not be executed during the execution of another interrupt.

END OF CHAPTER

Using a 9/Series Dual-processing System

30.0 Chapter Overview

Read this chapter to learn general information related to programming and operating a dual-processing system. Major topics in this chapter cover:

Topic:	On page:
Definition of a dual-processing system	30-1
Operating a dual-processing system	30-2
Synchronizing multiple part programs	30-7
Spindle control for dual-processing	30-12
Using interference checking	30-12

30.1 Defining of a Dual-processing System

A dual-processing system is basically two machine tools in one. One 9/Series control is capable of executing two separate part programs, controlling separate machining axes. Offsets, tool management, and axis feedrates are independent for each program. Each of these two independent systems is called a process. The 9/Series control is capable of running two processes simultaneously.

You can configure the 9/Series dual-processing mill to run on a wide variety of machines. For dual-processing to function correctly, your system installer must make certain AMP and PAL configuration considerations. Refer to the 9/Series AMP and 9/Series PAL reference manuals or your system installer's documentation for details.

This chapter focuses on a 7-axis horizontal mill. Many other applications, like a completely separate twin-milling machine are possible and easily configured by your system installer. Operation of these other applications must be interpreted from the 5-axis gantry application presented in this manual.

30.2 Operating a Dual-processing System

Dual-process systems operate almost exactly the same as their single-process counterparts. Each process functions as an independent 9/Series control.

With the exception of shared dual-processing paramacro parameters and shared axes, there is little shared data between processes. Each process has its own offset tables, programmable zone tables, and paramacro parameters. Your system installer:

- determines how your system operates in PAL. For example, pressing <CYCLE START> could be written in PAL to start both processes, start the selected active process, or separate <CYCLE START> buttons could be installed for each process. Most operations can be duplicated on a per process basis.
- establishes priorities for your processes in AMP. The priority of a process determines how much time the control spends executing blocks in one process relative to another process. Your system can be set to give one process almost exclusive use of the control's time with the other process executing only when time becomes available. Refer to your system installer's documentation for more information on the priority configuration in your system.

Selecting the Active Process

When you operate a 9/Series dual-process system, multiple processes exist that are usually controlled by the same MTB and operator panels. Your system installer can install separate MTB panels for each process on some machines. An active process must be selected to determine which process is currently being controlled by the operator panel. The line immediately above the softkey names on the CRT indicates which process is currently active. The currently active process appears in reverse video on that line.

You cannot switch the active process while you use the digitize feature, a tool path or QuickCheck graphic display, or within an active program search operation. If you attempt to switch the active process while using one of these features, the control displays an error message. Select an active process by using one of these methods:

Method:	Description:
[PROC SELECT] key	found on the operator panel next to the [TRANSMIT] key. Each time you press the [PROC SELECT] key, the currently active process changes as shown on the CRT.
System installer's key	your system installer can control which process is active at a given time through PAL. Refer to your system installer's documentation for details on the location and operation of this feature.
[SHIFT] key & {• } softkey on the end of the softkey rack	you can toggle between processes each time you press these keys. This functions identically to the [PROC SELECT] key.

All functions performed on the operator panel (this includes softkey functions), and position displays (except for split-screens) are representative of the currently active process. On split-screens, the CRT display shows position displays and process information for both processes.

Split-Screen Displays

All of the 9/Series position display screens are available as split-screens. A split-screen is a screen that shows both processes at the same time on the CRT, regardless of which process is active. Refer to page 8-21 for details on using the split-screen feature.

The Part Program Directory

Both processes share the system's part program directory. They both have access to the same programs. They can be activated from both processes. The system directory shows which process has programs active or being edited at any given time. An "A1" or "A2" next to the program name shows which process currently has the program active. An "A12" next to the program name shows the program is currently active in both process 1 and process 2.

If you execute a part program from a tape reader, each process must have its own tape reader. However, you can use one tape reader if you load the part program into memory for one process.

Editing a Part Program

An “E” next to the program name on the part program directory screen indicates that the program is currently being edited. Only one program can be open for editing at a time. You cannot edit programs in more than one process at the same time.

You cannot edit a program that is currently active (selected to run) in a different process. If you select process 2 and attempt to edit the active program in process 1, the control displays an error.

If the same program is active in both processes, you cannot edit the program unless you deactivate it in one of the processes.

QuickView prompts are available when you write your program (see page 5-17). These prompts help you by describing (often graphically) the programming of different operations on the 9/Series control. QuickView is available only to one process at a time. QuickView cannot be activated if either process currently has either QuickCheck Graphics or Active Program Graphics turned on.

QuickView does not necessarily prompt you for the currently active process selected on the control. Independent QuickView process-select softkeys let you select a process for programming. Changing the process with this key only changes the axes names and displays to the newly selected process for QuickView’s prompting. It does not change the actual currently active process on the control.

When you have a dual-process mill, additional softkeys on QuickView softkey level 4 are available. These softkeys are named after each of the process names. The currently selected process for QuickView prompts has its softkey shown in reverse video. You must still select a plane for each process for the QuickView feature as described in chapter 5.

Graphics

Active graphics or QuickCheck graphics can be active in only one process at a time. You cannot switch from graphic to graphic without disabling graphics in one process and then moving into the other process. You can use QuickCheck as a program “syntax only” checker (no graphics) in both processes at the same time.

When you plot the path of a shared axis on a graphics screen the modality of the process running graphics determines the line type being plotted (rapid or cutting move). For example if the X axis is shared and is performing a rapid move in process two, and graphics is running in process 1, process one’s current mode determines if the graphics display shows a dashed rapid or a solid cutting line. The actual axis speed is not affected by the graphics plot line type.

Error Messages

The control displays error messages on the screen for only the currently active process (except on split-screens). The name of the currently active process flashes in reverse video if an error occurs in another process. Change to the appropriate process to display the current errors for that process.

On a split-screen display, the control displays the full error messages for both processes, regardless of which process is currently active.

The control keeps separate error message logs for each process. The error logs list the errors in sequential order.

Refer to page NO TAG for more details on error messages.

Single-Digit Feedrates


Both processes share one table of single-digit feedrates. The control looks to the single-digit feedrate table when it encounters a single-digit F-word during block execution. The feedrate in this table that corresponds to the single digit becomes the active feedrate. You cannot define different single-digit feedrate tables for each process.


The rapids for each individual axis determine the feedrate used for F0. Your system installer determines the rapid for each axes.

Refer to page 3-23 for information on single-digit feedrates.

Reset Operations

Dual-process systems have a process reset operation, in addition to the normal block reset and control reset functions. These reset operations work as follows:

If you want to perform a:	Press:	The control will:
Block Reset	[RESET]	Skip the currently active block in the currently selected process (see chapter 2).
Control Reset	[SHIFT] + [RESET]	Return the control to the default settings, clear any program errors, and cancel MDI commands in all processes (see chapter 2). 
Process Reset	[SHIFT] + [PROC SELECT]	Return the program of the currently selected process to the first block. Does not affect the additional processes (only works on programs executed from control memory).

 If only one (or both) processes are still executing their part program when a control reset is performed, only processes not currently executing a part program will have a control reset performed. A message will occur in the processes executing the program stating that a control reset is not allowed on that process.

Communication Between Processes (Shared Paramacro Parameters)

A link between the different processes of the machine is provided with the dual-process shared paramacro parameters. These parameters are available only on dual-process systems. They provide a link through which common data can be passed between the different part programs executing in the different processes of the machine. The shared dual-process parameters are paramacro parameter numbers:

#7100 to #7199

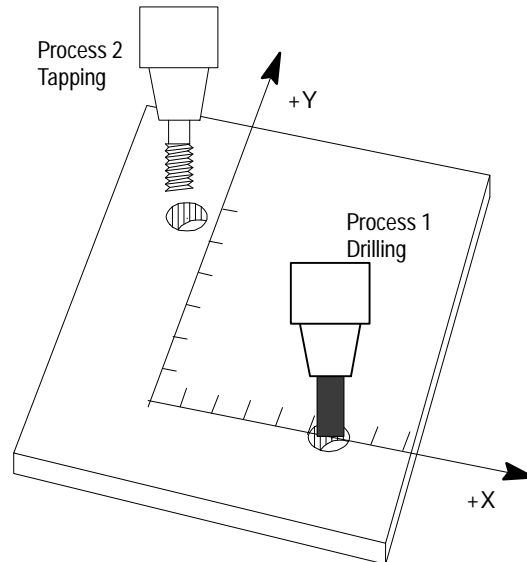
Refer to page 28-36 for details on these paramacro parameters.

30.3 Synchronizing Multiple Part Programs

On some machines or systems, it is often necessary to synchronize the operations of 9/Series dual processes. For example, if one process is drilling holes while the second process is tapping the holes, it is extremely important that the drilling process completes before the second process attempts tapping of the hole.

Figure 30.1
Example of When Process Synchronization is Necessary

First process must complete drilling before second process can begin tapping.



Part program synchronization is accomplished between processes through the use of special M-codes. These M-codes are in group 13, and they are non-modal. The M-codes reserved for synchronization are M100 - M199.

When you execute a synchronization M-code in one process, that process halts until a block containing the same synchronization M-code is executed in the second process. In the case of the tapping example in Figure 30.1, placing one of these M-codes before the tapping block and after the drilling block causes the tapping block to pause (not execute) until the drilling block has completed.

These synchronization M-codes cannot be programmed in a block with other commands. The only other types of information you can program in a synchronization block are:

- N-codes (block number)
- O-words (program name)
- Program comments (text enclosed by parentheses ())

Synchronization M-codes are not allowed in the last block of the part program. This can cause the part program to pause indefinitely, waiting for the next part program block (which does not exist) to become ready.

Synchronization M-codes are ignored during QuickCheck execution and during a Mid-Program Start operation.

Example 30.1
Example of Synchronization for Tapping (see Figure 30.1)

Process 1	Comment	Process 2	Comment
N10 G90 S500 G00 X0 Y0;	Start spindle and move to start point		
		N1 G90 G00 X-2. Y0 S30;	Move to start point and start tapping spindle
N20 G81 X-1 Y5.5 R3 Z1;	Drilling hole at X-1 Y5.5		
		N2 M100;	Wait for first hole to finish drilling and clear the position for tapping (process 1 block N40)
N30 X5.7 Y0;	Drilling hole at X5.7 Y0		
N40 M100;	Release process 2 block N3		
		N3 G84 X-1 Y5.5 R3 Z1F20;	Start tapping first hole
N50 G00 X0 Y0;	Move away from second hole for tapping		
		N4 M101;	Wait for second hole to finish drilling and clear the position for tapping (process 1 block N60)
N60 M101;	Release process 2 block N5		
		N5 X5.7 Y0;	Start tapping second hole
N70 M30;			
		N6 M30;	

Simple Synchronization vs Synchronization with Setup

There are two types of synchronization M-codes. They are:

- **M100 - M149 – Simple Synchronization.** These M-codes are for simple synchronization. When executed, this set of M-codes does not set up any program blocks that have already been read into program lookahead again.
- **M150 - M199 – Synchronization with Re-Setup.** This set of M-codes cancels any information already in block lookahead and sets up the blocks again before process execution is resumed. This second setup is only essential when shared information is being changed from one process to another, as in the case of the dual-processing shared paramacro parameters.

Example 30.2
Incorrect Use of Simple Synchronization with Shared Paramacro
Parameters

Process 1	Comment	Process 2	Comment
N17 #7100=100;	Paramacro parameter 7100 is set to 100		
		N32 M100;	Process pauses waiting for M100 in process 1. Block N33 is set up in buffer prepared for execution.
N18 Z#7100;			
N19 #7100=25	Paramacro parameter 7100 is set to 25		
N20 M100;	Process 2 is released from pause		
		N33 X#7100;	Destination of this block is dependent on when this block was read into the setup buffer. If this block was set up before process 1 executed block N19, X moves to 100. If this block was set up after block N19 is executed, X moves to 25.

Since there is no way to consistently tell when block N33 in Example 30.2 is set up, an inconsistent destination for the block can result. When block N33 is set up depends on many factors, like what features you are currently using, the speed of execution, and complexity of blocks in both processes (see block lookahead in chapter 21 for details). To make certain that block N33 is not set up until process 1 releases process 2, use one of the synchronization-with-setup M-codes.

Example 30.3
Correct Use of Synchronization with Set up for Shared Paramacro
Parameters

Process 1	Comment	Process 2	Comment
N17 #7100=100;	Paramacro parameter 7100 is set to 100		
		N32 M151;	Process pauses, waiting for M151 in process 1. Block N33 is setup in buffer prepared for execution.
N18 Z#7100;			
N19 #7100=25	Paramacro parameter 7100 is set to 25		
N20 M151;	Process 2 is released from pause and any setup buffer read into process 2 block lookahead is purged and set up again.		
		N33 X#7100;	X axis moves to 25.

You can use synchronization-with-setup any time synchronization is necessary, but it does have some cost to system efficiency and can slow your system down. We recommend that you use M150 - M199 only when you change shared paramacro parameters. For most applications, simple synchronization (M100-M149) is sufficient.

Important: You cannot use these synchronization with setup M-codes when cutter compensation is active. Use one of the simple synchronization M-codes or turn off cutter compensation before programming the synchronization with setup M-code.

Coordinating Synchronization Between Processes

Remember that both processes are executing coordinated part programs. Failing to coordinate part programs correctly can result in the processes executing different synchronization codes and mutually locking each other out.

Example 30.4
Mismatched Synchronization Codes

Process 1	Comment	Process 2	Comment
N7 #7100=25;			
		N32 M101	Process 2 paused, waiting for M101 in process 1.
N8 M100	Process 1 paused, waiting for M100 in process 2.		

In Example 30.4, both processes are paused indefinitely, waiting for the other process to execute the corresponding M-code. The control displays an error message when this situation occurs, indicating that execution of that block has been suspended until it synchronizes with the other process. Perform a Block Reset to allow part program execution to continue.

You can help prevent this situation by:

- being consistent and always start your programs with the same synchronization M-codes
- making certain each program contains the same number of synchronization M-codes
- making sure the synchronization M-codes are placed in the same corresponding order between processes

Synchronization in MDI Mode

Synchronization M-codes can be programmed in MDI mode. These can prove useful when attempting to manually start multiple programs from some point other than the beginning or when it is necessary to execute MDI programs on both processes simultaneously. By inserting a synchronization M-code (by using MDI and pressing cycle start) you can start a process executing, but place it in pause until you start the second process and it executes the corresponding synchronization code.

Example 30.5
Synchronization in MDI Mode

Process 1 (MDI mode)	Comment	Process 2 (MDI mode)	Comment
N1 M100;	You can press <CYCLE START> and this block pauses while you MDI the necessary information into process 2.		
		N1 M100;	Pressing <CYCLE START> at this block has the effect of starting both processes' MDI program at the same time.
G28 X10 Z10;			
		G28 X10 Z10;	

Program Interrupts During Synchronization Blocks

Program interrupts can be performed while a process is being paused by a synchronization M-code (see chapter 29 for details on interrupts). If the process is paused when the interrupt is requested, the request is recognized; however, the interrupt does not execute until the synchronization M-code is released by the other process, executing the appropriate synchronization code.

<CYCLE STOP> During Synchronization Blocks

Press <CYCLE STOP> places in cycle suspend mode while it is paused by a synchronization M-code. This lets you perform most manual operations (such as jog retract) and other functions normally available to a process in cycle suspend mode. No loss of synchronization occurs while the process is in cycle suspend.

Block retrace does not retrace through part program blocks with synchronization M-codes in them. The error message "NO FURTHER RETRACE ALLOWED" appears.

For example, press <CYCLE STOP> to place process 1 in cycle suspend mode, while process 1 is waiting for process 2 to execute an M101. Later, when you request <CYCLE START> for process 1, the synchronization M-code is re-activated and process 1 is again paused, waiting for process 2 to execute an M101. If, while process 1 is in cycle suspend mode, process 2 executed an M101, process 2 pauses at that synchronization block. Pressing <CYCLE START> for process 1 releases the pause for both process, and they resume program execution.

30.4 Spindle Control for Dual- processing Systems

The typical dual mill configuration contains a separate spindle for each process. In some cases the spindles are on completely separate machines; therefore at no time are spindle commands shared by both processes.

The separate spindle configuration operates identically to a single-process system. Each process has full and total control of its assigned spindles. Both processes can have multiple spindles in their control. These spindles are assigned to only one process and cannot be controlled by more than one process. When a process has more than one spindle under its control, use a G12.x command to select the currently active spindle. See chapter 17 for details.

Remember that both processes are executing coordinated part programs. Failure to do so can result in the processes executing different synchronization codes and mutually locking each other out between processes. This configuration is not covered by this documentation. Refer to your system installer's documentation if you have a spindle that is shared by both processes.

30.5 Using Interference Checking with a Dual-process Mill

The Interference Checking feature is designed to help prevent collisions by the axes of a dual-processing machine.

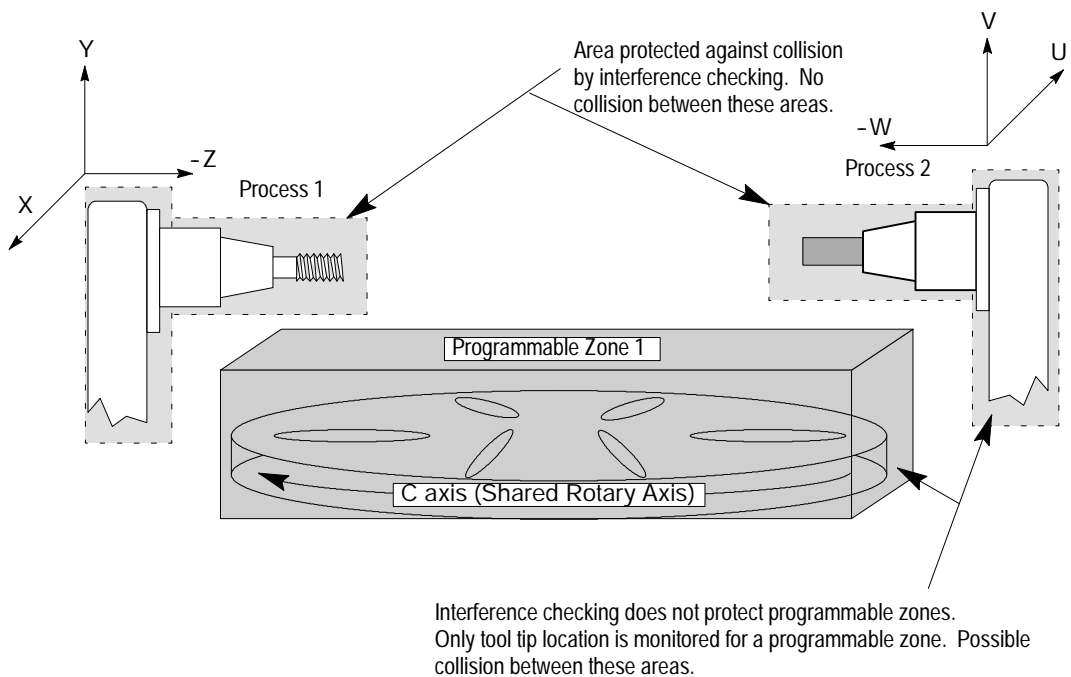
Interference checking provides an area (usually around the cutting tool or tool or axis fixtures for each process) that defines a boundary that moves when the tool moves relative to the axes. The other process cannot enter into this boundary. This helps prevent collisions.

For interference checking to function properly, you must define and activate this boundary for both processes. When one process's boundary attempts to enter another process's boundary, one process enters cycle suspend, preventing it from entering the protected area. In the case of an unavoidable direct collision, both processes stop. An error message is displayed for both processes. The two boundaries cannot intersect.



CAUTION: These interference boundaries only help prevent collision with another interference boundary configured for another process. They do not protect against collisions with other machine fixtures that may or may not be protected by a programmable zone or software overtravel. They do not account for any tool offsets that are active at the time. They are absolute positions.

Figure 30.2
Interference Check Helps Prevent Tool Fixtures from Colliding



Important: Shared axes can only be defined in an interference zone in one process. If the system installer defines the same shared axis to be in an interference zone for both processes the control generates an error.

Activating Interference Checking

The interference boundaries for each process are entered into the interference checking tables. These tables relate the boundaries to specific tool or offset geometries. The system installer selects the number of boundaries that are available (from 1-32) for each process.

Each process can have a different interference boundary number active at the same time. Activating an interference boundary for only one process is not sufficient to activate interference checking. Each process must have an active, valid interference boundary.

The system installer determines how the different interference boundaries are activated for each process through PAL. Refer to your system installer's documentation for details on activating these boundaries.

Before interference checking can be activated, all axes that make up a boundary must be homed.

Important: Your system installer has the option to determine with PAL when interference checking becomes active. Interference checking is available in one of the following mode combinations:

- manual mode
- automatic and MDI modes
- manual, automatic, and MDI modes

Refer to your system installer's documentation for details.

Once interference checking is turned on, it remains active in that mode (Auto, MDI, or Manual) until turned off through the system installer's PAL. This includes when no program is active, a control reset is performed, or even when the control is in E-STOP (unless PAL has been specifically written to turn off interference under these conditions). Interference checking can also be active in mechanical handle feed mode (servo off).

Interference checking is not performed on the part program during QuickCheck mode. Interference checking is still active in QuickCheck; however, since axis motion is not usually present during QuickCheck, programmed collisions are not detected.

Using Interference Checking to Prevent Collisions

When two protected areas are about to collide, the control suspends motion, stopping one or both of the processes and preventing a collision.

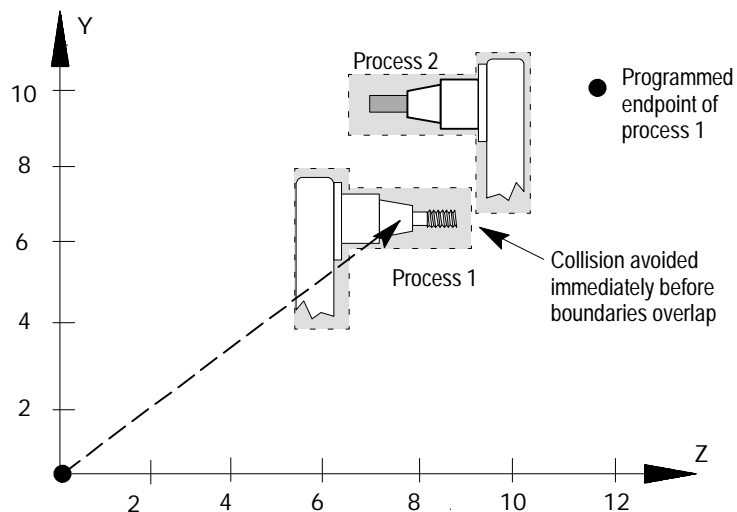
In Example 30.6, process 1 collides with process 2. Since process 2 is stationary, the control puts process 1 in cycle suspend to prevent a collision. Once the control detects the collision, it suspends the action.

Interference checking operates in real time. It is not checked in block lookahead. The control detects a collision immediately before the boundaries overlap and suspends one or both of the processes, halting the axes with an abrupt, un-decelerated stop. In Example 30.6, even though the endpoint of the tool for process 1 is not within the interference boundary for process 2, the block is halted since the boundaries intersect in route.

Example 30.6
Collision Programmed with Interference Checking On

Process 1	Process 2
N1 G90 G20;	N1 G90 G20;
N2 G00 W0 U0;	N2 G00 Z7 X9.5;
N3 G00 W12 U-10;	N3 M30;
N4 M30;	

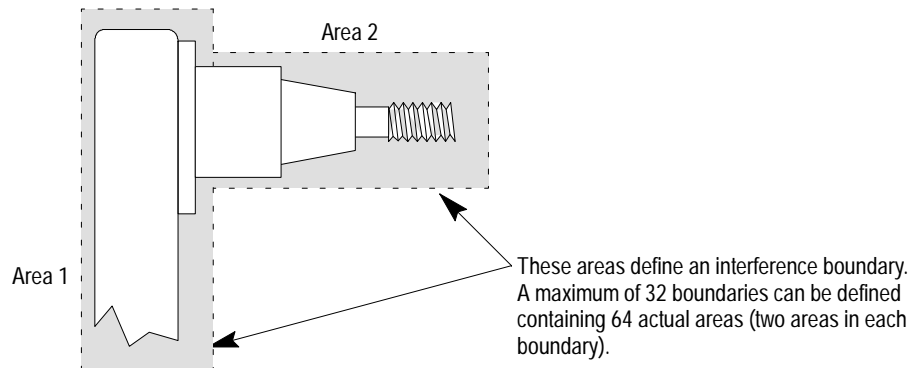
Figure 30.3
Results of Example 30.6



30.5.1 Measuring Interference Boundaries

The control can store as many as 32 different boundaries for each process. Two separate areas make up each of these boundaries. Both axes are activated when the boundary is activated through PAL. Figure 30.4 illustrates the use of two areas to make up interference boundary 01.

Figure 30.4
Using Two Areas to Define an Interference Checking Boundary



At least one area must be defined for a boundary to be considered valid for activation. If only one area is defined, the control assumes the second area does not exist.

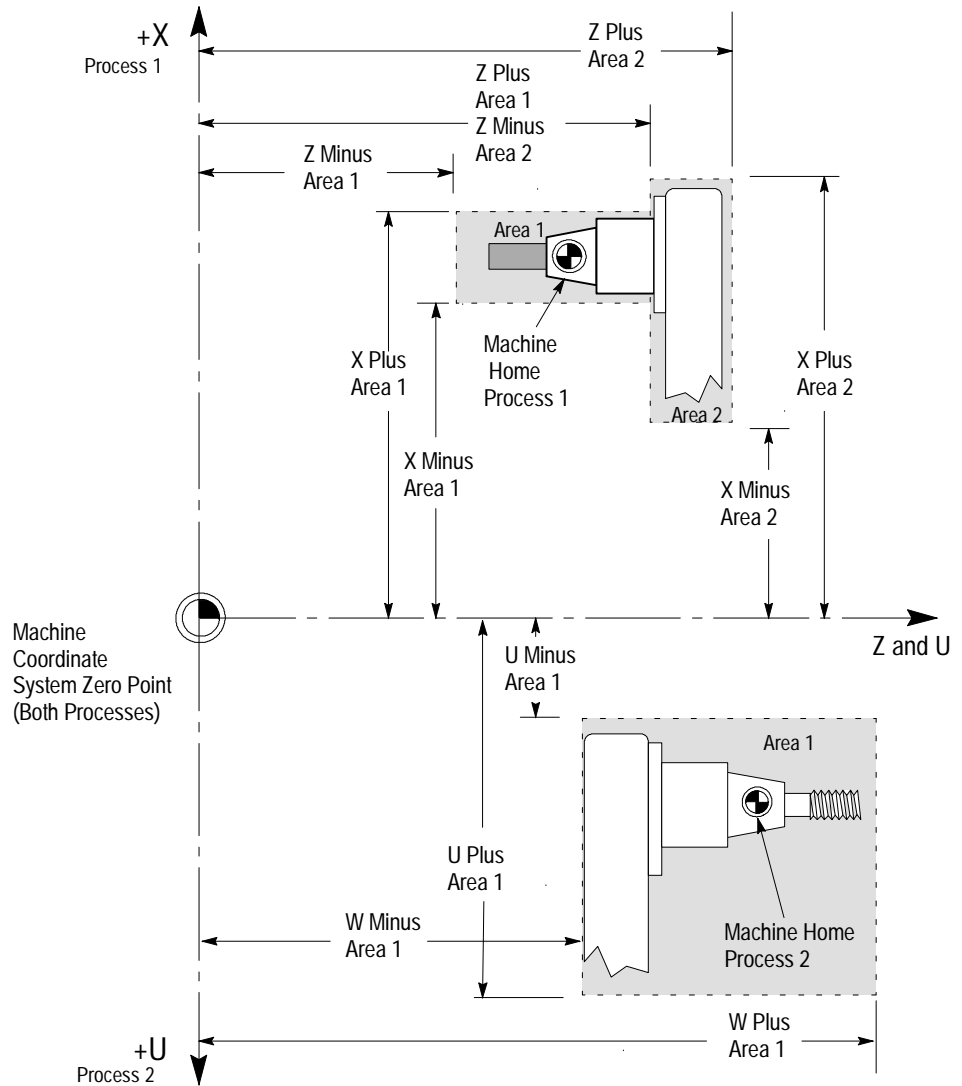
Areas for interference checking are defined in the machine coordinate system for the fixtures positions as the axis sits at the home position. Only two axes can be entered into the interference tables to define a boundary. Your system installer selects these two axes in AMP. Follow this procedure to measure values defining your interference areas:

1. Return the axes to the home position (with no offsets active). This can be accomplished by manually homing the machine (see chapter 5) or by programming a G28 (see chapter 14).
2. Measure the values defining the interference area for each axis as shown in Figure 30.5.

You will probably need to take different measurements for all of the different fixtures and tools you will be using. Regardless of the tool or fixture being used, the machine must remain at machine home (with no offsets active) when the measurements are taken.

Important: Your system installer determines the relationship of the machine coordinate systems between processes (relative location of zero points and direction of positive travel) in AMP and through hardware. This manual assumes the machine coordinate systems of both processes are as shown in Figure 30.5. Refer to your system installer's documentation for details on how your machine coordinate systems are configured.

Figure 30.5
Measuring Interference Checking Areas



Important: These areas are measured from the machine coordinate zero point to the extremes of the fixture encompassed by the zone when the machine is at home. The machine coordinate system zero point and machine home are frequently not the same point on the machine. Machine home is a fixed mechanical position established by the homing sequence off hardware homing switches. The machine coordinate system zero point is established in AMP and referenced from the machine home point (see chapter 11 for details). Refer to your system installer's documentation for information regarding the location of machine coordinate system zero. For interference checking to function properly, both processes must have the same zero point, but they can have different home locations.



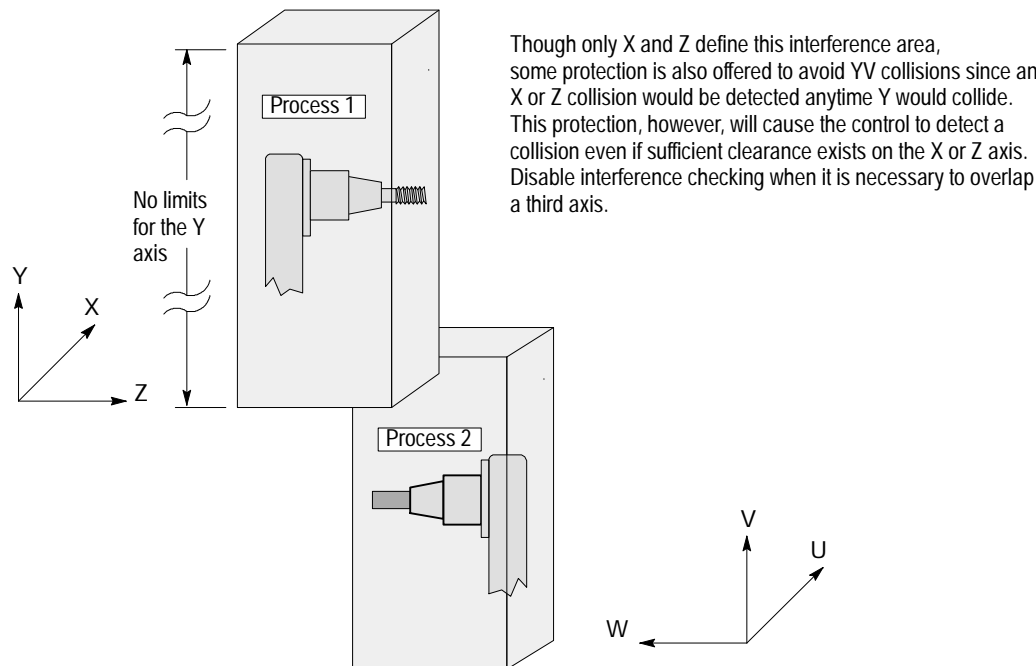
CAUTION: The distance between the boundaries before a collision is detected is dependant upon factors such as:

- speed of the axes
- direction of axis travel with relationship to one another

For example, a programmed collision between two axes traveling at rapid directly towards one another may not be detected in time to fully stop the axes before some overlap of the interference boundary occurs. Make your boundary areas as large as possible (definitely larger than the actual physical dimensions of the protected fixture) to allow for this potential overlap condition.

Only two axes can be entered in an interference table (available axes are configured in AMP). To prevent a collision, other axes must rely on protection by this interference area, or depend on you to use extra care when programming or manually positioning the axes to prevent a collision.

Figure 30.6
Protecting Additional Axes with Interference Checking



30.5.2 Entering Interference Values Manually

To manually enter values into the interference checking tables, follow this procedure:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	FRONT PANEL	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
--	------------------	----------------	----------------	-----------------	------------------	---

	OFFSET	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→
--	--------	-----------------	---------------	-----------------	--	---

2. Press the {PROGRAM PARAM} softkey.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVI CE SETUP	MONI - TOR	TI ME PARTS	→
---	-----------------	-----	------------------	---------------	----------------	---

↑	PTOM SI /OEM					→
---	-----------------	--	--	--	--	---

3. Press the {INTERF CHECK} softkey to display the interference checking data entry screen shown in Figure 30.7.

(softkey level 3)

↑	ZONE LI MI TS	F1 - F9	DRLCYC PARAM		INTERF CHECK	
---	------------------	---------	-----------------	--	-----------------	--

Figure 30.7
Interference Checking Data Table

INTERFERENCE TABLE						PAGE 1 OF 32
TOOL NO *1	AREA 1 [INCH]	AREA 2 [INCH]				
X PLUS	1.5000	1.5000				
X MINUS	-.5000	-1.0000				
Z PLUS	1.5000	6.0000				
Z MINUS	0.0000	1.5000				

<PROCESS 1>					
↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	MORE ZONES	BACKUP INTERF

4. Select a process. Refer to the system installer's documentation for details on selecting a process, or press the [PROC SELECT] key.
5. Select a boundary number to enter by using one of these two methods. Press:
 - the {MORE ZONES} softkey, or press the [•] key while holding down the [SHIFT] key.
 - the {SEARCH NUMBER} softkey. Enter the desired boundary number to search for and press the [TRANSMIT] key.

This boundary number is typically the same as the tool geometry number (H-word) that is active when the tool and/or fixture is being controlled. Refer to your system installer's documentation for details on which tool or fixture corresponds to which interference boundary number (1-32).

6. Use the up or down cursor keys to move the block cursor to the interference area parameter to be changed. The selected field appears in reverse video.
7. If necessary, change the measurement units by using MDI. Units currently used in the table are determined by the current mode of the process (G20 inch or G21 metric).

8. Enter the boundary area values as determined on page 30-16. Enter values in one of two ways:

Press This Softkey:	Then:	Press:	The New Value:
{REPLCE VALUE}	Type in the new value.	[TRANSMIT]	replaces the old value for that feedrate.
{ADD TO VALUE}	Type in the new value.	[TRANSMIT]	is added to the old value for that area.

9. Repeat this procedure for each process until all boundaries are entered.

30.5.3 Entering Interference Values through Programming (G10L5 and G10L6)

You can enter data in the interference tables by programming the correct G10 command. This section describes the use of the G10 commands.

Important: The active boundary and the value in the interference boundary table change when a G10 code modifies the table value. Both changes activate immediately.

When the process is in incremental mode (G91), any values entered in the table with the G10 command are added to the currently existing offset values. When the process is in absolute mode (G90), any values entered in an offset table with the G10 command replace the currently existing offset values.

Values can be entered into the table as inch or metric values. Select the values you intend to enter by first programming the G-code that establishes the mode you wish to use (G20 or G21).

This is a representation of the basic format for modifying the tables.

$$G10 L\left\{\begin{matrix} 5 \\ 6 \end{matrix}\right\} P_ X_ Z_ I_ K_;$$

Where :	Is :
L(5-6)	The definition of which area in the table is being modified. L5 - Modifies the Area 1 values L6 - Modifies the Area 2 values
P	The boundary number of the interference boundary that is having its values changed is specified following the P address.
X	The value to add to (in G91 mode) or replace (in G90 mode) the positive X axis value. This value is always a radius value.
Z	The value to add to (in G91 mode) or replace (in G90 mode) the positive Z axis value.
I	X axis integrand. The value to add to (in G91 mode) or replace (in G90 mode) the negative X axis value. This value is always a radius value. I is the integrand word of the X axis.
K	Z axis integrand. The value to add to (in G91 mode) or replace (in G90 mode) the negative Z axis value. K is the integrand word of the Z axis.

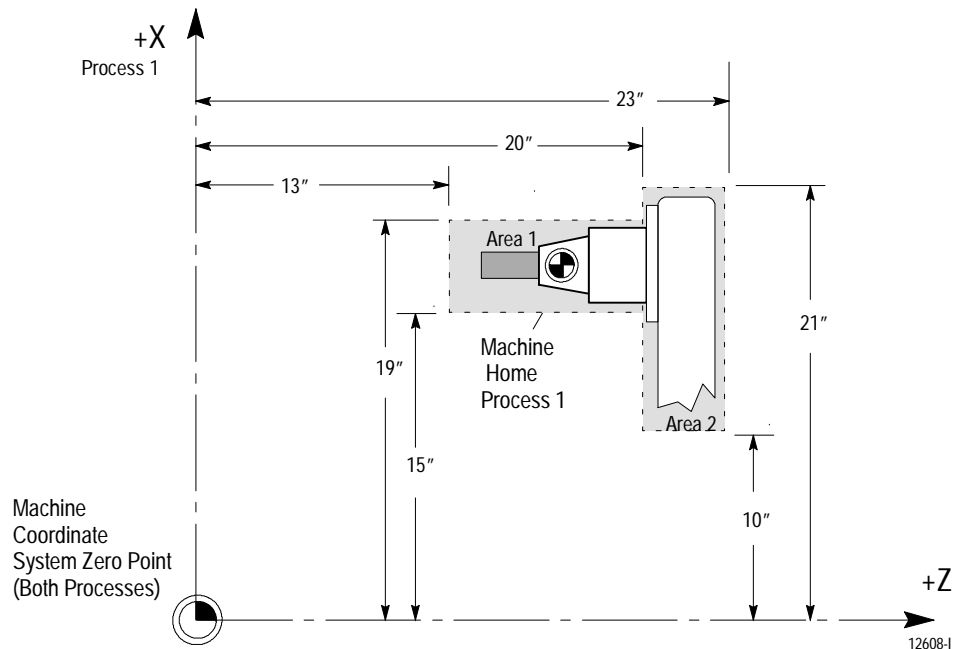
Programming this G10 code can change only the table values for the process that has the G10 part program currently active. You must run separate G10 programs in each process to set up each area.

Important: G10 blocks cannot be programmed when TTRC is active.

Example 30.7
Using G10 to Change the Interference Boundaries

N1 G90 G20;
 N2 G10 L5 P1 Z20 K13 X19 I15; Boundary number 1 area 1 is defined.
 N3 G10 L6 P1 Z23 K20 X21 I10; Boundary number 1 area 2 is defined.

Figure 30.8
Resulting Boundary from Example 30.7



30.5.4 Backing Up Interference Tables

The control can save all of the information that is entered in the interference tables as a backup. This is done by the control generating a program consisting of G10 blocks. These G10 blocks contain the boundary numbers and their respective interference boundary values. Any time you run this program, the set of values contained in these G10 blocks replaces the current values in the interference tables. See page 30-21 for details on the format of this G10 program.

The interference table program can be saved in control memory as a program, or it can be sent to a peripheral device. The interference tables can be easily setup again by executing this program in the appropriate process.

To back up the interference tables, follow these directions:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	FRONT PANEL	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
	OFFSET	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

2. Press the {PRGRAM PARAM} softkey.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVI CE SETUP	MONI - TOR	TI ME PARTS	→
↑	PTOM SI /OEM					→

3. Press the {INTERF CHECK} softkey to display the interference checking data entry screen as shown in Figure 30.7.

(softkey level 3)

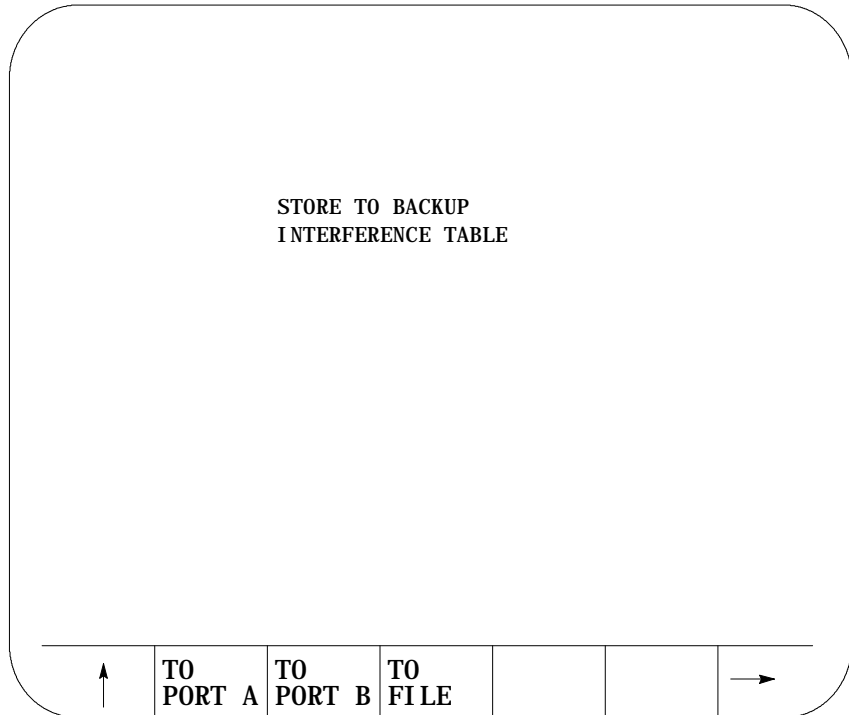
↑	ZONE LI MITS	F1- F9	MI L CYC PARAM	PROBE PARAM	INTERF CHECK	
---	-----------------	--------	-------------------	----------------	-----------------	--

4. Press the {BACKUP INTERF} softkey. Figure 30.9 shows the backup interference boundary screen.

(softkey level 4)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	MORE ZONES	BACKUP INTERF	
---	------------------	-----------------	-----------------	---------------	------------------	--

Figure 30.9
Backup Interference Boundary Screen



5. Determine the destination for the G10 program:

To Send the G10 Program To:	Press This Softkey:	Go to Step:
peripheral attached to port A	{TO PORT A}	7.
a peripheral attached to port B	{TO PORT B}	7.
to control memory	{TO FILE}	6.

6. Press the {TO FILE} softkey. The control asks for a program name. By using the alphanumeric keys on the operator panel, enter the program name you wish to call the backup program. Press the [TRANSMIT] key once you have entered the program name. See chapter 10 on program names. The G10 program is saved under the file name just entered.
7. Press the {TO PORT A} or {TO PORT B} softkey. A {YES} or {NO} softkey is displayed. Press {YES} to confirm or {NO} to abort. If you confirm the operation, the G10 program is written to a peripheral attached to the selected port.

30.6 Shared Axes on Dual-processing Systems

Your system installer can configure an axis to be shared by different processes. With this feature multiple processes can execute part program commands or perform manual operations on the same shared axis.

A shared axis can not be commanded by more than one process simultaneously. Control of the shared axis must be changed from process to process thru the system installer's PAL program.

30.6.1 Operating a Shared Axis

Shared axes are shown on the position display in the order your system installer defines them in AMP. Shared axes are displayed only for the process that currently has control of the shared axis (except on graphics and axis monitor screens). When control of the shared axis is changed to a different process, the axis is removed from the position display and added to the new controlling process.

You can select to plot a shared axis for display on one of the graphics screens regardless of the process currently controlling the shared axis. The machine info window on graphics screen always shows only the axes currently assigned to that process.

Homing a Shared Axis

A shared axis is homed like any other axis. It must be homed before it can be positioned by a part program block. Home the shared axis in the process that it is currently assigned. There is only one home position for a shared axis (this position is independent of the current process). Once a shared axis is homed in one process it becomes homed in any additional processes as well. You do not need to home the shared axis in each process.

Programmable Zones

Each process contains two independent programmable zones. Zones are checked on a per process basis. The system installer can setup shared axis zone values in AMP. A shared axis has the same zone values in both processes. Shared axis zone values can be manually entered in either process through the zone configuration table (see chapter 3 for details).

Zone 3 values can be changed in the part program as discussed in chapter 12. Each process can have a separate programmed value for for a shared axis in zone 3 however these programmed values are only temporary and do not overwrite the original AMP or manually entered table values.

Shared axis values in a programmable zone are monitored only in the process currently controlling the shared axis. Shared axes are no longer part of a programmable zone once they leave the process. Zones defined to contain shared axes can be enabled and disabled in any process regardless of the process currently controlling the shared axis. If a zone is activated without a shared axis member, the shared axis zone values are ignored for that process until the shared axis is switched into that process.

Block Retrace

Any part program blocks prior to an axis process switch can not be retraced. If you attempt to retrace beyond the point that an axis switch occurred, the control generates an error. Also an axis process switch can not be performed if you are currently performing a block retrace.

Scaling

Scaling is performed on a per process basis. If you switch processes for a scaled axis, scaling is removed in the new process. Scaling is then re-applied to the axis when it returns to the process in which scaling is active for that axis. You can scale the shared axis by different amounts in different processes.

Axis Inhibit, Axis Detach, and Servo Off

When you turn any of these three features on for a shared axis, it applies to that axis in all processes. Control of that axis can be changed from one process to another. However, the state of Axis Inhibit, Axis Detach, and Servo Off remains constant from process to process.

Mirroring

There are two types of mirroring available:

- Programmable Mirror Image (G50.1-G51.1) - This type of mirroring is performed on a per process basis. The shared axis is mirrored only in the process requesting that axis be mirrored with a G51.1 block.
- Manual Mirror Image - This type of mirroring is performed on a per axis basis. Manual mirroring is performed thru PAL or thru the front panel softkey. Shared axes that are mirrored remain mirrored in all processes as long as manual mirroring remains active.

Part Rotation

Both programmed coordinate system rotations (G68-G69) and external part rotation are performed on a per process basis. Rotation is removed from a shared axis when it changes process. The shared axis applies any rotation that is active in the new process. Rotation remains active in a process that is not controlling a shared axis however this rotation is not applied until control of the shared axis is returned to the process.

Plane Selection

You can not activate a plane that contains a shared axis that is currently in another process. You can however switch an axis that is in the currently active plane over to a different process and still keep the plane active.

When the active plane contains a shared axis currently controlled by another process, programming any plane dependent features (such as circular interpolation) results in an error. The plane becomes available when control of the shared axis is returned to the process.

30.6.2 Switching a Shared Axis to a Different Process

The system installer determines what axes are shared and how a shared axis is changed from process to process. Using AMP and PAL the system installer determines the process for a shared axis at power up, control reset, and E-STOP reset. Refer to your system installer's documentation for details.

Even though the system installer's PAL program determines how and when a shared axis changes processes, there are some global restrictions that can not be altered. Before a request to change a shared axes process is performed, both processes that share the axis must be in one of the following states:

- E-STOP
- Cycle Stop (not cycle suspend)
- Manual mode with the shared axis not being positioned
- Executing an M99 in a main program
- Postlude State (established through PAL)

Other restrictions are as follows:

If the shared axis is currently:	Then:
performing a <u>manual motion</u> (including continuous, incremental, or handwheel jog, homing, jog on the fly, or angled jogs)	the request to change process for that axis is ignored until the manual motion is completed ^❶
being positioned by the <u>PAL axis mover</u>	the request to change processes is ignored until the PAL axis mover has completed moving the shared axis ^❶
in the active plane and <u>cutter compensation</u> is active	the request to change processes is ignored until the plane is changed or cutter compensation is de-activated ^❶
involved in a modal <u>fixed cycle</u> (such as drilling, pockets, or 7300 cycles)	the request to change processes is ignored until the cycle is canceled ^❶
a member of a <u>deskew</u> axis (split axis)	the request to change processes can only occur on the master axis in the deskew pair. Requests to switch the deskew slave servo are ignored. The deskew slave servo automatically moves to the process of the deskew master servo.
If either process is performing a:	Then:
<u>jog retract</u> or <u>block retrace</u>	shared axes can not be switched into or out of the process. This is independent of whether the shared axis is involved in the jog retract or block retrace operation ^❶
<u>program interrupt</u>	a shared axis can not be switched into or out of the process ^❶
synchronization operation with a synchronization M-code	a shared axis can not be switched into or out of the process ^❶
<p>❶ The request for an axis to change processes is ignored until the prohibiting feature is completed or canceled. Then the change in process only takes place provided the request for the axis change is still true.</p>	

30.6.3 Setting up a Shared Axis

Your system installer performs the majority of set up operations in PAL and AMP to define a shared axis configuration. This section covers operations you should perform on the control to properly operate the shared axis.

Setup Tables

When assigning table values (such as the tool offset tables, coordinate offset tables, and on-line AMP tables) shared axis data only appears on the table when the current process has control of the shared axis. For example, if you need to enter tool length offset data for a shared X axis, you can enter this offset data from any process as long as the X axis is currently controlled by that process.

Important: G10 part program blocks can only change shared axis offset table values if the shared axis is currently assigned to the process running the G10.

Axis data is displayed only for axes currently assigned to the selected process. When control of a shared axis is changed the table display of the shared axis changes also. For example if process one has control of an X shared axis and you are viewing the X axis on a tool offset table, when control of the X axis is changed to process two, the X axis display disappears on the process one tables and appears on the process two tables. No change to the actual physical data occurs. All table data for a shared axis stays the same between processes (entering a three inch tool length offset for a shared axis in one process stays a three inch tool length offset for all processes unless changed).

Changes made to tables that require an update and exit operation to be saved (such as On-line AMP screens) must be saved before the shared axis changes processes. If for example, you change zone information for a shared axis in process one and change control of that axis to process two before pressing the {UPDATE & EXIT} softkey, changes to the shared axis are discarded and the last value saved for that axis is used.

Shared Axis Coordinate Offsets

Each process maintains separate and independent shared axis:

- work coordinate system G54-G59.3 offsets
- G52 offsets
- external offsets

If any G92 offset is active on a shared axis it must be canceled (or made zero) before the shared axis can change processes.

These offsets are applied to the shared axis when it is switched into the process. All part program blocks already read into the setup buffers are re-setup in all processes when a shared axis changes processes.

You can not change the offset for an axis that is not currently assigned to the process through a part program (G52, and G92). You can however change coordinate system tables without the shared axis being in the process using PAL or by manually inputting the data through the {OFFSET} softkey. If the shared axis is not in the process activating the new work coordinate system (G54-G59.3) the data for the shared axis is not activated until that axis is moved under the control of that process.

Shared Axis Tool Offsets

Tool offset data for shared axes are shared from process to process. There are no separate tool offset table entries between processes. If you alter the tool offset value for a shared axis tool number in process one, the same geometry offset value for the shared axis in process two is used. This feature allows you to activate a tool offset for a shared axis and have that offset carried on to additional processes as that axis changes processes.

Important: You can only change the tool offset tables for axes in your current process. If your shared axis is not in the current process it is not available on the offset table screen for editing. G10 commands also only modify shared axis offset values when the shared axis is assigned to the process executing the G10 command.

If you require different tool offsets for a shared axis between processes you should use independent tool offset numbers for each process (for example only use tools 1-50 for process one and tools 51-100 for process two).

Tool offsets are re-initialized when an axis changes processes. A shared axis can change processes even if the tool offset for that axis has not yet been made fully active (delay shift or delay move).

If a tool is activated with an offset for a shared axis not currently in the process, the motion and position shift of that offset is deferred until the shared axis is moved into the process. This is independent of the AMP selected tool offset activation type (delay or immediate shift/move).

Example 30.8
Changing Processes with Tool Offsets

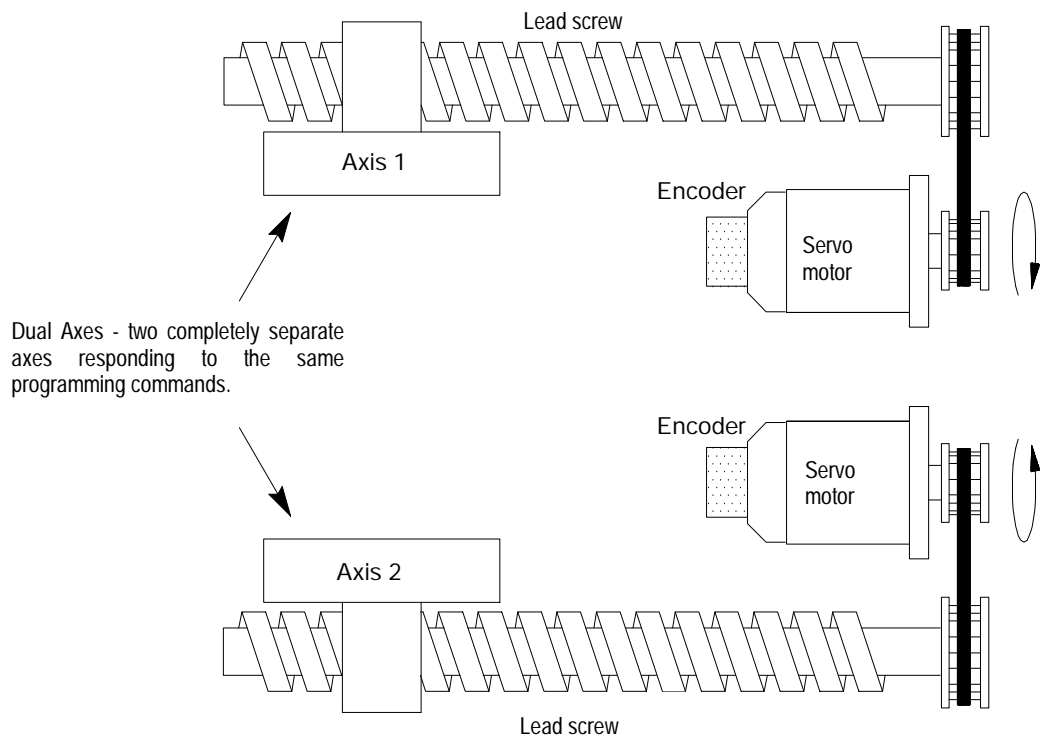
Process One Activates this Tool	Process Two Activates this Tool	Shared Axis
T10 H10; (controls shared axis)	T0 H0;	Process one activates tool offset on shared axis as defined in AMP (delayed/immediate shift/move). When process two takes control of the shared axis, the shared axis tool offset is canceled in process two.
T11 H11; (controls shared axis)	T11 H11;	Has same offset for both processes. When you shift the shared axis between processes, the offset remains active on the axis. When the shared axis changes processes the tool offsets are re-initialized and the AMP selected activation method (delayed/immediate shift/move) still occurs.
T11 H11;	T22 H22; (controls shared axis)	Process two activates tool offset on shared axis as defined in AMP (delayed/immediate shift/move). When you switch the shared axis to processes one, the offset changes to the tool offset value requested in process one. Since the offsets are re-initialized when shared axes change processes, the AMP selected activation method (delayed/immediate shift/move) still occurs.

30.7
Dual Axes on a
Dual-processing System

The Dual Axis feature allows the part programmer to simultaneously control multiple axes while programming commands for only one. It differs from the split axis feature of the control in that the split axis feature is used to control a single axis positioned by two servo motors.

The dual axis feature is especially useful for gantry mills with dual cutting heads and other machines running with parallel cutting tools. Figure 19.1 shows a typical configuration for dual axes.

Figure 30.10
Dual Axis Configuration



The 9/Series control supports two groups of dual axes. This is the total number of groups allowed on the system for both processes (i.e. two groups in one process or one group in each process).

Dual axis operation on dual processing systems is virtually identical to dual axis operation on single processing systems (see chapter 20 for details). The difference occurs on dual processing system when one or more of the axes in the dual group is configured as a shared axis between processes.

30.7.1 Decoupling a Dual Axis Group

A dual axis group is assigned in AMP to a specific process. All axes in the dual group must be configured to be part of the dual axis group and must be AMPed to be in the same process (called the default process for the group). Dual axes can only exist as a dual group in their respective default process. You can not share a dual group as an axis with another process. You can only share the independent members of the dual group with another process once the group is decoupled.

Coupling/Decoupling allows all axes in a dual axis group to be either:

- coupled as a dual axis group
- decoupled so each axis in the dual group can be programmed as an independent axis.

Coupling/Decoupling is a dual group function. All axes must be in the dual groups process before they can be either coupled or decoupled. When a coupling or decoupling occurs a re-setup occurs of any part program blocks read into the controls block look ahead buffer. This may causes a slight hesitation in program execution while the control sets up the new look ahead buffer.

The system installer uses PAL to determine how a dual axis group is decoupled. Refer to your system installer's documentation for details. Coupling/Decoupling of the axes can only occur when all axes in the dual group are not moving (including jogs or part programming, etc...) and are in the AMP default process for that dual group. If a coupling/decoupling is attempted during part program execution it occurs between blocks when interpolation is completed.

Once decoupled the axis park feature that is normally available to the individual axes of a dual axis group is no longer available. Axes can be parked only when the dual axis group is coupled.

To change the process of a shared axis member of a dual group, you must first decouple the dual axis group because couple/decouple is a group function. When a dual group is decoupled, the axes in the group behave like completely independent axes. Each dual group member get its own AMP assigned axis name.

Before a request to decouple a dual axis is performed, the process controlling the dual group must be in one of the following states:

- E-STOP
- Cycle Stop (not cycle suspend)
- Manual mode with the shared axis not being positioned
- Executing an M99 in a main program
- Postlude State (established through PAL)

Other restrictions are as follows:

If the dual axis is currently:	Then:
performing a <u>manual motion</u> (including continuous, incremental, or handwheel jog, homing, jog on the fly, or angled jogs)	the request to decouple that axis is ignored until the manual motion is completed ^❶
being positioned by the <u>PAL axis mover</u>	the request to decouple that axis is ignored until the PAL axis mover has completed moving the dual axis ^❶
in the active plane and <u>cutter compensation</u> is active	the request to decouple that axis is ignored until the plane is changed or cutter compensation is de-activated ^❶
used to perform CSS calculations (the dual axis has been assigned as the CSS axis and the control is currently in CSS mode)	the request to decouple that axis is ignored until CSS mode is canceled ^❶
in a process that is performing a <u>jog retract</u> or <u>block retrace</u>	the dual axes can not be decoupled. This is independent of whether the dual axis is involved in the jog retract or block retrace operation ^❶
in a process that is performing a <u>program interrupt</u>	a dual axis can not be switched into or out of the process ^❶
involved in a modal fixed cycle (such as drilling, pockets, or 7300 cycles)	the request to change processes is ignored until the cycle is canceled ^❶

❶ The request for an axis to change processes is ignored until the prohibiting feature is completed or canceled. Then the change in process only takes place provided the request for the axis change is still true.

30.7.2 Independently Programming Dual Axis Members

When coupled all axes of a dual group are programmed by the name of the AMPed master axis. Axes in a dual axis group can be programmed as independent axes once they have been decoupled. Once decoupled each axis gets its own independent axis name and integrand letter (defined in AMP) as opposed to the dual axis group name and integrand used to program the axis when coupled. For example a dual group consisting of two axes is programmed with a X axis name. Once the dual group is decoupled both axes in the dual group are programmed with their own axis name X and W.

Important: Once an axis has been decoupled from the dual group it still retains many traits from the original dual group. Features such as acc/dec, maximum feedrates, jog feedrates and jog increments are all adopted from the original group.

Shared axes that have been decoupled from a dual group can change processes. Once a shared axis is controlled by another process it can not be recoupled with the dual group until it returns to its original default process.

Slave axes can be defined in active primary or parallel planes (G17, G18, or G19) however the plane containing a slave axis can not be activated unless the dual group is decoupled and the slave axis is in the process requesting the plane.

An axis that is decoupled from its dual group can have an integrand letter assigned to it in AMP by the system installer. This integrand is used with that axes originally assigned AMP axis name to perform functions such as circular interpolation. Plane dependant operations (such as circular interpolation or cutter compensation) are available to a dual axes while coupled (provided the dual axis is defined in the active plane). Once decoupled these plane dependant functions are available to the members of a dual group as independent axes (provide that the axis is used in the active plane). Planes can contain dual axes from within the same dual group however these planes are only available when the dual axis group is decoupled and in the process that is currently activating the plane.

Dual Drilling Axis

If the slave of a dual axis is AMPed as the fixed drilling axis, the entire dual group operates as a drilling axis when coupled. Once decoupled however the drilling axis traits are independent for each axis. The system installer determines which axes are drilling axes in AMP. For example your control has a dual axis made up of X1, X2, and X3. The system installer has AMPed X2 to be the fixed drilling axis. When coupled the X axis is programmed as a drilling axis. When decoupled only the X2 axis is programmed as the drilling axis (note X2 when decoupled uses its AMP defined axis name, not X2).

END OF CHAPTER

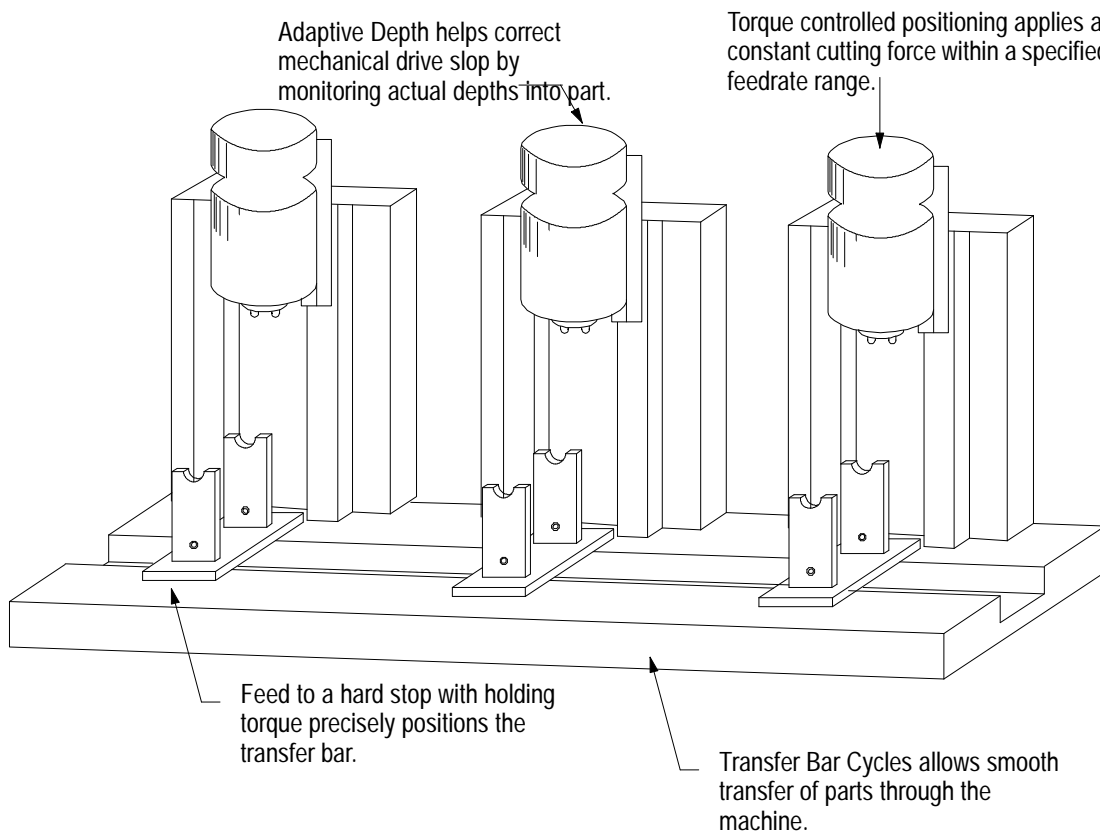
Using Transfer Line Cycles

31.0 Chapter Overview

This chapter details the user-defined cycles that are included with the transfer line option. The cycles also contain options for these features:

Topic:	For more information see page:
Feed to Hard Stop (Hard Stop Sense Zone)	14-40
Adaptive Depth	27-18
Adaptive Feedrate	18-9

These features, and how they are used on a transfer line, are briefly described here:



31.1 Transfer Line Cycles

The 9/Series T-Line-9 system comes with a set of part program templates for a wide variety of transfer line applications including:

Drilling/Boring/Tapping Application	Infeed	Hole Bottom	Retract	Template Name
Drilling Cycle, No Dwell/Rapid Out	Feed	Retract	Rapid Traverse	QV01
Drilling Cycle, Dwell/Rapid Out	Feed	Dwell /Retract	Rapid Traverse	QV02
Deep Hole Drill Cycle, Chip Clear	Intermittent Feed	Retract	Rapid Traverse	QV03
Deep Hole Drill Cycle, Chip Break	Intermittent Feed	Retract	Rapid Traverse	QV04
Right-Hand Tapping Cycle	Feed	Spindle Reversed / Retract	Feed	QV05
Right-Hand Solid-Tapping Cycle	Feed	Spindle Reversed / Retract	Feed	QV06
Left-Hand Tapping Cycle	Feed	Spindle Reversed / Retract	Feed	QV07
Left-Hand Solid-Tapping Cycle	Feed	Spindle Reversed / Retract	Feed	QV08
Boring Cycle, No Dwell/Feed Out	Feed	Retract	Feed	QV09
Boring Cycle, Spindle Shift	Feed	Oriented Spindle Stop / Retract	Rapid Traverse	QV10
Boring Cycle, Spindle Stop/Rapid Out	Feed	Spindle Stop /Retract	Rapid Traverse	QV11
Back Boring Cycle	Feed	Oriented Spindle Stop / retract	Rapid Traverse	QV12
Boring Cycle, Dwell/Feed Out	Feed	Dwell/Retract	Feed	QV13
Transfer Bar and Cross Slide Applications	Use			Template Name
Single Axis Lift Cycle	To lift and set the part off of and on to the transfer bar. Includes a soft touch, soft set.			QV14
Single Axis Transfer Cycle	To transfer the part between stations.			QV15
Two-Axis Transfer Bar Cycle	To lift, set and transfer the part between stations.			QV16
Single Axis Cross Cycle	To make a groove or a cut in a part using a cross slide.			QV17
Single Axis Feed Cycle	To feed a cutting tool in to or out of a part before or after a cut with a cross-slide is made.			QV18
Two-Axis Cross Feed Cycle	A combination of the single axis cross cycle and the single axis feed cycle.			QV19

31.1.1 Using Transfer Line Cycles

The cycles for the transfer line are user-defined. With the transfer line option, there are 19 templates that perform drilling, boring and transfer line functions. You can customize pre-written templates by using QuickView. In QuickView, there is a screen for each cycle. On each screen there is a graphic representation of each cycle and prompts with which you customize the program. Once you enter the information through QuickView the control generates a part program that functions like a cycle.

In the example on the next page, we show a part program that has been created by the control from information entered through QuickView. The lines of the part program that are affected by the information you enter in QuickView are in bold text. Other phases of the cycle are described by the text to the right of the part program.

```

N00001(QV09 BORING CYCLE G85)
N00002(DRILL SLIDE VARIABLES)
N00003IF[#1131EQ0]GOTO26 (INITIALIZES VARIABLES ONE TIME)
N00004#500=90 (G90/G91)
N00005#501=10 (HOLE POSITION 1ST AXIS)
N00006#502=0 (HOLE POSITION 2ND AXIS)
N00007#503=15 (DEPTH OF HOLE)
N00008#504=12 (CLEAR POSITION)
N00009#505=15 (FULL RETURN POSITION 1ST AXIS)
N00010#506=0 (FULL RETURN POSITION 2ND AXIS)
N00011#507=200 (FEEDRATE)
N00012#508=0 (TOOL CHANGE POSITION)
N00013#509=. 2 (HARD STOP SENSE)
N00014#510=0 (ADAPT. DEPTH INCREMENT)
N00015#511=3 (M03/M04)
N00016#512=500 (SPINDLE SPEED)
N00017#525=0. 001 (RETURN POS. TOLERANCE INCH)
N00018IF[#1005EQ0]GOTO20(TEST FOR INCH MODE)
N00019#526=0. 0254 (RETURN POS. TOLERANCE METRIC)
N00020#1102=0, #1103=0, #1104=0 (ALL OUTPUTS OFF)
N00021(CHECK TO SEE IF MAIN SLIDE IS RETURNED)
N00022IF[[#5041GT[#505+#525]]OR[#5041LT[#505-#525]]]GOTO24
N00023#1103=1 (RETURN OUTPUT)
N00024#1131=0, #1130=0(INTERNAL FLAGS)
N00025(-----)
N00026G01G90 (MODAL G CODES)
N00027(-----)
N00028 (MAIN PROGRAM LOOP)
N00029(---MAIN SLIDE ADVANCE MOTION---)
N00030WHILE [#1002] AND [#1102EQ0] DO1
N00031#1103=0, #1104=0 (RET. AND TOOL OUTPUTS OFF)
N00032IF[#5041GE#503]GOTO28 (POSITION CHK)
N00033IF[#520EQ1]GOTO39 (RETRACT PHASE CHECK)
N00034IF[#5041GE#504]GOTO37 (POSITION CHK)
N00035G00X#501Y#502 (RAPID HOLE POSITION)
N00036G00X#504S#512M3 (RAPID POSITION R PLANE)
N00037G01G24I#509X#503F#507S#512M3 (DEPTH POSITION)
N00038#520=1(SET RETRACT PHASE)
N00039G01X#504F#507 (RETURN POSITION)
N00040#520=0(CLEAR RETRACT PHASE)
N00041#1102=1 (DEPTH OUTPUT ON)
N00042END1
N00043(---MAIN SLIDE RETURN MOTION---)
N00044WHILE [#1003] AND [#1103EQ0] DO1
N00045#1102=0, #1104=0 (ADV. AND TOOL OUTPUTS OFF)
N00046G00X#505 (RETURN POSITION)
N00047G00Y#506 (RETURN POSITION)
N00048#1103=1 (RETURN OUTPUT ON)
N00049#520=0(CLEAR RETRACT PHASE)
N00050END1
N00051(---MAIN SLIDE TOOL CHANGE---)
N00052WHILE [#1004] AND [#1104EQ0] DO1
N00053#1102=0, #1103=0 (ADV. AND RET. OUTPUTS OFF)
N00054G00X#508M05 (TOOL POSITION)
N00055#1104=1 (TOOL OUTPUT ON)
N00056END1
N00057(-----)
N00058GOTO26(REPEAT MAIN LOOP)
N00059(-----)
N00060M02 (END MAIN PROGRAM)

```

The CNC sets up the cycle. This is the only time that the CNC initializes the variables for this cycle.

The CNC checks its position.

The CNC sets modal G-codes.

CNC advances the slide, and then it begins and completes the boring operation.

CNC moves the the slide to the return position.

CNC moves the tool to the tool change position and then ends the cycle.

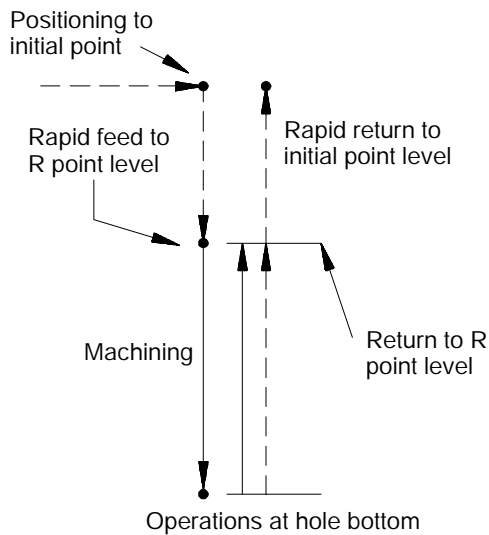
Using QuickView to Customize the Cycles

Though your transfer line control comes with part program templates, you need to customize that template into a working part program for your application. QuickView prompts that are designed to work with the part program templates let you to modify your program template.

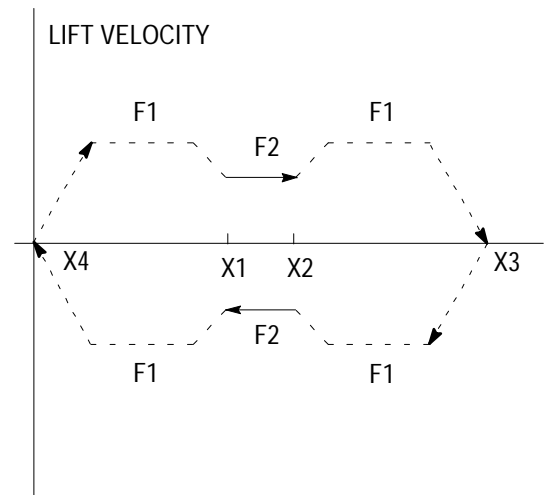
In QuickView, the illustrations vary slightly for drilling/boring cycles and transfer cycles. The graphics for the drilling/boring cycles are position profiles, and the graphics for the transfer cycles are velocity profiles. An example of each is given:

Figure 31.1
Milling Fixed Cycle Operations

————— Cutting feedrate
- - - - - Maximum cutting feedrate



Mill Cycle Illustration in QuickView



Transfer Cycle Illustration in QuickView

31.1.2
Selecting the
Program to Edit or Create

Before you begin editing a part program, the control needs to be in E-stop or the bit for Stop Program Cycle for Local Manual Control must be set.

1. Press the {PROGRAM MANAGE} softkey. The program directory screen is displayed.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWI TCH LANG		→

Figure 31.2
Part Program Directory

SELECTED PROGRAM						
DI RECTORY				PAGE 1 OF 1		
NAME		SIZE	COMMENT			
0 FILES 150 METERS FREE						
↑	ACTI VE PRGRAM	EDI T PRGRAM	RESTR T PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

- Type 1 for the selected program name and then press {EDIT PROGRAM}. The control names the created part program O00001.

(softkey level 2)

↑	ACTIVE PROGRAM	EDIT PROGRAM	RESTRT PROGRAM	DISPLY PROGRAM	COPY PROGRAM	→
↑	VERI FY PROGRAM	PRGRAM COMENT	DELETE PROGRAM	RENAME PROGRAM	INPUT DEVI CE	→
↑	REFORM MEMORY					→

Figure 31.3
Program Edit Screen

INSERT :

EDIT

FILE : 000001 POS 1*1 MODE : CHAR

↑	MODI FY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXI T EDI TOR	→
---	-------------------	-----------------	----------------	------------------	------------------	---

3. From the edit menu, press the {QUICK VIEW} softkey.
(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
---	------------------	-----------------	----------------	------------------	----------------	---

↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
---	------------------	------------------	------------------	---------------	---------------	---

↑	DI G I T Z E					→
---	-----------------	--	--	--	--	---

4. The softkey functions will change to those indicated below.
(softkey level 4)

↑	QPATH+ PROMPT	GCODE PROMPT	M I L L PROMPT	TRNSFR PROMPT	PLANE SELECT	
---	------------------	-----------------	-------------------	------------------	-----------------	--

31.1.3 Creating a Transfer Line Part Program

The control prompts you for information it needs to create part programs. To select the cycle you want to create a part program for, and enter the information for the cycle, follow these steps:

1. From the QuickView menu press the {TRNSFR PROMPT} softkey. The transfer line cycle selection screen appears.

(softkey level 4)

↑	QPATH+ PROMPT	GCODE PROMPT	MILL PROMPT	TRNSFR PROMPT	PLANE SELECT
---	------------------	-----------------	----------------	------------------	-----------------

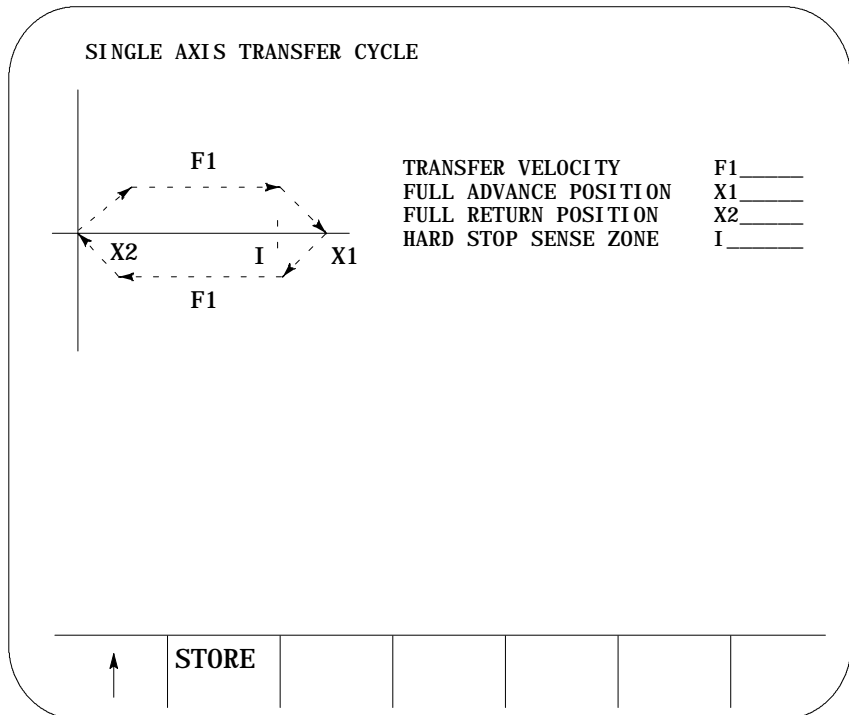
TRANSFER PROMPT MENU DISPLAY		PAGE 1 OF 2
01	DRILL CYCLE, NO DWELL/RAPID OUT	
02	DRILL CYCLE DWELL/RAPID OUT	
03	DEEP HOLE DRILL CYCLE, CHIP CLEAR	
04	DEEP HOLE DRILL CYCLE, CHIP BREAK	
05	RIGHT HAND TAPPING CYCLE	
06	RIGHT HAND SOLID TAPPING CYCLE	
07	LEFT HAND TAPPING CYCLE	
08	LEFT HAND SOLID TAPPING CYCLE	
09	BORING CYCLE, NO DWELL/FEED OUT	
10	BORING, SPINDLE STOP RAPID OUT	
11	BORING CYCLE, SPINDLE SHIFT	
12	BACK BORING CYCLE	
13	BORING CYCLE, DWELL/FEED OUT	

↑	SELECT					
---	--------	--	--	--	--	--

Press the <SHIFT> and <• > keys to select the next page of transfer line cycles. Press the <SHIFT> and <◀ > keys to select the previous page of transfer line cycles.

2. Position the cursor at the desired cycle name by using the up and down cursor keys. The selected cycle name is shown in reverse video.

- Once the correct cycle is selected, press the {SELECT} softkey. A screen with prompts for that cycle and a graphical representation of that cycle is displayed.



- Use the up and down cursor keys to select the parameters to be changed or entered. The selected item is shown in reverse video.

Axis words followed by a 1 or 2 are prompting for the first and second coordinate positions respectively.

Prompts that are shown in reverse video indicate that this parameter is optional and doesn't need to be entered in this block.

- Type in the desired data for that parameter on the input line of the CRT and press the [TRANSMIT] key.

Refer to the template descriptions at the end of this chapter for definitions of the prompts. These template descriptions assume that the default axis names of X and Y have not changed in AMP.

6. After all data for the cycle has been entered store the data by pressing the {STORE} softkey.

(softkey level 6)

↑	STORE					
---	-------	--	--	--	--	--

The control will generate the cycle's part program. See the section titled Editing Part Programs to adjust your settings.

7. Press the {EXIT EDITOR} softkey to leave the editor and make the program available for execution.

(softkey level 3)

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
---	------------------	-----------------	----------------	------------------	----------------	---

↑	STRING SEARCH	RENUM PROGRAM	MERGE PROGRAM	QUICK VIEW	CHAR/ WORD	→
---	------------------	------------------	------------------	---------------	---------------	---

↑	DI G I T Z E					→
---	-----------------	--	--	--	--	---

8. Take the control out of E-stop or manual control mode (through PAL). The control executes the program and monitors PAL input for information on how to proceed.

31.1.4 Editing Part Programs

Once you press the [STORE] softkey, the control generates a part program. Here is an example of a part program that the control generates:

```
INSERT :  
  
EDIT  
FILE : 000001 POS 1*1 MODE : CHAR  
N00001(QV15 SINGLE AXIS TRANSFER CYCLE)  
N00002(TRANSFER VARIABLES)  
N00003IF[#1131EQ0]GOTO29 (INITIALIZES VARIABLES ONE TIME)  
N00004#500=195 (TRANSFER VELOCITY)  
N00005#501=1.234 (FULL ADVANCE POS.)  
N00006#502=5.678 (FULL RETURN POS.)  
N00007#503=4.321 (HARD STOP SENSE)  
N00008(TURN OFF ALL OUTPUTS)  
N00009#1100=0, #1101=0  
N00010G01G90 (MODAL G CODES)
```

↑	MODIFY INSERT	BLOCK DELETE	BLOCK TRUNC	DELETE CH/WRD	EXIT EDITOR	→
---	------------------	-----------------	----------------	------------------	----------------	---

You can edit the program in three ways:

- use QuickView to change the information at the prompts
- use the part program editor to change the paramacro values
- use the part program editor to change the part program

Changing the Part Program through QuickView

If you need to modify the program, you can do so by entering different information at the prompts in QuickView. Since the control generates a new part program after you save the QuickView changes, you need to delete or rename the previous program you made with QuickView before you enter new information. If the previous program is not deleted or renamed the control will generate a part program, and insert it within the existing part program. The control doesn't overwrite a previous program with the new program.

Important: When you have made changes to a part program with QuickView, the prompts may retain the values you entered in another part program. In other words, if you make changes to the values in Template 1 and store them to create a part program, the values will appear when you call up that template when creating another program. The prompt values that are on the screen when you press {STORE} are the values that the control will use to create the part program, even if they are values that are "left over" from when you edited the template in another program.

Changing the Paramacro Values with the Part Program Editor

You can change the values assigned to the paramacros with the part program editor. These values are displayed in the previous screen example in lines N0004 through N0007. These values correspond to the list of prompts that are displayed in QuickView. For instance, paramacro #503 on line N0007 corresponds to the prompt HARD STOP SENSE ZONE - I. To learn how to use the part program editor, refer to the appropriate chapter in this manual.

Important: When you change the values with the part program editor, you are making edits outside the QuickView environment. These values are not passed on to the Quickview prompts, and do not appear when Quickview generates a part program.

Table 31- A
Standard T-Line-9 Paramacro Variables

Paramacro	When a 1 is assigned to this value, the control:
1000	raises the transfer bar during a transfer cycle
1001	lowers the transfer bar during a transfer cycle
1002	advances the transfer bar or completes the drill cycle
1003	returns the transfer bar or retracts the tool
1004	stops the part program execution for any manual operations that need to be performed on the station. If the PLC assigns a 0 to this variable, then the control goes into automatic execution mode. A cycle advance/return or transfer bar/raise lower request must be made by the PLC for the CNC to begin automatic axis motion.
1005	performs the cycle in metric units. If this bit is set to 0, then the cycle is performed in English units.
Paramacro	When the control assigns a 1 to this value, it indicates to PAL that:
1100	the transfer bar has been completely raised
1101	the transfer bar has been completely lowered
1102	the transfer bar is fully advanced or the drill cycle is at the cycle complete position
1103	the transfer bar is fully returned or the drill cycle is at the cycle retract position
1104	the tool is fully returned to the tool change position (as specified in the part program)

If you want to activate a paramacro through remote I/O or through the fiber optic ring, use this table to determine what remote I/O flag or ring point is assigned to each variable:

QV#	Application	Paramacro Variables										
		1000	1001	1002	1003	1004	1005	1100	1101	1102	1103	1104
01	Drilling Cycle, No Dwell/Rapid Out			○	○	○	○			○	○	○
02	Drilling Cycle, Dwell/Rapid Out			○	○	○	○			○	○	○
03	Deep Hole Drill Cycle, Chip Clear			○	○	○	○			○	○	○
04	Deep Hole Drill Cycle, Chip Break			○	○	○	○			○	○	○
05	Right-Hand Tapping Cycle			○	○	○	○			○	○	○
06	Right-Hand Solid-Tapping Cycle			○	○	○	○			○	○	○
07	Left-Hand Tapping Cycle			○	○	○	○			○	○	○
08	Left-Hand Solid-Tapping Cycle			○	○	○	○			○	○	○
09	Boring Cycle, No Dwell/Feed Out			○	○	○	○			○	○	○
10	Boring Cycle, Spindle Shift			○	○	○	○			○	○	○
11	Boring Cycle, Spindle Stop/Rapid Out			○	○	○	○			○	○	○
12	Back Boring Cycle			○	○	○	○			○	○	○
13	Boring Cycle, Dwell/Feed Out			○	○	○	○			○	○	○
14	Single Axis Lift Cycle	○	○				○	○	○	○	○	○
15	Single Axis Transfer Cycle			○	○		○	○	○	○	○	○
16	Two-Axis Transfer Bar Cycle			○	○		○	○	○	○	○	○
17	Single Axis Cross Cycle	○	○			○	○	○	○	○	○	○
18	Single Axis Feed Cycle	○	○			○	○	○	○	○	○	○
19	Two-Axis Cross Feed Cycle	○	○	○			○	○	○	○	○	○
	Remote I/O Bit	\$RMI2.00	\$RMI2.01	\$RMI2.02	\$RMI2.03	\$RMI2.04	\$RMI2.05	\$RMO2.00	\$RMO2.01	\$RMO2.02	\$RMO2.03	\$RMO2.04
	Ring I/O Point	A00	A01	A02	A03	A04	A05	A11	A12	A13	A14	A15
Flag Description												
Raise transfer bar												
Lower transfer bar												
Advance transfer bar or drill cycle												
Return transfer bar or drill cycle												
Return drill cycle axis to the tool change position												
Stop program cycle for local, manual control; Set-up mode												
Transfer bar at raised position												
Transfer bar at lowered position												
Transfer bar at advanced position or drill cycle at cycle complete position												
Transfer bar at returned position or drill cycle at cycle returned position												
Drill cycle axis at tool change position												

Changing the Program with the Part Program Editor

You can change program generated by QuickView with the part program editor. To learn how to use the part program editor, refer to the appropriate chapter in this manual. The changes you make to to the part program will be specific to that part program and will not change the templates. The templates are kept in the protected directory, and are used by QuickView every time it generates a program.

This table lists special M codes that are used to communicate with the control's PAL program and with Allen Bradley remote I/O.

Table 31- B
Standard T-Line-9 M codes

M codes	Description:
M90	Transfer Clear
M91	Transfer Not Clear

31.1.5 Reloading Part Program Templates

Part program templates were loaded on your control when it was shipped from Allen-Bradley. They are however stored in a volatile area of control memory (requires power to be maintained). Common causes for losing the part program templates stored in this area are:

Memory Lost Because:	Detail:
Power Loss	The control comes standard with 5 to 7 days of supercap power to maintain volatile memory. You can purchase an optional battery backup that will maintain this memory for 6 months without power.
Format RAM Disk	Various control conditions can result in memory becoming corrupt. When this occurs you must perform a reformat operation on the volatile memory area which removes any part program templates.

You can identify that part program templates have been lost by trying to create a part program using the QuickView screens as discussed on page 31-12. When you select the part program type to create the control will display the error "PART PROGRAM TEMPLATE MISSING". This message tells you that the transfer line part program templates are missing or corrupted.

Replacing the Part Program Templates

If for some reason the part program templates are lost, follow this procedure to re-install the project. You must first obtain an ODS backup of the T-Line-9 standard program templates from the Allen-Bradley Motion Control Division. This ODS backup project is included on floppy disk with this documentation or can be obtained using the Allen-Bradley Motion Control Division electronic bulletin board (216)-646-3963.

The transfer line project can be restored using a personal computer with the Allen-Bradley ODS software installed on its hard drive. Contact your Allen-Bradley sales office if you do not have a copy of the ODS software.

Restore the project to ODS as described in your ODS user's manual. See the section titled Restoring a Backed up Project in the chapter titled Using ODS Project Management. The ODS workstation will copy the transfer line AMP, PAL, and part program templates from the ODS backup project titled Tran230.

The following procedure describes the download procedure for restoring the part program templates to the control.

Important: When a transfer line program template is downloaded from ODS to the control, it must be inserted into the protected program directory on the control. You can do this by selecting the protected directory on the control before beginning the download. The protected directory can be password protect to help prevent unauthorized edits. Refer to your 9/Series operation and programming manual for details.

On the control, you must configure port B for serial communication. The serial communication parameters of this port must correspond to those of the workstation. Refer to your 9/Series operation and programming manual for details on configuring the 9/Series serial port (the default for port B is already configured for ODS communications).

To download a part program from ODS to the control's memory, follow these steps:

1. Select the TRANS Project. The name of the active project is displayed in the upper right portion of screen. If the transfer line project is not the active project:
 - a. Press [F2] to select the Project Menu.
 - b. Press [0] to open the transfer line project.
 - c. Use the cursor keys to select the Transferline project and press [ENTER]. The workstation shows the active project as Transferline.
2. Connect your ODS workstation to the control using the serial cable discussed in chapter 6 of your ODS manual. Both the single and 3-axis 9/230 processors only have one serial port. It is labeled and configured as Port B.

- Press [F3] to pull down the Application menu.

The workstation displays this screen:

Proj: TRAN230		Appl: Upload		Util: Get PAL I/O
[F1 - File]	[F2 - Project]	[F3 - Application]	[F4 - Utility]	[F5 - Configuration]
		AMP (A)		
		PAL (P)		
		I/O Assignments (I)		
		Part Program (R)		
		Upload (U)		
		Download (D)		

- Use the arrow keys to highlight the Download application then press [ENTER] or press [D].
- Press [F4] to pull down the Utility menu.

Proj: TRAN230		Appl: Download		Util: Get PAL I/O
[F1 - File]	[F2 - Project]	[F3 - Application]	[F4 - Utility]	[F5 - Configuration]
		Send AMP params (A)		
		Send PAL and I/O (P)		
		Send Part Program (R)		

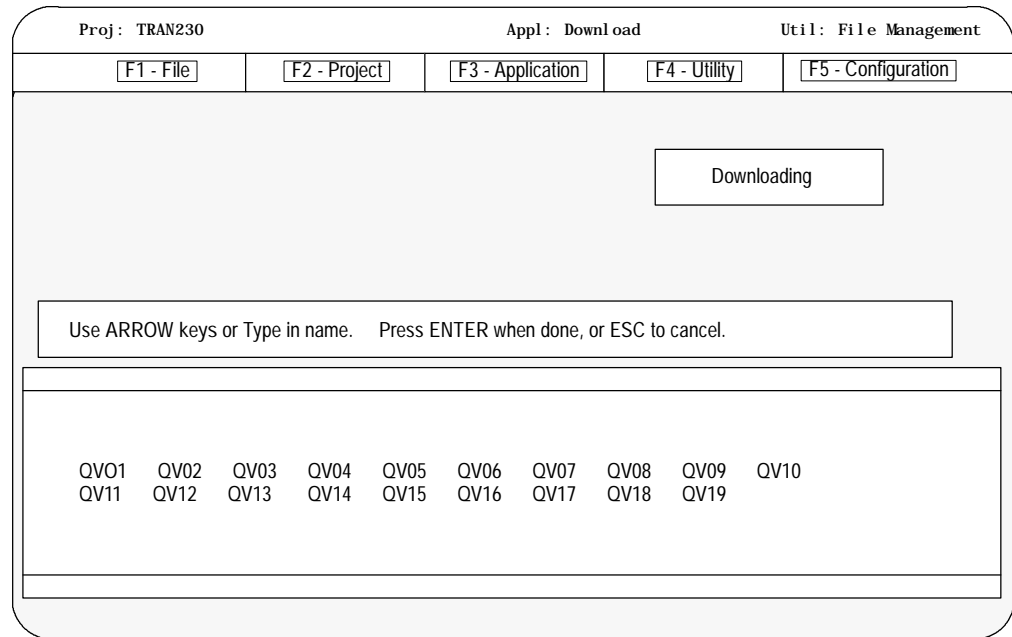
6. Use the arrow keys to highlight the Send Part Program option then press [ENTER], or press [R].

The workstation displays this screen:

Proj: TRAN230		Appl: Download		Util: File Management							
[F1 - File]	[F2 - Project]	[F3 - Application]	[F4 - Utility]	[F5 - Configuration]							
<table border="1"><tr><td colspan="2">Download Destination</td></tr><tr><td>Control</td><td>(C)</td></tr><tr><td>Storage</td><td>(S)</td></tr></table>						Download Destination		Control	(C)	Storage	(S)
Download Destination											
Control	(C)										
Storage	(S)										

7. Use the arrow keys to highlight the control as the download destination and press [ENTER], or press [C].

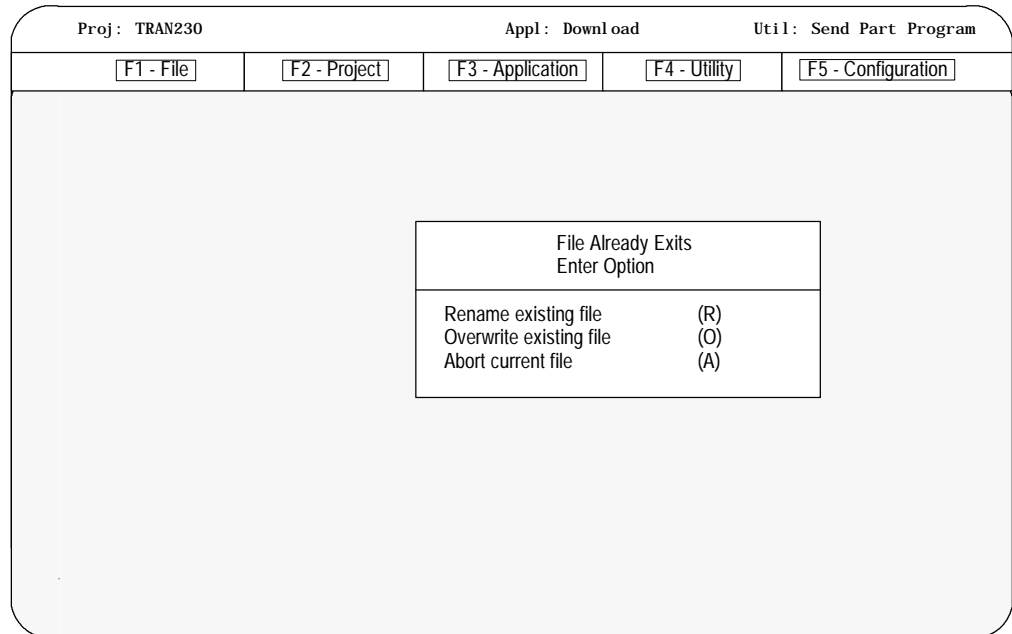
The workstation displays the part program files that are stored in the active project directory of the workstation:



8. Type in the part program name * to download all of the transfer line templates to the control, then press [ENTER].

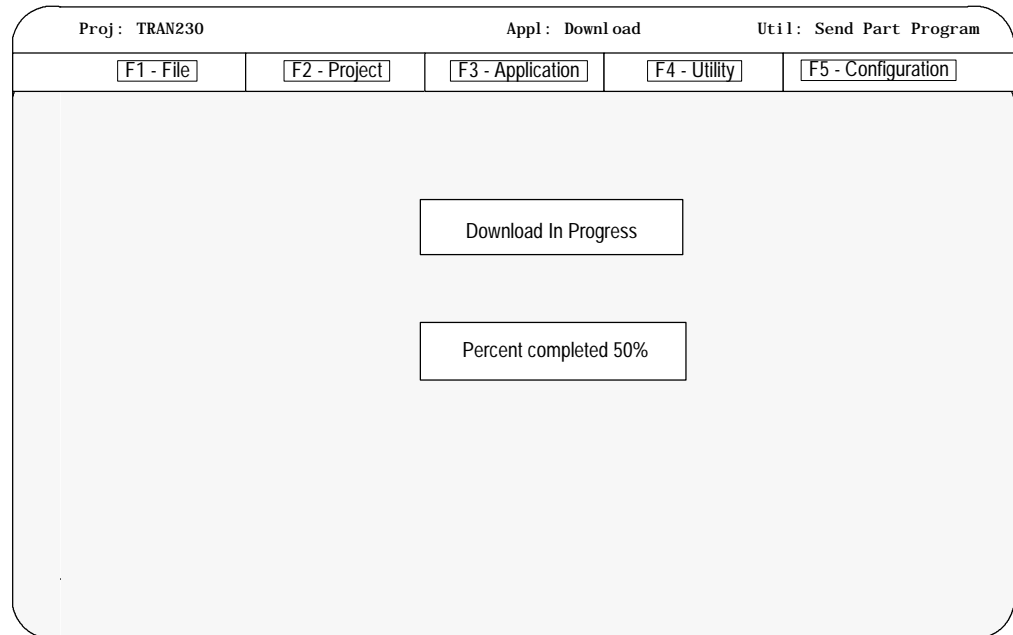
When you download a program to the control, the control does not display a message to indicate that a download is taking place. It may take several minutes for the control to complete the download. You may want to go to the directory screen and monitor the download. When the program finishes downloading, the control updates the size of the program.

If some of the program templates still exist in control memory, the workstation displays this screen:



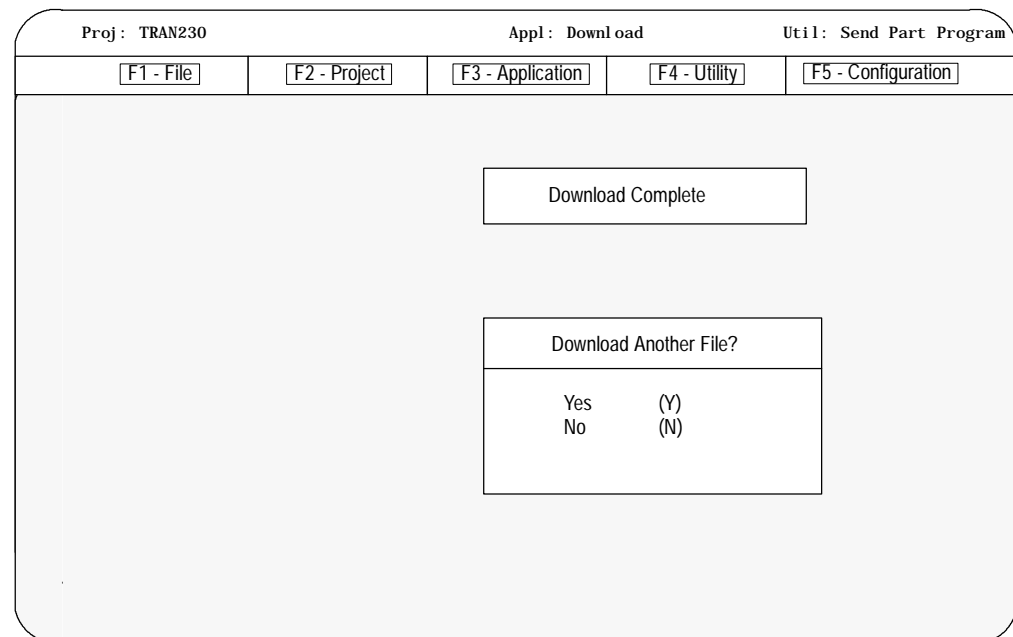
Select Overwrite existing file and press [ENTER], or press [O]. The Overwrite option will overwrite the old template and replace it with the one you are downloading.

After selecting the Rename or Overwrite option, or if the file being downloaded did not already exist on the control, the workstation displays this screen:



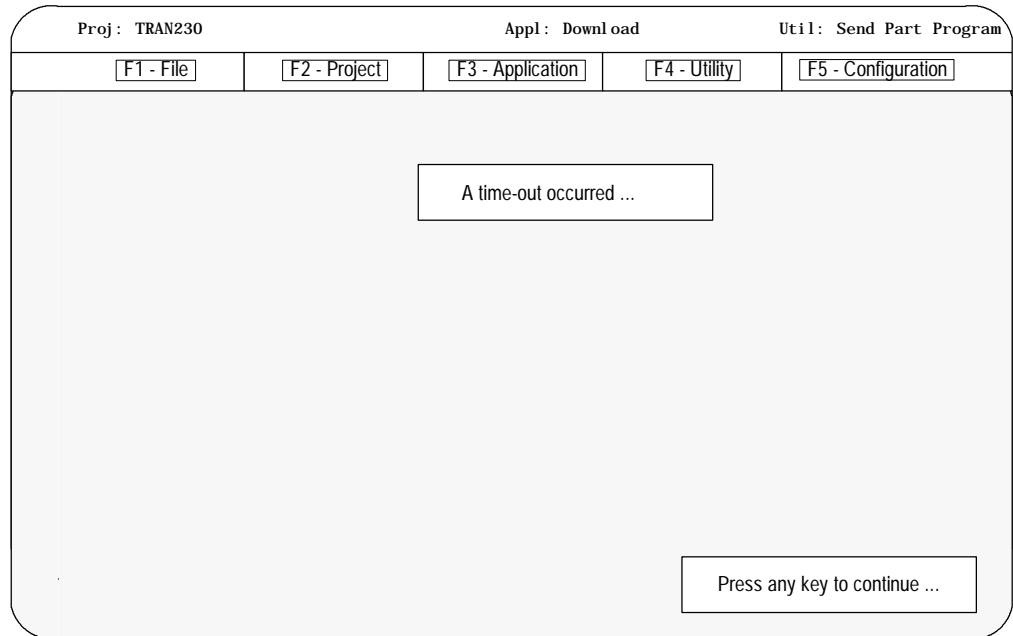
The percentage of the download process that has currently been completed is displayed on the screen. This value is updated continually throughout the download process.

When the download process is complete, you see this screen:



9. Select “No.” The workstation returns to the main menu line of ODS.

If the workstation is unable to complete the download procedure in the allotted time frame due to communication failure, it displays this screen:



Press any key to return to the ODS main menu.

If the Download Failed

If the download failed and the message “A time-out occurred ...” is displayed on the workstation you should check the following.

- **Device Setup** - The controls port that the ODS workstation is connected to must be configured to communicate to ODS. This is performed through the Device Setup softkey. Refer to your Mill Operation and Programming manual.
- **Cable Connection** - Make sure you made the correct cable as described in chapter 6 of your ODS reference manual. Make sure all cable connections are tight.
- **Program Active or being Edited** - Make sure you have not selected one of the part program templates as the active program (selected for execution) or as open for editing. Programs that are active or open for editing display either an A or an E next to the program name on the part program directory. You must exit the editor or deactivate the part program template before that specific program will download.

31.1.6 Running the Cycles

Once you enter information in the QuickView screens, the cycle acts just like a part program. The program runs within these conditions:

- G90 - all the cycles operate in absolute mode. If you enter try to a G91 into the QuickView prompt you'll get an error message.



WARNING: If you enter a G91 into a transfer line part program, the control will run in incremental mode, even though the program was designed for absolute mode. Using G91 in any of the transfer line part programs may result in unpredictable axis motion.

- G98/99 - all the cycles return to the R plane just as if a G99 was programmed. If you try to program a G98, it has no effect on the cycle.
- Plane Select, G17, G18, G19 - these G-codes have no effect on the system. Since all of the cycles in the transfer line option use only two axes, plane select doesn't apply. In all of the cycles:
 - the first axis is the drilling axis. Its default name is X.
 - the second axis is the positioning axis. Its default name is Y.
 - if you select only one axis, it will be the drilling axis.

Using Spindles with the Cycles

If the system is using the default AMP and PAL, spindle command is sent from PAL to the PLC, and the PLC controls the spindle. Your system installer may have modified the system's PAL and AMP if you are using a solid tapping cycle, or if the configuration requires the spindle to be controlled by the CNC.

31.2 Understanding the QuickView Templates

This section defines the QuickView prompts for all the templates, and shows the QuickView graphic for each cycle.

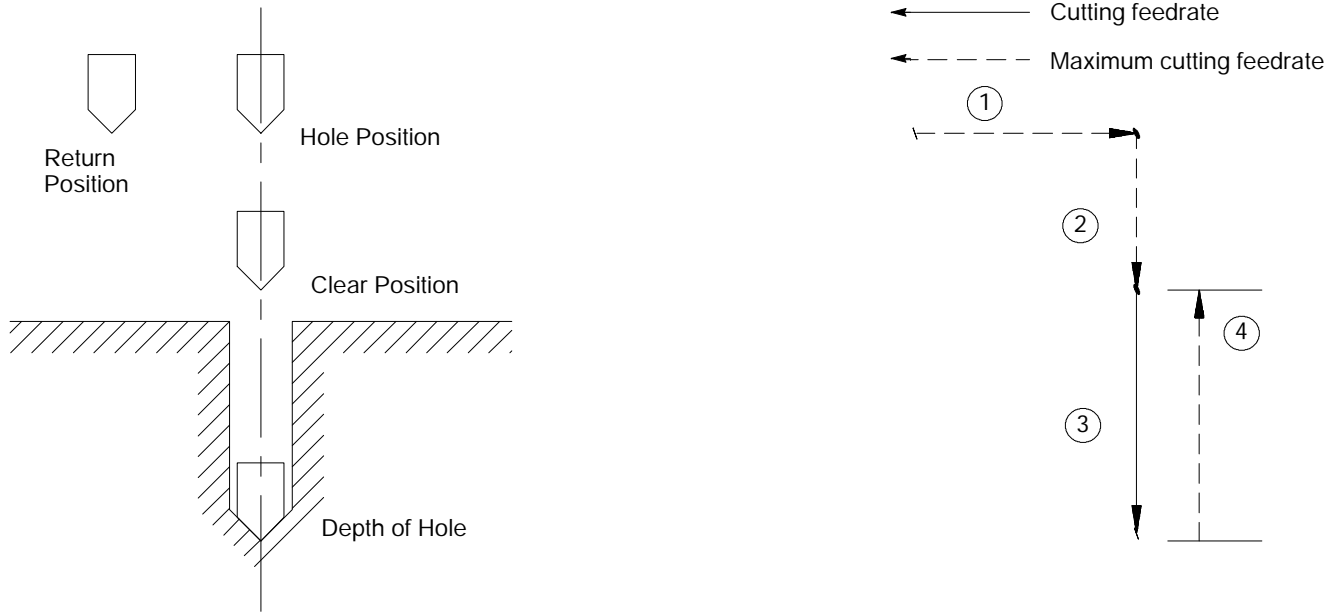
All of the templates include required fields that need to be completed for the control to create the cycle. Some cycles includes optional fields for features such as Feed to Hard Stop, Adaptive Depth, and Adaptive Feed. These features are mutually exclusive. If you enter data for more than one feature, an error message appears.

Template 1: Drilling Cycle, No Dwell/Rapid Out

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a drilling operation.
X	503	DEPTH OF HOLE	The location to which the tool drills into the part.
R	504	CLEAR POSITION	The location the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The cutting feedrate for drilling/boring operations. This is also the maximum feedrate for operations that use adaptive depth.
X	508	TOOL CHANGE POSITION	The location at which you can perform a tool change operation.
E	509	ADAPTIVE FEED MINIMUM	This is the minimum feedrate at which the control performs an adaptive feed operation. The control will try to maintain this minimum feedrate, even if it means increasing the adaptive feed torque percent.
Q	510	ADAPTIVE FEED TORQUE %	This is the amount of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1 to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. Important: The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
I	511	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	512	ADAPTIVE DEPTH INCREMENT	The amount of distance that the control will increment the tool into the part during an adaptive depth operation.
M	513	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	514	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.4
Drilling Cycle without Dwell

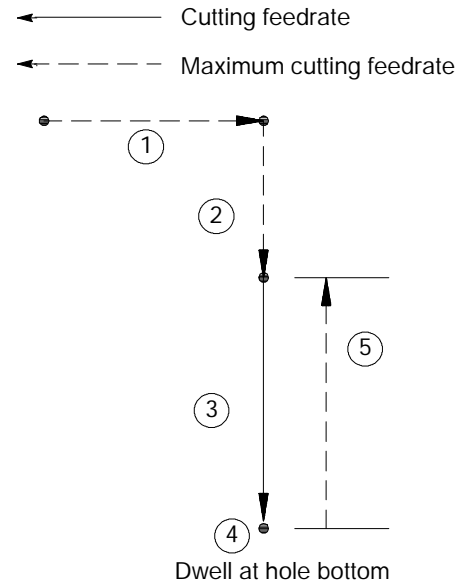
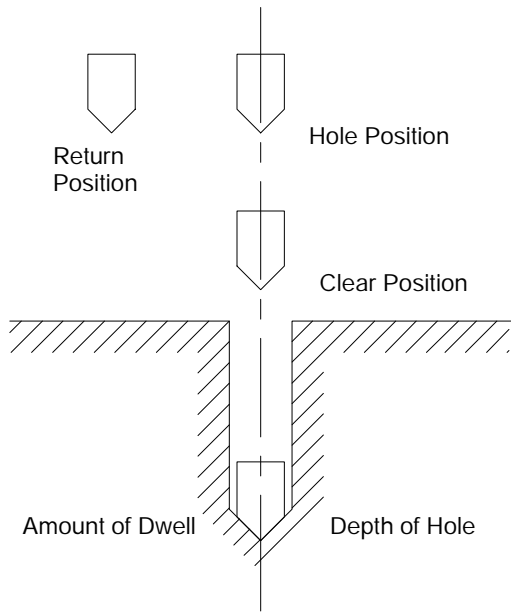


Template 2: Drilling Cycle, Dwell/Rapid Out

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a drilling operation.
X	503	DEPTH OF HOLE	The location to which the tool drills into the part.
R	504	CLEAR POSITION	The location to which the tool retracts after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The cutting feedrate for drilling/boring operations. This is also the maximum feedrate for operations that use adaptive depth.
P	508	AMOUNT OF DWELL	The amount of time the axis pauses before it retracts from the hole; measured in seconds.
X	509	TOOL CHANGE POSITION	The location at which you can perform a tool change operation.
E	510	ADAPTIVE FEED MINIMUM	This is the minimum feedrate at which the control performs an adaptive feed operation. The control will try to maintain this minimum feedrate, even if it means increasing the adaptive feed torque percent.
Q	511	ADAPTIVE FEED TORQUE %	This is the amount of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1 to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. Important: The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
I	512	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	513	ADAPTIVE DEPTH INCREMENT	The amount of distance that the control will increment the tool into the part during an adaptive depth operation.
M	514	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	515	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.5
Drilling Cycle, Dwell/Rapid Out



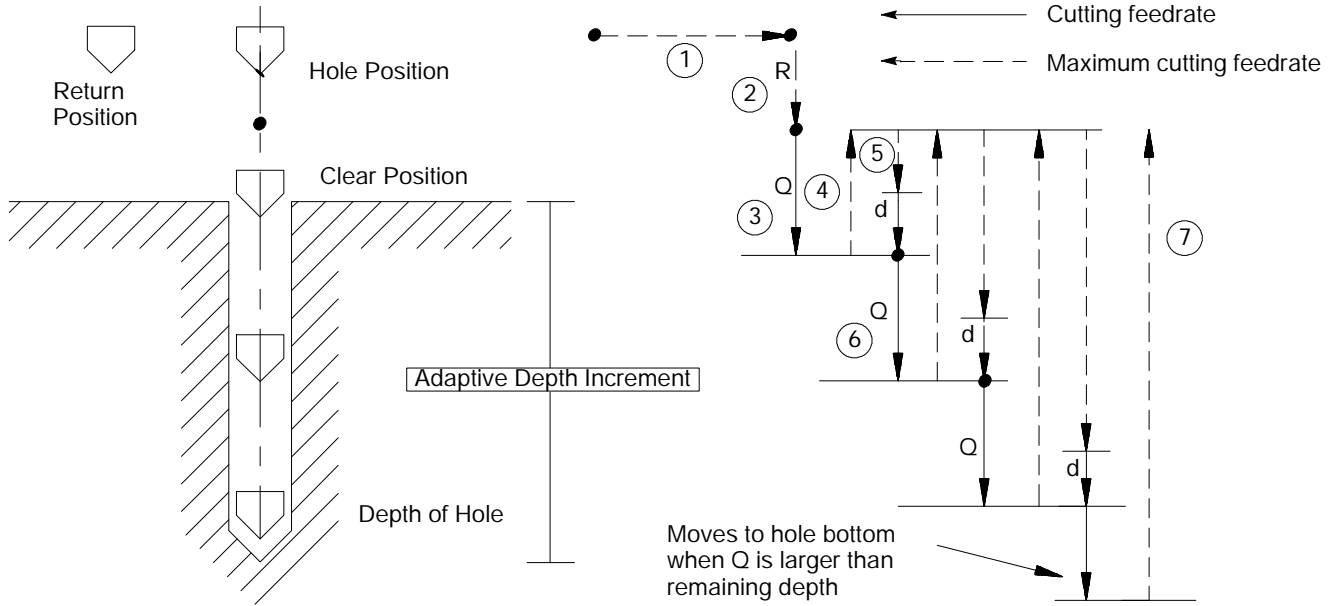
Template 3: Deep Hole Drill Cycle, Chip Clear

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a drilling operation.
X	503	DEPTH OF HOLE	The location to which the tool drills into the part. If the cycle uses adaptive depth, this position needs to be beyond the adaptive depth increment.
R	504	CLEAR POSITION	The location to which the tool retracts after an operation. It is completely free of the part. This also known as the R plane.
Q	505	INFEEED AMOUNT	This defines the infeed amount for each step into the hole.
X, Y	506, 507	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	508	FEEDRATE	The feedrate for drilling/boring operations. This is also the maximum feedrate for operations that use adaptive depth.
X	509	TOOL CHANGE POSITION	The location at which you can perform a tool change operation.
E	510	ADAPTIVE FEED MINIMUM	This is the minimum feedrate at which the control performs an adaptive feed operation. The control will try to maintain this minimum feedrate, even if it means increasing the adaptive feed torque percent.
Q	511	ADAPTIVE FEED TORQUE %	This is the amount of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1 to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. Important: The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
I	512	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	513	ADAPTIVE DEPTH INCREMENT	The amount of distance between the surface of the part and the depth of hole.
M	514	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	515	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

d - This value is written into the cycle. It is .01 inches (.254 mm).

Figure 31.6
Deep Hole Drill Cycle, Chip Clear



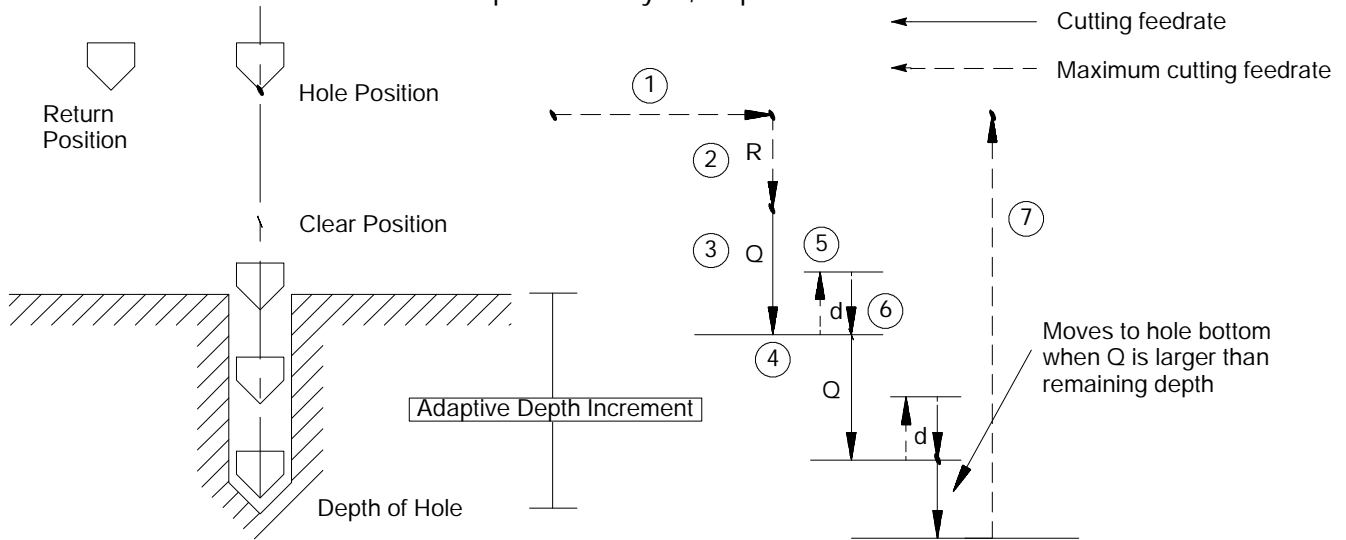
Template 4: Deep Hole Drill Cycle, Chip Break

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a drilling operation.
X	503	DEPTH OF HOLE	The location to which the tool drills into the part. If the cycle uses adaptive depth, this position needs to be beyond the adaptive depth increment.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
Q	505	INFEEED AMOUNT	This defines the infeed amount for each step into the hole
X, Y	506, 507	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	508	FEEDRATE	The feedrate for drilling/boring operations. This is also the maximum feedrate for operations that use adaptive depth.
P	509	AMOUNT OF DWELL	The amount of time the axis pauses before it retracts from the hole; measured in seconds.
X	510	TOOL CHANGE POSITION	The location where the controls starts and stops a cycle.
E	511	ADAPTIVE FEED MINIMUM	This is the minimum feedrate at which the control performs an adaptive feed operation. The control will try to maintain this minimum feedrate, even if it means increasing the adaptive feed torque percent.
Q	512	ADAPTIVE FEED TORQUE %	This is the amount of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1 to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. Important: The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
I	513	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	514	ADAPTIVE DEPTH INCREMENT	The amount of distance between the surface of the part and the depth of hole.
M	515	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	516	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

d - This value is written into the cycle. It is .01 inches (.254 mm).

Figure 31.7
Deep Hole Drill Cycle, Chip Break

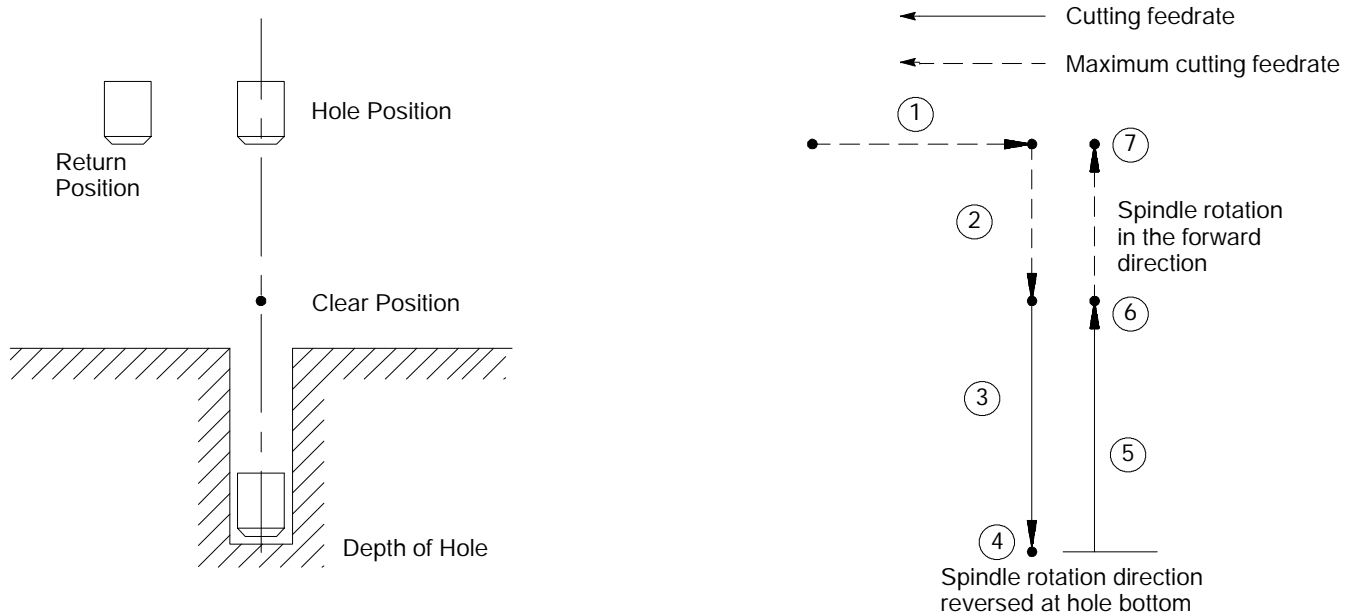


Template 5: Right-Hand Tapping Cycle

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a tapping operation.
X	503	DEPTH TO TAP	The location to which the tool taps into the part.
R	504	CLEAR POSITION	The location the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The feedrate for tapping operations, and all moves represented by the solid lines in the QuickView screens.
P	508	AMOUNT OF DWELL	The amount of time the axis pauses before it retracts from the hole; measured in seconds.
X	509	TOOL CHANGE POSITION	The location to which the control moves so that a tool change may be performed.
S	510	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.8
Right-Hand Tapping Cycle

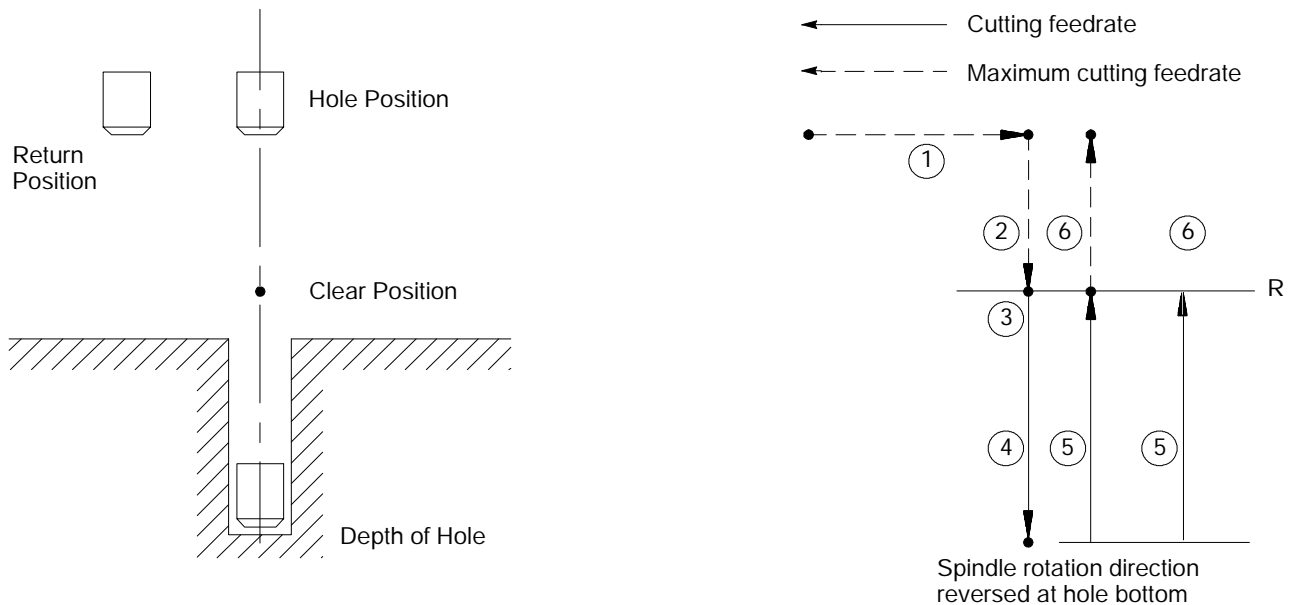


Template 6: Right-Hand Solid-Tapping Cycle

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location the tool moves to before it begins a tapping operation.
X	503	DEPTH TO TAP	The location where the tool completes the tap into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
S	507	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.
F	508	TAP LEAD	The thread lead along the drilling axis. This value gives the cycles its feedrate. Interpreted as inches per revolution or millimeters per revolution, depending on if English or metric mode is active.
X	509	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
Q	510	ORIENT ANGLE	The angle the spindle orients to before starting the tap.
D	511	TAP-OUT SPEED	The speed of the spindle as the tap returns to the R plane.

Required entry Optional entry

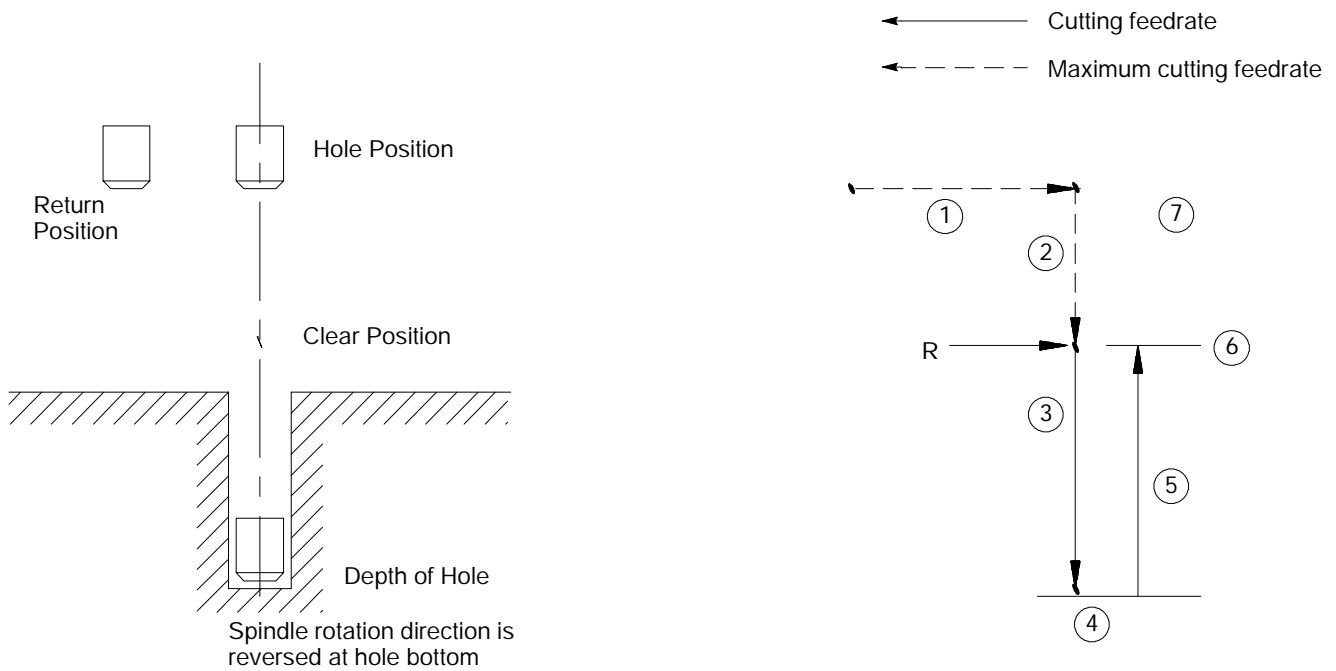
Figure 31.9
Right-Hand Solid-Tapping Cycle



Template 7: Left-Hand Tapping Cycle

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location the tool moves to before it begins a tapping operation.
X	503	DEPTH TO TAP	The location where the tool completes the tap into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The feedrate for tapping operations, and all moves represented by the solid lines in the QuickView screens.
P	508	DWELL TIME	The amount of time the axis pauses before it retracts from the hole; measured in seconds.
X	509	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
S	510	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.
<input type="checkbox"/> Required entry <input type="checkbox"/> Optional entry			

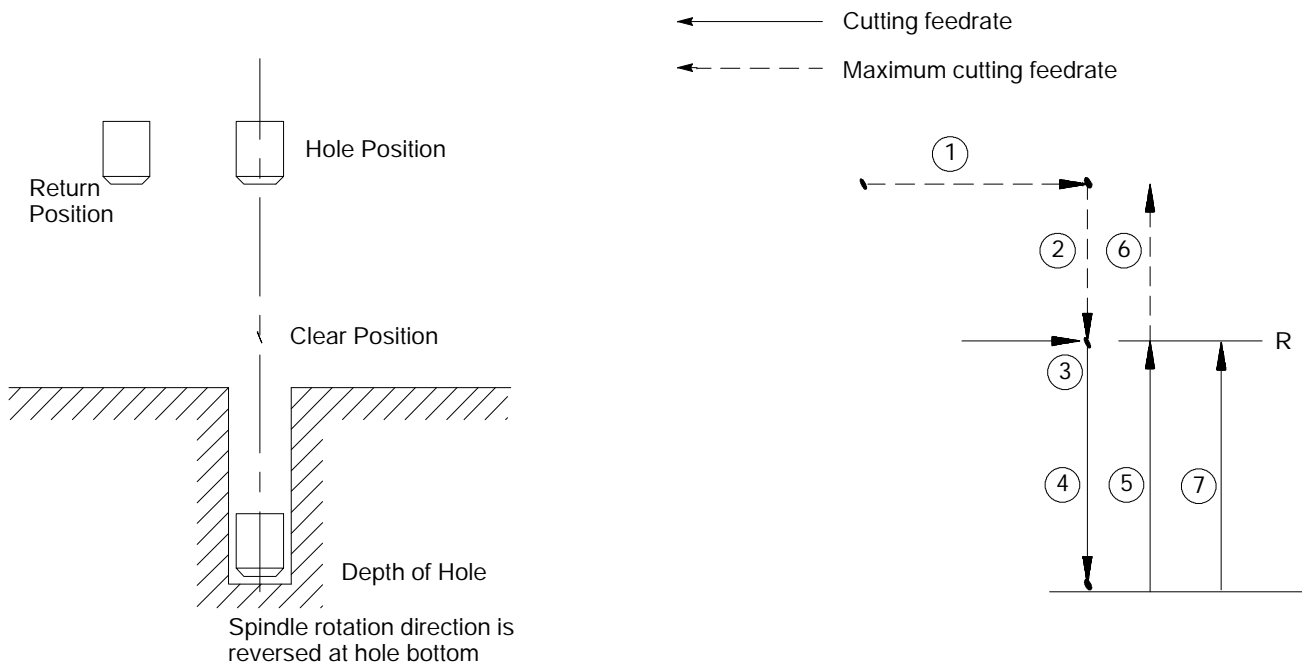
Figure 31.10
Left-Hand Tapping Cycle



Template 8: Left-Hand Solid Tapping Cycle

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location the tool moves to before it begins a tapping operation.
X	503	DEPTH TO TAP	The location where the tool completes the tap into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
S	507	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.
F	508	TAP LEAD	The thread lead along the drilling axis. This value gives the cycles its feedrate. Interpreted as inches per revolution or millimeters per revolution, depending on if English or metric mode is active.
X	509	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
Q	510	ORIENT ANGLE	The angle the spindle orients to before starting the tap.
D	511	TAP-OUT SPEED	The speed of the spindle as the tap returns to the R plane.
<input type="checkbox"/> Required entry <input type="checkbox"/> Optional entry			

Figure 31.11
G74.1: Left-Hand Solid-Tapping Cycle

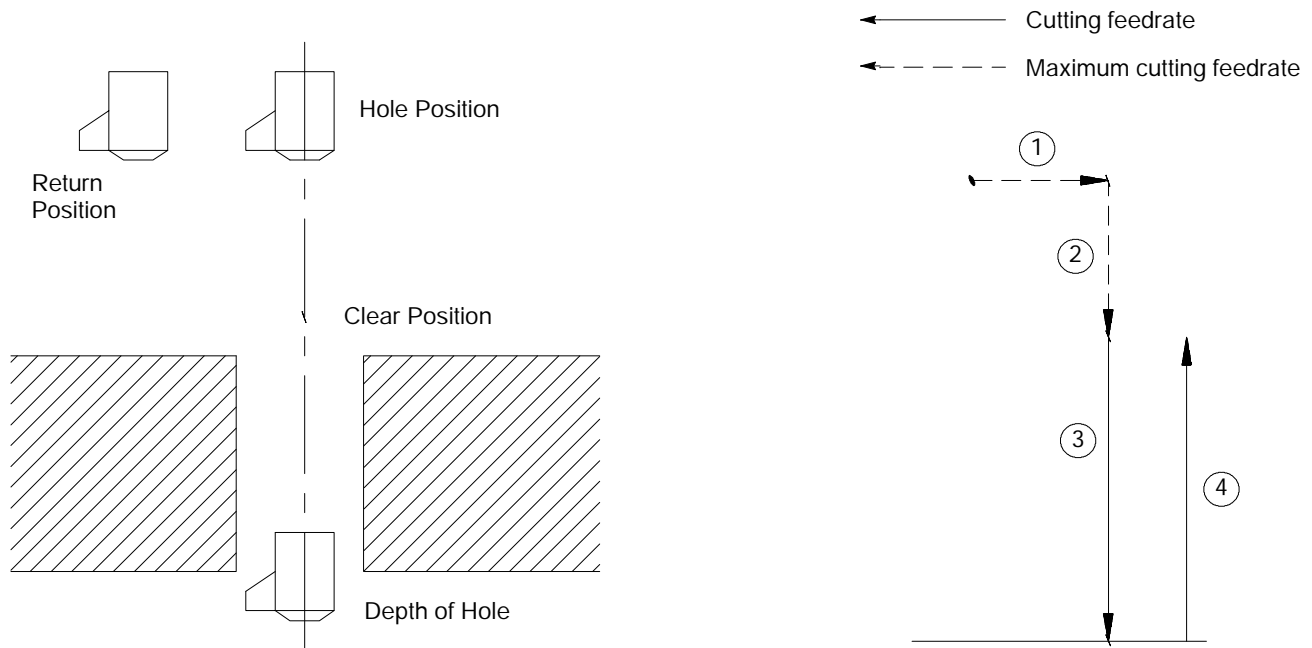


Template 9: Boring Cycle, No Dwell/Feed Out

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a boring operation.
X	503	DEPTH OF HOLE	The location to which the tool bores into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The feedrate for drilling/boring operations. This is also the maximum feedrate for operations that use adaptive depth.
X	508	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
I	509	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	510	ADAPTIVE DEPTH INCREMENT	The amount of distance that the control will increment the tool into the part during an adaptive depth operation.
M	511	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	512	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.12
Boring Cycle (Without Dwell, Feed Out)

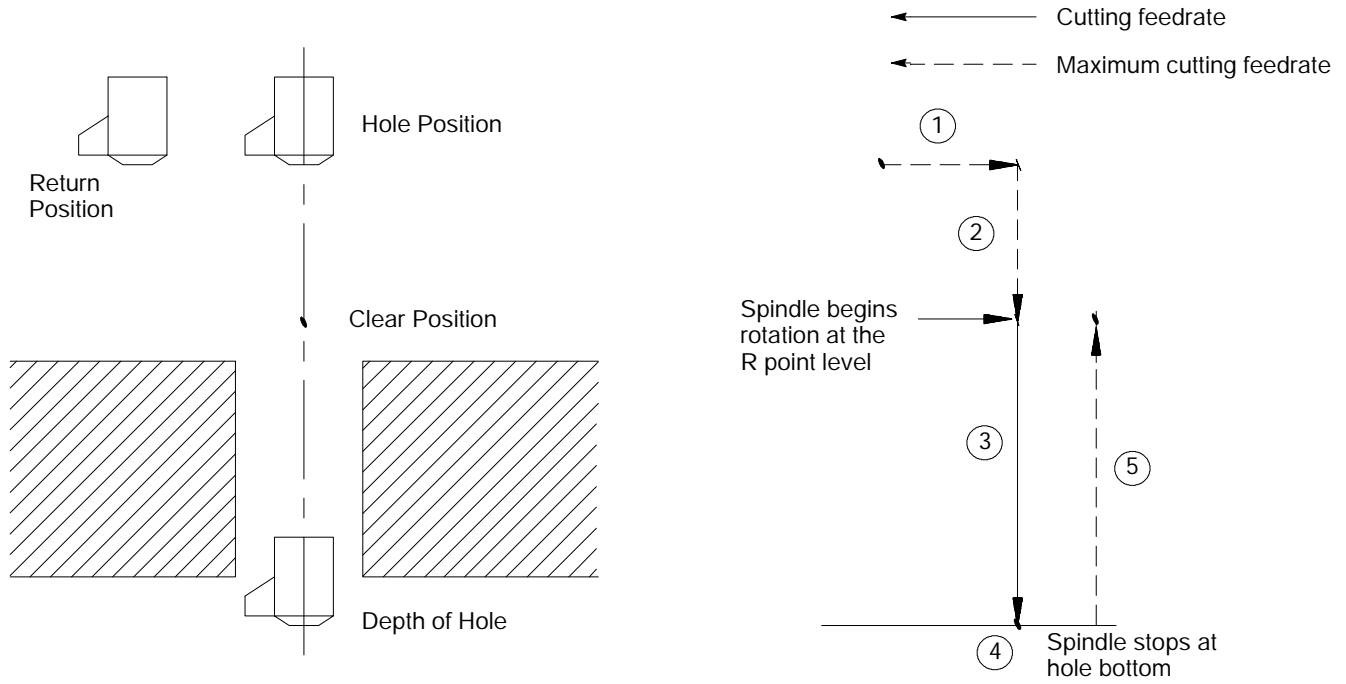


Template 10: Boring Cycle, Spindle Stop/Rapid Out

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a boring operation.
X	503	DEPTH OF HOLE	The location to which the tool bores into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The feedrate for drilling/boring operations, and all moves represented by the solid lines in the QuickView screens. This is also the maximum feedrate for operations that use adaptive depth.
P	508	AMOUNT OF DWELL	The amount of time the axis pauses before it retracts from the hole; measured in seconds.
X	509	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
I	510	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	511	ADAPTIVE DEPTH INCREMENT	The amount of distance that the control will increment the tool into the part during an adaptive depth operation.
M	512	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	513	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.13
Boring Cycle, Spindle Stop/Rapid Out

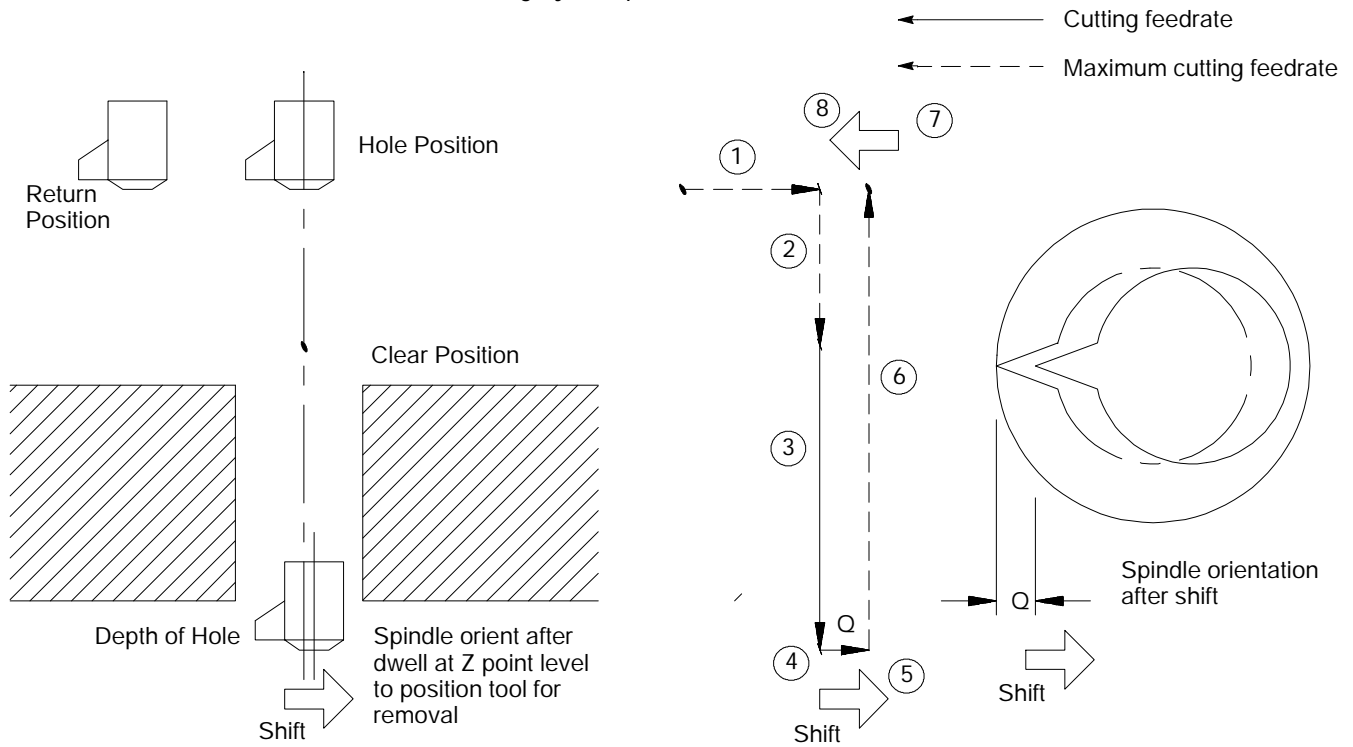


Template 11: Boring Cycle, Spindle Shift

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a boring operation.
X	503	DEPTH TO TAP OR HOLE	The location to which the tool bores into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The feedrate for drilling/boring operations, and all moves represented by the solid lines in the QuickView screens. This is also the maximum feedrate for operations that use adaptive depth.
Q	508	SHIFT AMOUNT	The amount of distance the spindle shifts so that the tool can retract without damaging the part. The orient of the part when it completes the cut is set in AMP.
X	509	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
I	510	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	511	ADAPTIVE DEPTH INCREMENT	The amount of distance that the control will increment the tool into the part during an adaptive depth operation.
M	512	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	513	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.14
Boring Cycle, Spindle Shift

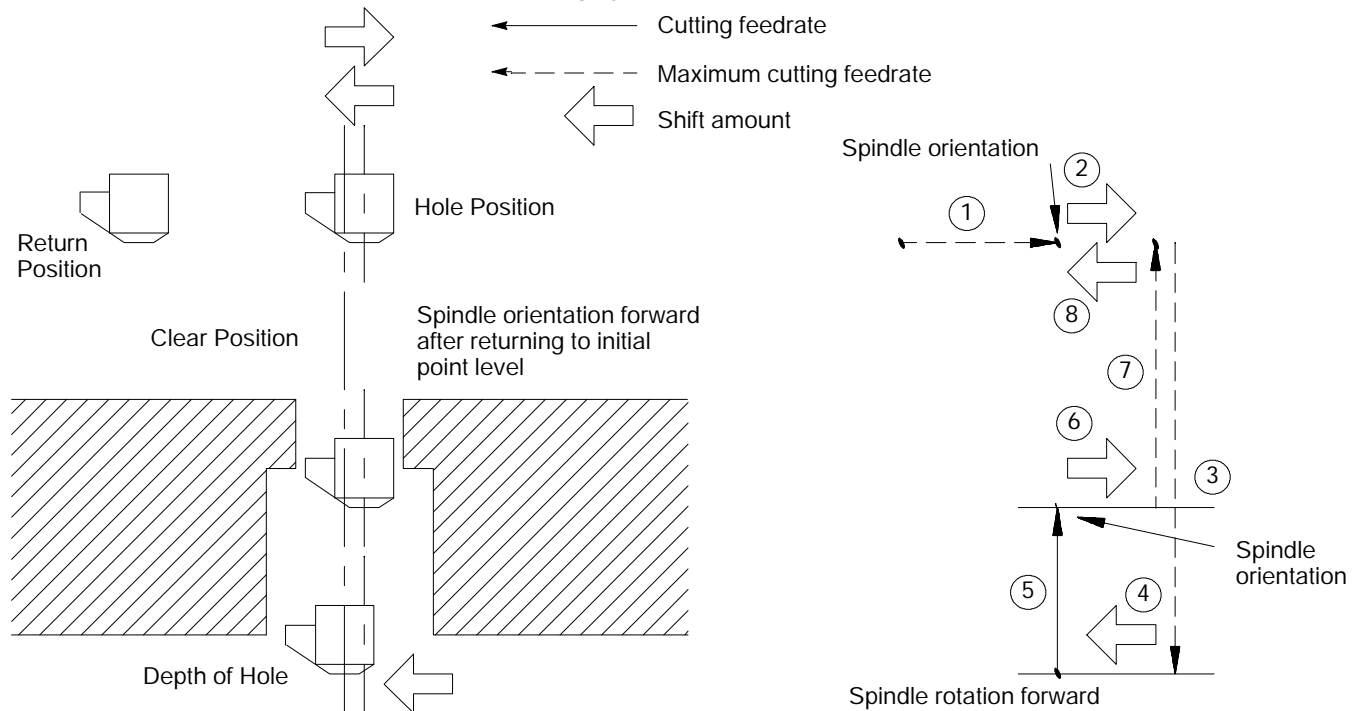


Template 12: Back Boring Cycle

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a boring operation.
X	503	DEPTH OF HOLE	The location to which the tool bores into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The feedrate for drilling/boring operations, and all moves represented by the solid lines in the QuickView screens. This is also the maximum feedrate for operations that use adaptive depth.
Q	508	SHIFT AMOUNT	The amount of distance that spindle shifts along the AMPed axis. The axis that the spindle orients to after the cut is set in AMP. This cannot be used if the I and J words are used.
X	511	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
I	512	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	513	ADAPTIVE DEPTH INCREMENT	The amount of distance that the control will increment the tool into the part during an adaptive depth operation.
M	514	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	515	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.15
Back Boring Cycle

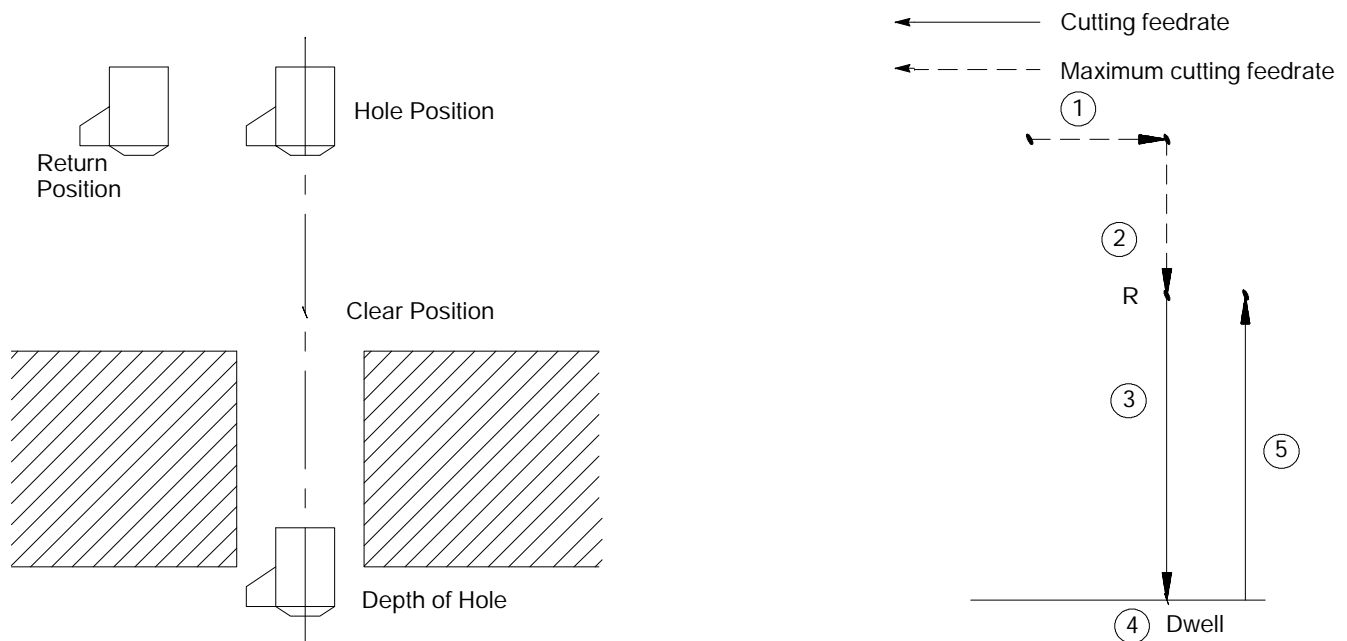


Template 13: Boring Cycle, Dwell/Feed Out

Letter	Paramacro	Label	Description
G	500	G90/91	G-codes G90 or G91 for absolute or incremental modes. At this time only absolute mode, G90, is available.
X, Y	501, 502	HOLE POSITION X, Y	The location to which the tool moves before it begins a boring operation.
X	503	DEPTH OF HOLE	The location to which the tool bores into the part.
R	504	CLEAR POSITION	The location that the tool retracts to after an operation. It is completely free of the part. This also known as the R plane.
X, Y	505, 506	RETURN POSITION X, Y	The location where the controls starts and stops a cycle.
F	507	FEEDRATE	The feedrate for drilling/boring operations, and all moves represented by the solid lines in the QuickView screens. This is also the maximum feedrate for operations that use adaptive depth.
P	508	AMOUNT OF DWELL	The amount of time the axis pauses before it retracts from the hole; measured in seconds.
X	509	TOOL CHANGE POSITION	The location the control moves to so that a tool change may be performed.
I	510	HARD STOP SENSE ZONE	Once it reaches this location, the control knows to expect a hard stop before reaching the hole bottom.
I	511	ADAPTIVE DEPTH INCREMENT	The amount of distance that the control will increment the tool into the part during an adaptive depth operation.
M	512	M03/M04	The M-code used to turn the spindle clockwise or counter-clockwise.
S	513	SPINDLE SPEED	The speed of the spindle. Measured in revolutions per minute.

Required entry Optional entry

Figure 31.16
Boring Cycle, Dwell/Feed Out

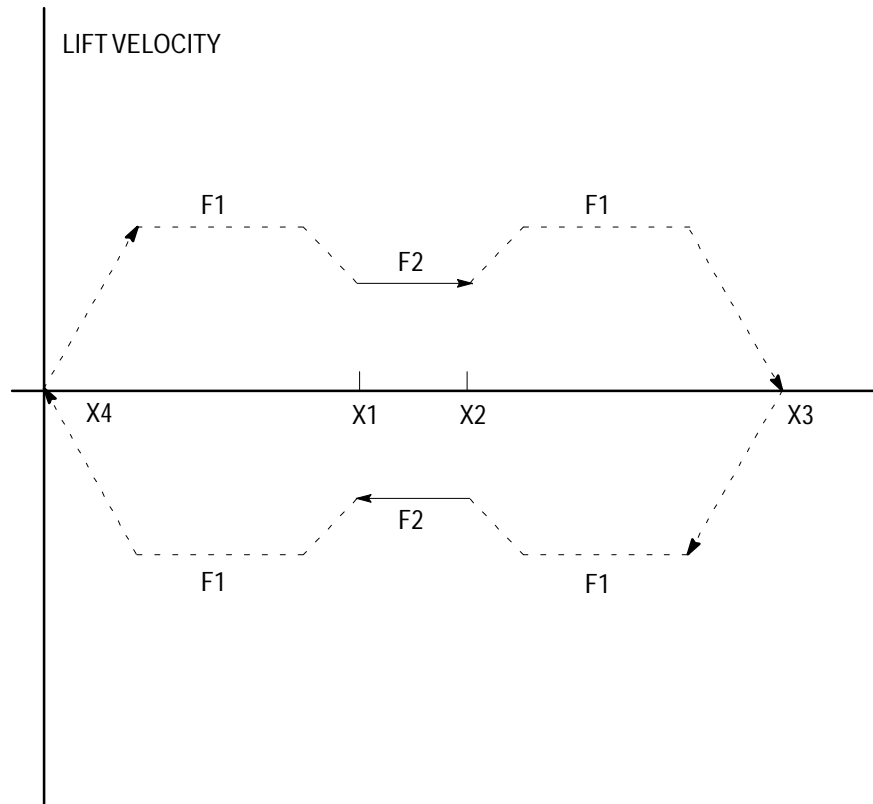


Template 14: Single Axis Lift Cycle

Letter	Paramacro	Label	Description
F ₁	500	MAX. LIFT VELOCITY	The velocity of the bar goes when it approaches a part before the low soft touch position, and after the soft touch high position.
F ₂	501	SOFT TOUCH VELOCITY	The velocity of the bar when it goes between the soft touch low position and soft touch high position.
X ₁	502	SOFT TOUCH LOW POSITION	The location where the transfer bar reduces velocity as it approaches the part.
X ₂	503	SOFT TOUCH HIGH POSITION	The location where the transfer bar has made contact with the part. Once the bar has the part, it continues at maximum lift velocity.
X ₃	504	FULL RAISE POSITION	The location where the part has been lifted to its maximum position and is about to be transferred to the station.
X ₄	505	FULL LOWER POSITION	The location where the bar has been fully lowered.

Required entry Optional entry

Figure 31.17
Single Axis Lift Cycle

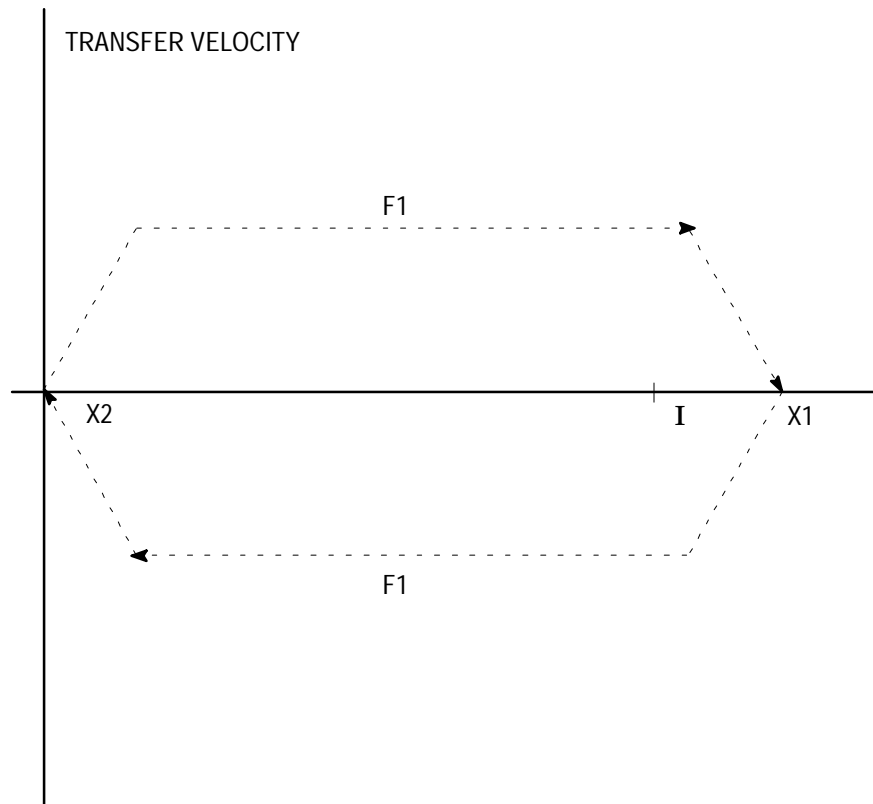


Template 15: Single Axis Transfer Cycle

Letter	Paramacro	Label	Description
F ₁	500	TRANSFER VELOCITY	The velocity of the bar as it transfers the part to the station.
X ₁	501	FULL ADVANCE POSITION	The location that indicates that the part has been fully transferred to the station.
X ₂	502	FULL RETURN POSITION	The location that indicates that the bar has returned to the station it started at.
I	503	HARD STOP SENSE ZONE	A zone in which a hard stop may be present.

Required entry Optional entry

Figure 31.18
Single Axis Transfer Cycle

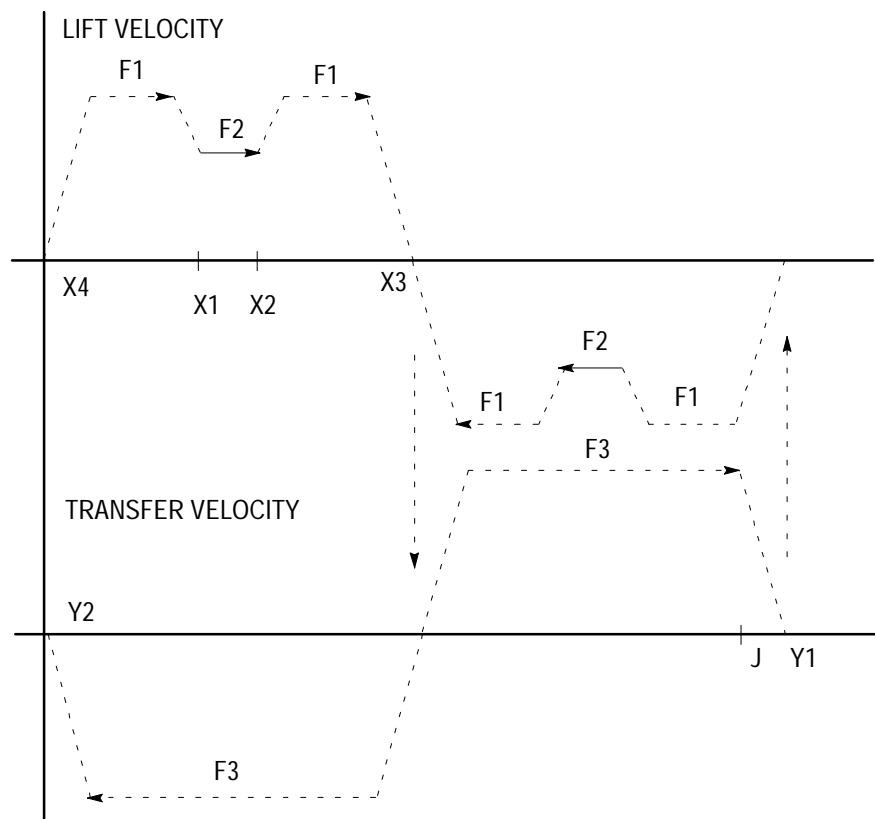


Template 16: Two-Axis Transfer Bar Cycle

Letter	Paramacro	Label	Description
F ₁	500	MAX. LIFT VELOCITY	The velocity of the bar goes when it approaches a part before the low soft touch position, and after the soft touch high position.
F ₂	501	SOFT TOUCH VELOCITY	The velocity of the bar when it goes between the soft touch low position and soft touch high position.
X ₁	502	SOFT TOUCH LOW POSITION	The location where the transfer bar reduces velocity as it approaches the part.
X ₂	503	SOFT TOUCH HIGH POSITION	The location where the transfer bar has made contact with the part. Once the bar has the part, it continues at maximum lift velocity.
X ₃	504	FULL RAISE POSITION	The location where the part has been lifted to its maximum position and is about to be transferred to the station.
F ₃	505	TRANSFER VELOCITY	The velocity of the bar as it transfers the part to the station.
Y ₁	506	FULL ADVANCE POSITION	The location that indicates that the part has been fully transferred to the station.
Y ₂	507	FULL RETURN POSITION	The location that indicates that the bar has returned to the station it started at.
X ₄	508	FULL LOWER POSITION	The location where the bar has been fully lowered.
J	509	HARD STOP SENSE ZONE	A zone in which a hard stop may be present.

Required entry Optional entry

Figure 31.19
Two-Axis Transfer Bar Cycle

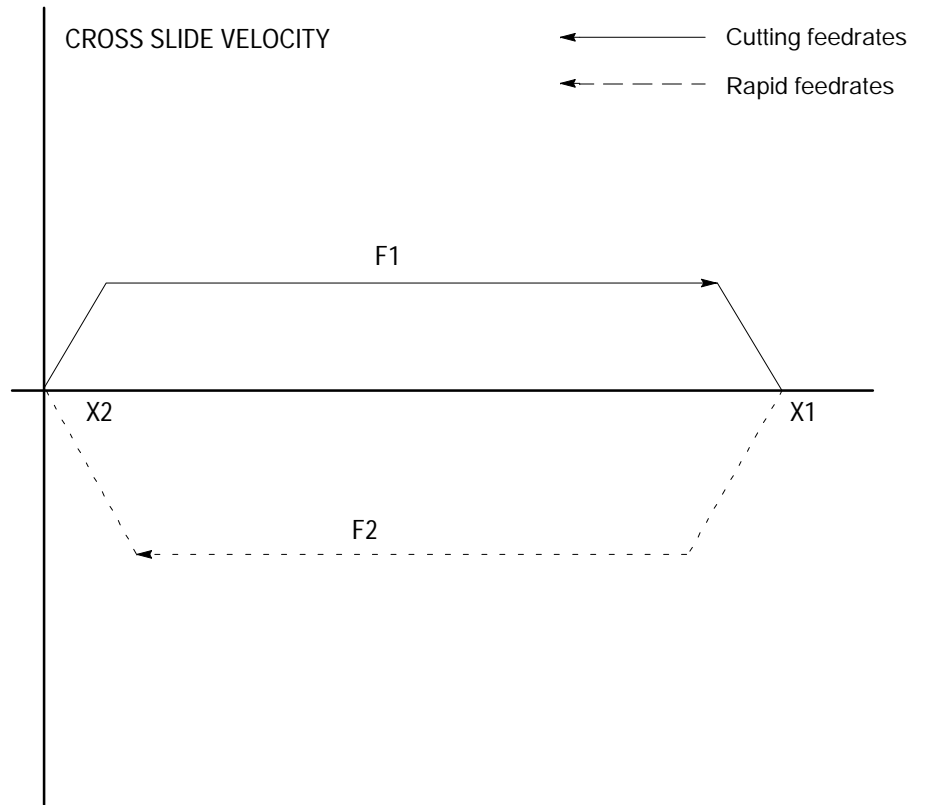


Template 17: Single Axis Cross Cycle

Letter	Paramacro	Label	Description
F ₁	500	CROSS FEEDRATE	The velocity of the tool as it traverses the part. This is the maximum feedrate if the adaptive feed feature is used.
X ₁	501	CROSS FINAL POSITION	The final position of the tool after it traverses the part.
X ₂	502	CROSS RETURN POSITION	The position of the slide when it started the cycle.
F ₂	503	CROSS RETURN RAPID	The velocity of the slide as it returns to its starting position.
E	504	ADAPTIVE FEED MINIMUM	This is the minimum feedrate at which the control performs an adaptive feed operation. The control will try to maintain this minimum feedrate, even if it means increasing the adaptive feed torque percent. For more information on adaptive feed, refer to page 18-9.
Q	505	ADAPTIVE FEED TORQUE %	This is the amount of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1 to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. Important: The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
X	506	CROSS TOOL CHANGE POSITION	The position the slide moves to so that a tool change may be performed.

Required entry Optional entry

Figure 31.20
Single Axis Cross Cycle

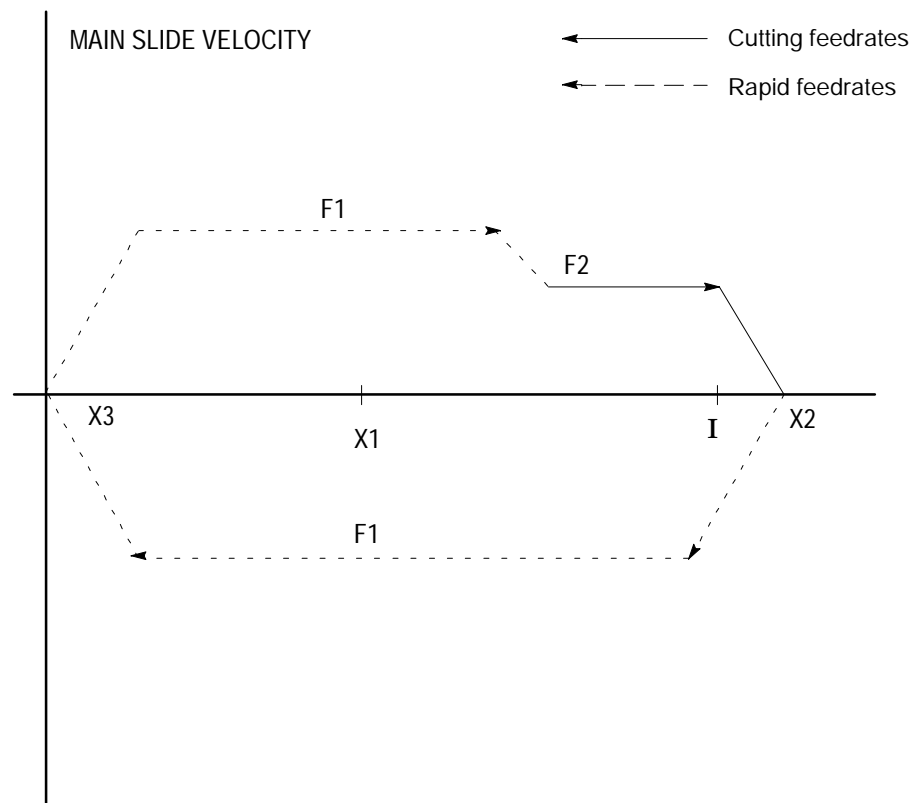


Template 18: Single Axis Feed Cycle

Letter	Paramacro	Label	Description
F ₁	500	MAIN RAPID FEEDRATE	The velocity of the tool as it approaches the part.
X ₁	501	MAIN FEED START	The position of the tool as it drills into the part.
F ₂	502	MAIN FEEDRATE	The velocity of the tool as it drills into the part.
X ₂	503	MAIN FINAL POSITION	The depth of the tool at its final position in the part.
X ₃	504	MAIN RETURN POSITION	The position of the tool when it started the cycle.
X	505	MAIN TOOL CHANGE POSITION	The position the tool moves to so that a tool change may be performed.
I	506	HARD STOP SENSE ZONE	The position that indicates to the control that a hard stop is on the axis.

Required entry Optional entry

Figure 31.21
Single Axis Feed Cycle

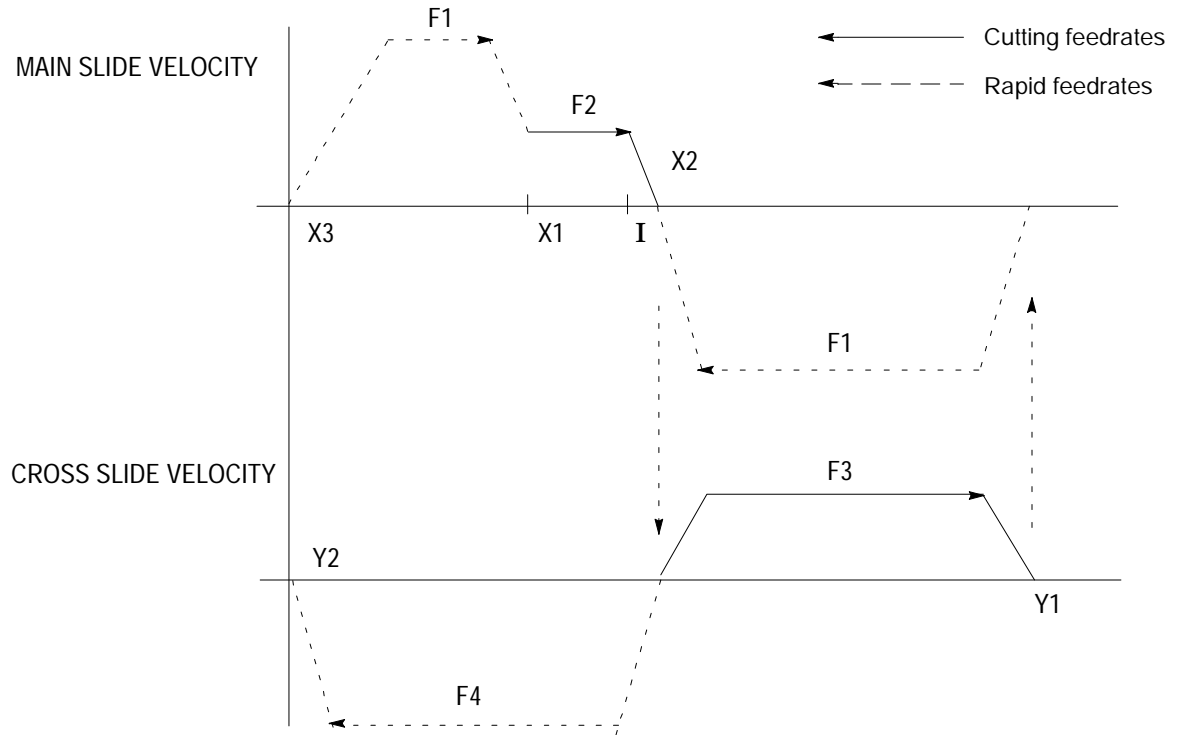


Template 19: Two-Axis Cross Feed Cycle

Letter	Paramacro	Label	Description
F ₁	500	MAIN RAPID FEEDRATE	The velocity of the tool as it approaches the part.
X ₁	501	MAIN FEED START	The position of the tool as it drills into the part.
F ₂	502	MAIN FEEDRATE	The velocity of the tool as it drills into the part.
X ₂	503	MAIN FINAL POSITION	The depth of the tool at its final position in the part.
F ₃	504	CROSS FEEDRATE (MAX)	The velocity of the tool as it traverses the part.
Y ₁	505	CROSS FINAL POSITION	The final position of the tool after it traverses the part.
X ₃	506	MAIN RETURN POSITION	The position of the tool when it started the cycle.
Y ₂	507	CROSS RETURN POSITION	The position of the slide when it started the cycle.
F ₄	508	CROSS RETURN RAPID	The velocity of the slide as it returns to its starting position.
E	509	ADAPTIVE FEED MINIMUM	This is the minimum feedrate at which the control performs an adaptive feed operation. The control will try to maintain this minimum feedrate, even if it means increasing the adaptive feed torque percent. For more information on adaptive feed, refer to page 18-9.
Q	510	ADAPTIVE FEED TORQUE %	This is the amount of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1 to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. Important: The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
Y	511	CROSS TOOL CHANGE POSITION	The position the slide moves to so that a tool change may be performed.
X	512	MAIN TOOL CHANGE POSITION	The position the tool moves to so that a tool change may be performed.
I	513	HARD STOP SENSE ZONE	The position that indicates to the control that a hard stop is on the axis.

Required entry Optional entry

Figure 31.22
Two-Axis Cross Feed Cycle



Softkey Tree

Appendix Overview

This appendix explains softkeys and includes maps of the softkey trees.

Understanding Softkeys

We use the term softkey to describe the row of 7 keys at the bottom of the CRT. The function of each softkey is displayed on the CRT directly above the softkey. Softkey names are shown in this manual between the { } symbols.

Softkeys are often described in this manual as being on a certain level, for example, softkey level 3. We use the level of the softkey to determine the location or necessary path to reach that particular softkey function. For example, to get to a softkey on level 3, you must press a specific softkey on level 1 followed by a specific softkey on level 2.

Specific softkeys for all levels change depending on the previous softkey pressed, with the exception of softkey level 1, which always remains the same. Softkey levels are all referenced from softkey level 1.

The softkeys on opposite ends of the softkey row have a specific use that remains standard throughout the different softkey levels. On the left is the exit softkey displayed with the up arrow {•} and on the right is the continue softkey displayed with the right arrow {•}.

- Use the exit softkey {•} on the far left to regress softkey levels. For example, if you are currently on softkey level 3 and you press the exit softkey, the softkeys change to the softkeys previously displayed on softkey level 2. When you press the exit softkey while holding down the [SHIFT] key, the softkey display returns to softkey level 1 regardless of the current softkey level.
- When more than 5 softkey functions are available on the same level, the control activates the continue {•} softkey at the far right of the softkey area. When you press the continue softkey, the softkey functions change to the next set of softkeys on that level.

The continue softkey is not available if there are 5 or fewer softkey functions on that level.

For example :

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
--	------------------	--------	----------------	-----------------	------------------	---

When softkey level 1 is reached, the previous set of softkeys is displayed. Press the continue softkey {• } to display the remaining softkey functions on softkey level 1.

(softkey level 1)

	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→
--	----------------	------------------	---------------	-----------------	--	---

On softkey level 1, the exit softkey is not displayed since the softkeys are already on softkey level 1.

The softkey functions for level 1 softkeys are explained in this appendix. Softkey functions for level 2 or higher are explained in the sections that apply to specific operations. A “tree” of softkeys listing all the softkeys and their levels is included in the back of this appendix.

Important: Some of the softkey functions are purchased as optional features. This manual assumes that all available optional features have been purchased for the machine. If the feature has not been purchased, blank keys may appear.

Describing Level 1 Softkeys (softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
	FRONT PANEL	ERROR MESSAGE	PASS- WORD	SWI TCH LANG		→

If you want to:	Press:
Edit, activate, or copy a program from a peripheral or control memory	{PRGRAM MANAGE}
Display or enter tool offset data, the work coordinate system offset data, etc.	{OFFSET}
View and modify the local and global parameter assignments for paramacros	{MACRO PARAM}
Check the part program, graphics, QuickCheck, and active program without actually moving an axis	{PRGRAM CHECK}
Enter and display inhibit zone limits, canned cycle parameter data, communications, AMP, etc.	{SYSTEM SUPORT}
Perform various operations that would normally be done by using an MTB panel, cycle start, jogging, etc.	{FRONT PANEL}
Display error messages, including an error log of old messages	{ERROR MESSAGE}
Enter or assign passwords and access levels to selected features	{PASSWORD}
Change the language displayed on the screen of the control	{SWITCH LANG}
Display more softkeys on the same level when there are more softkeys on a level than can be displayed at once	{• }
Display the previous level or previous row of softkeys	{• }

Using the Softkey Tree

The remainder of this appendix shows the softkey tree. This tree illustrates the entire softkey layout on the control in an easy-to-use flow-chart type format. This flow chart has been drawn to have no 4-way intersections (no 4 lines connected at any one point). If you see what appears to be a 4-way intersection, it is really only a crossover point for lines that do not intersect.

AXIS POSITION DISPLAY FORMAT SOFTKEYS

PRGRAM
A B S
TARGET
D T G
AXIS SELECT
M CODE STATUS
PRGRAM D T G
A L L
G CODE STATUS
SPLI T ON/OFF

NOTE: The first 4 softkeys (from PRGRAM to DTG) toggle between small and large screen display.

THE FUNCTION SELECT SOFTKEYS LEVEL 1

WITH POWER UP (AXIS POSITION) DISPLAY SCREEN

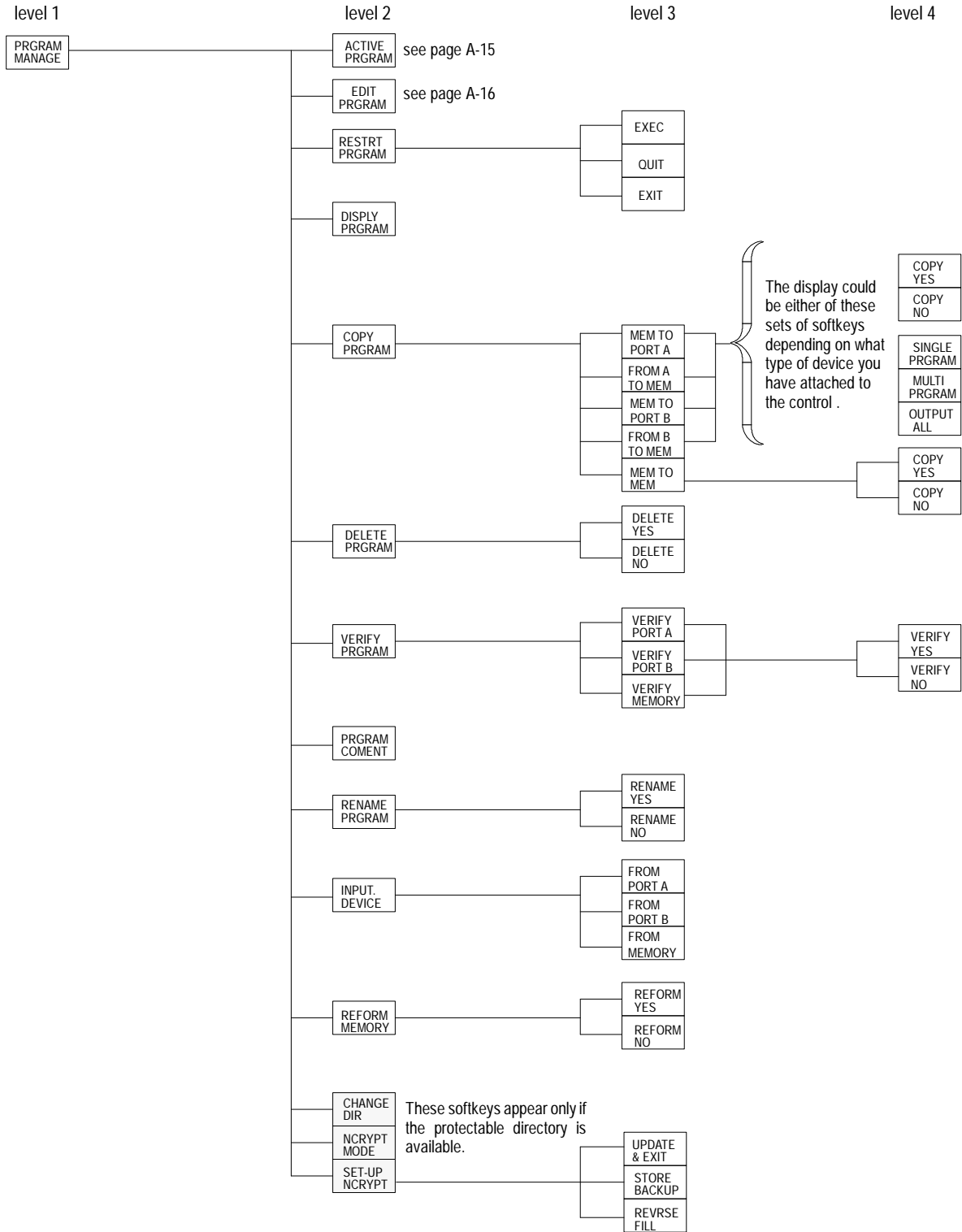
PRGRAM MANAGE	see page A-6
OFFSET	see page A-7
MACRO PARAM	see page A-9
PRGRAM CHECK	see page A-10
SYSTEM SUPORT	see page A-11

Optional PAL flag set to display "front panel"
when MTB is not part of the total CNC system

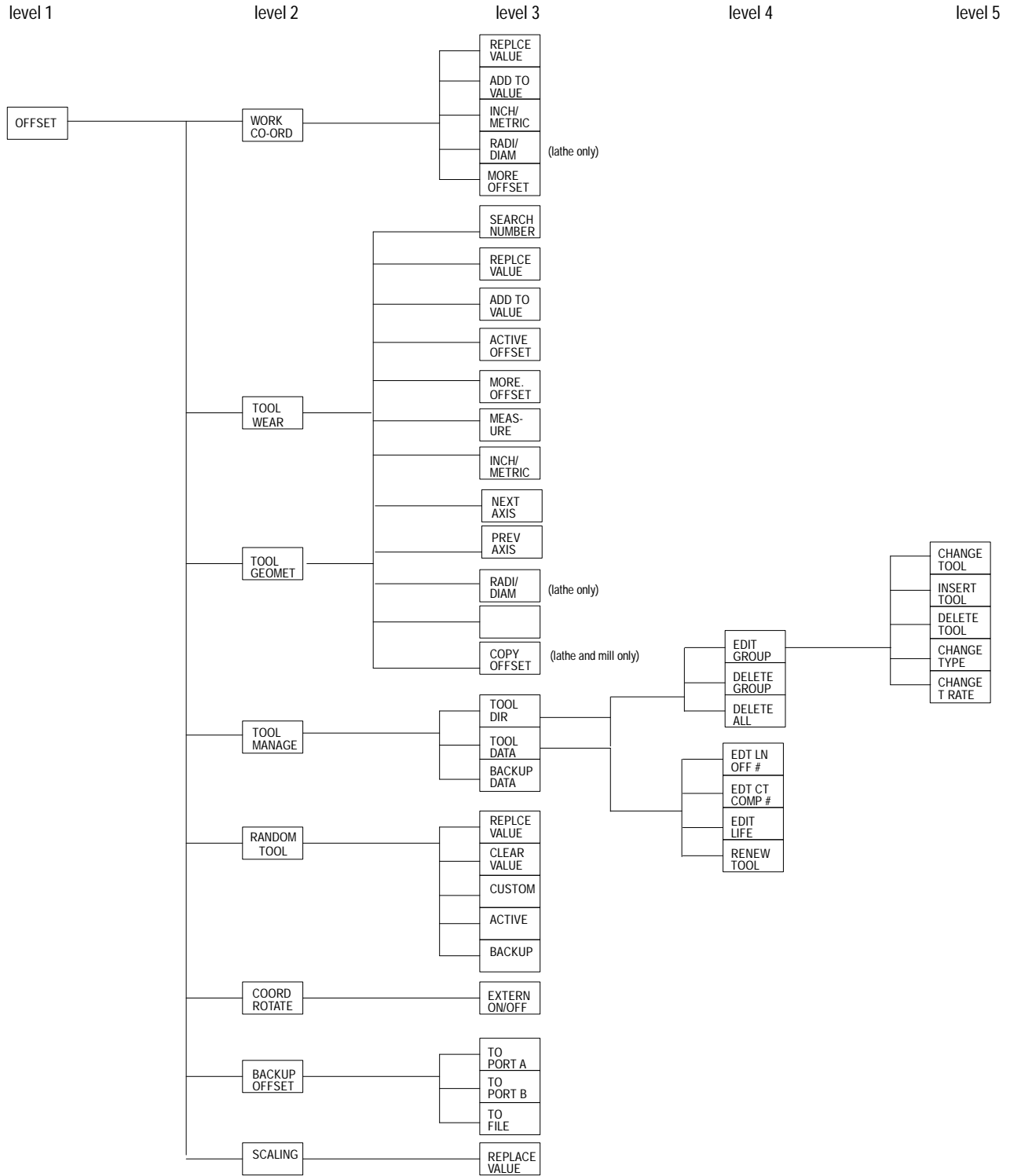
FRONT PANEL	see page A-13
ERROR MESSAGE	see page A-13
PASS- WORD	see page A-14
SWI TCH LANG	

PAL Display Page Option: Five softkeys available on third
screen. Five additional softkeys can be programmed on
[PROGRAM MANAGE], [SYSTEM SUPORT], and [OFFSET]
softkey levels. This feature allows for a total of 20 additional
softkeys to be used for PAL displays.

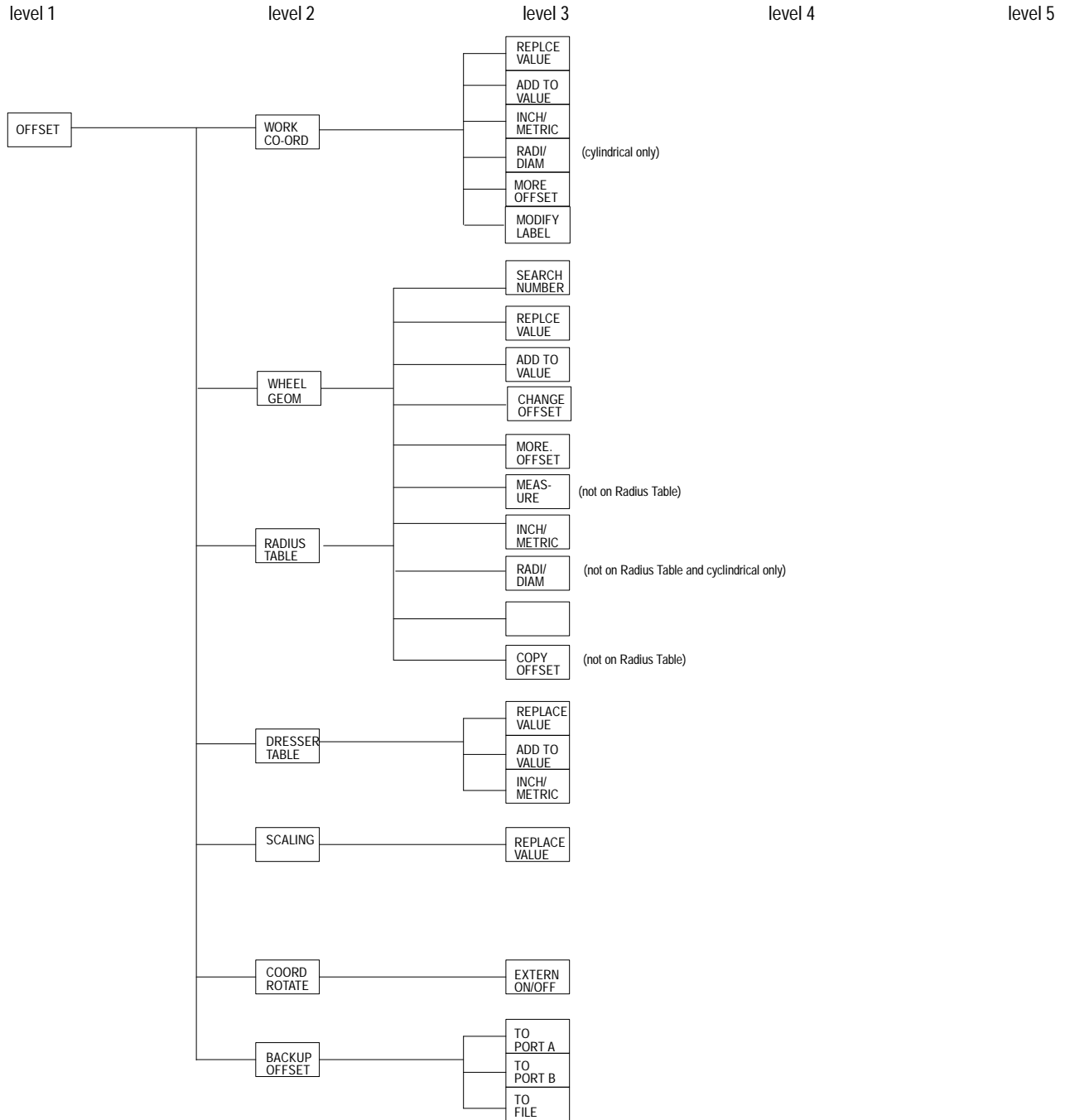
PRGRAM MANAGE



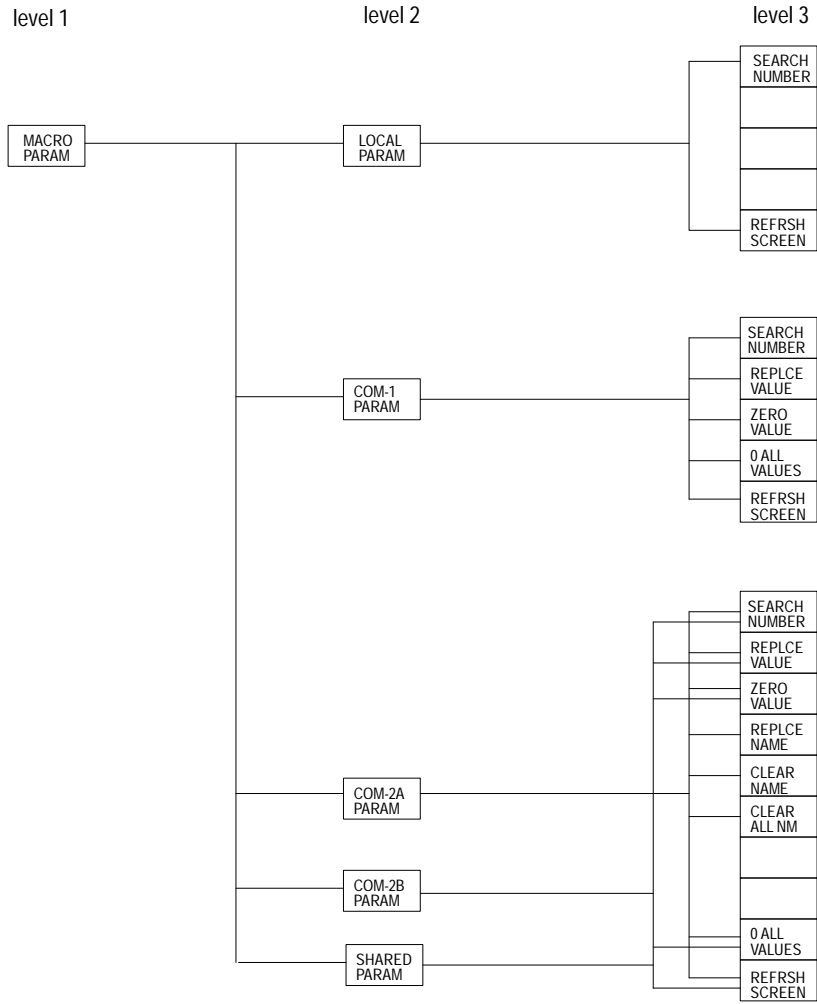
OFFSET (Lathe & Mill)



OFFSET (Grinder)

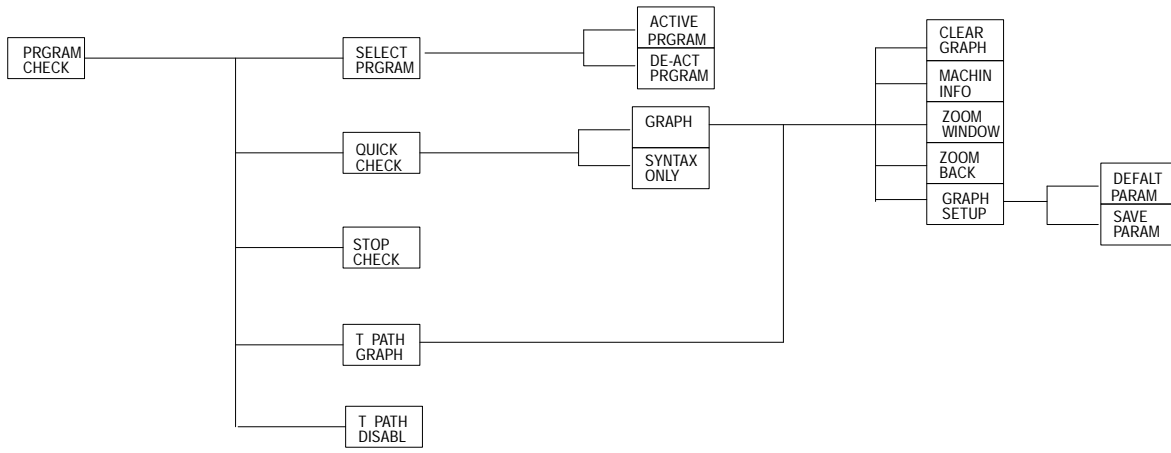


MACRO PARAM

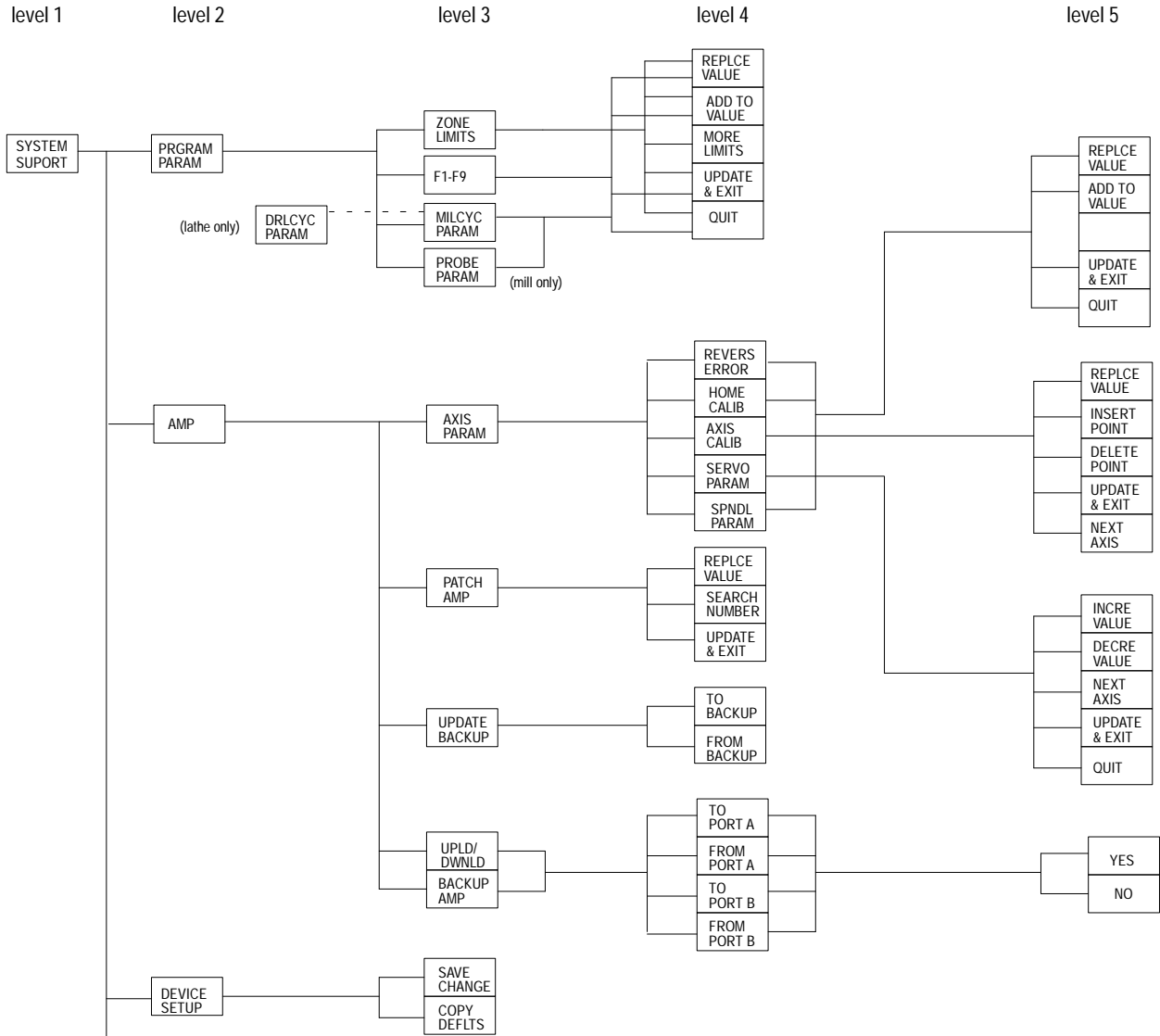


PRGRAM CHECK

level 1 level 2 level 3 level 4 level 5

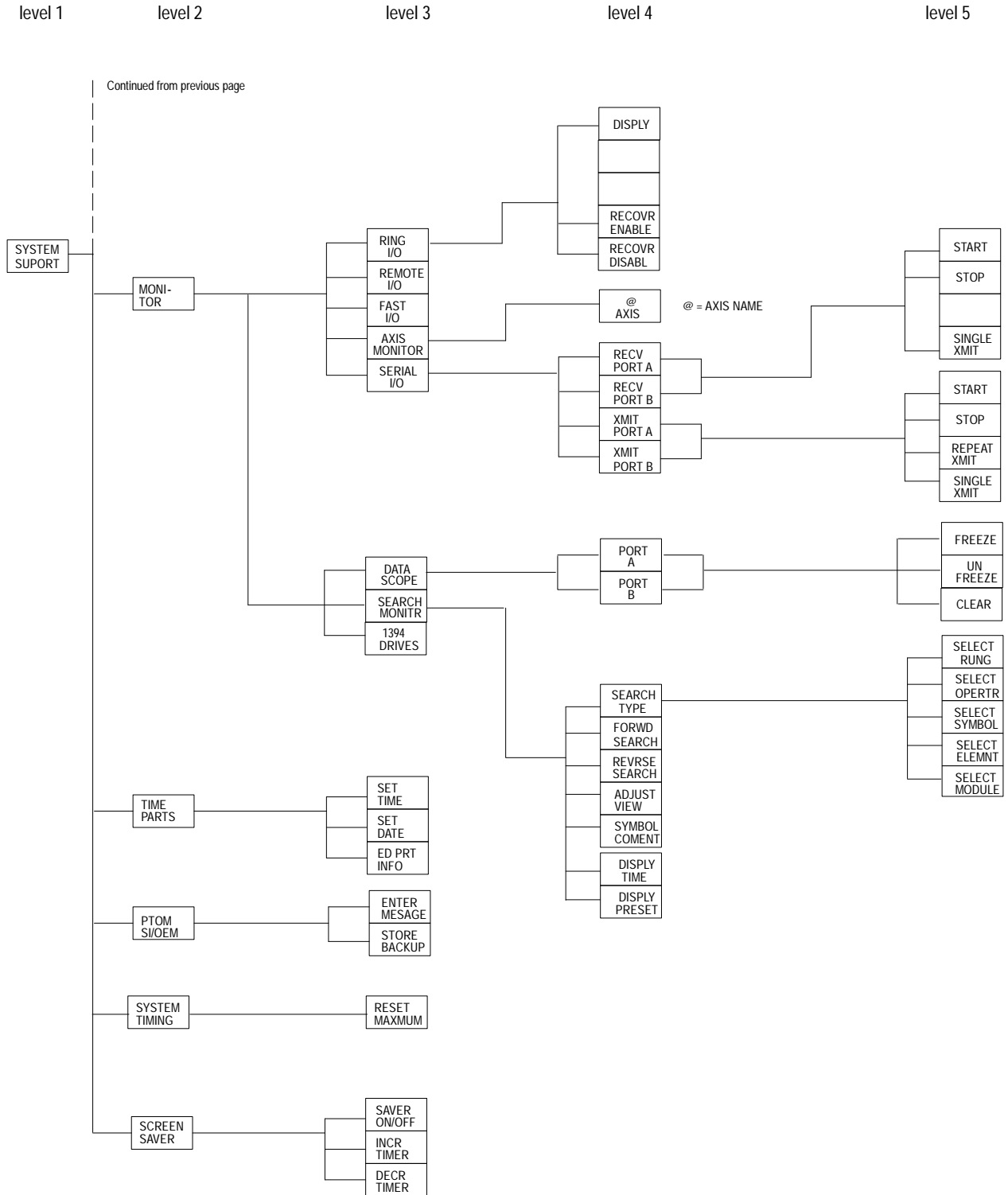


SYSTEM SUPPORT

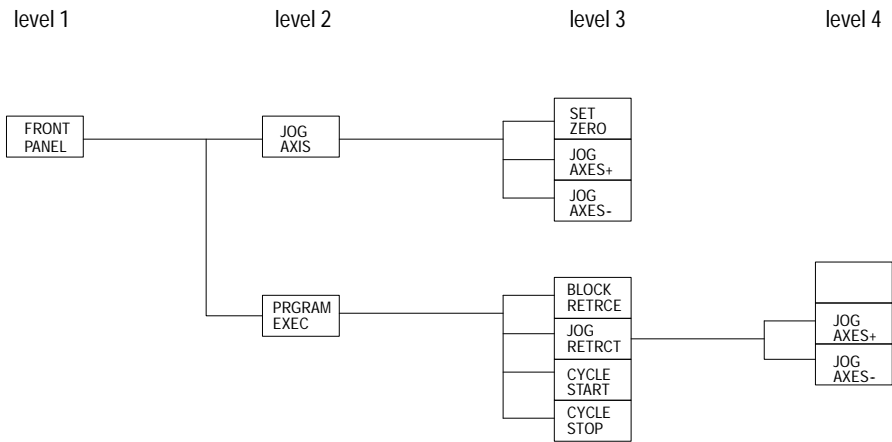


Continued on next page

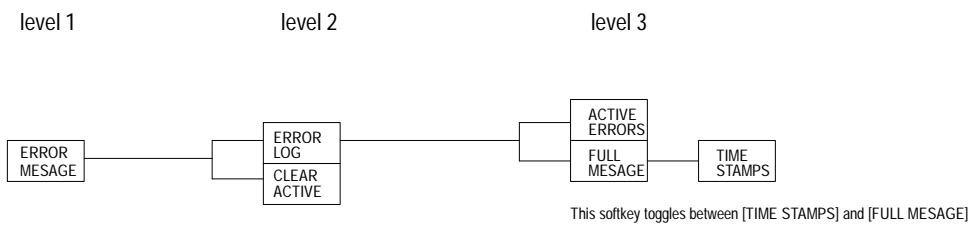
SYSTEM SUPPORT (continued)



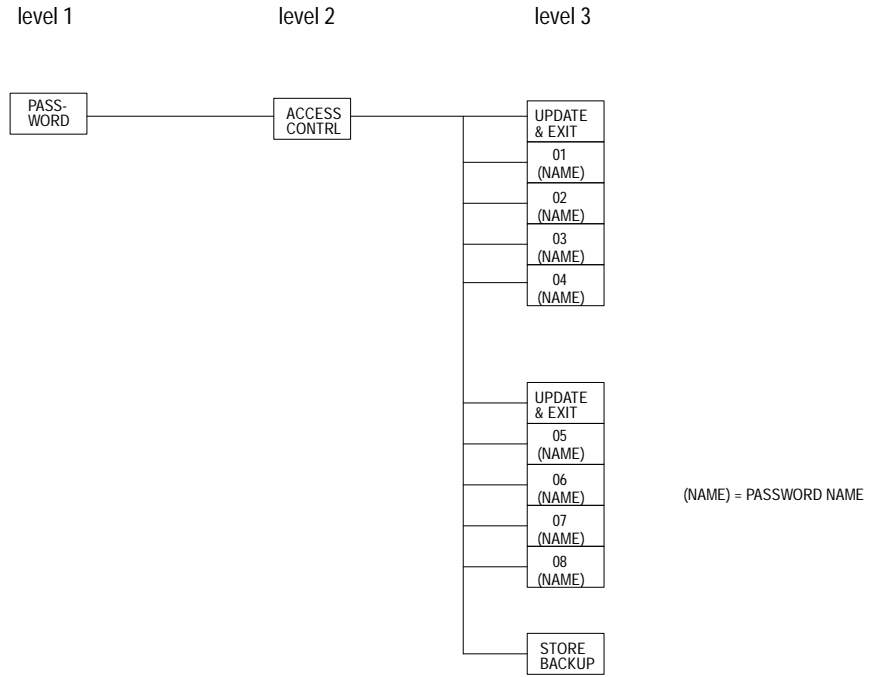
FRONT PANEL



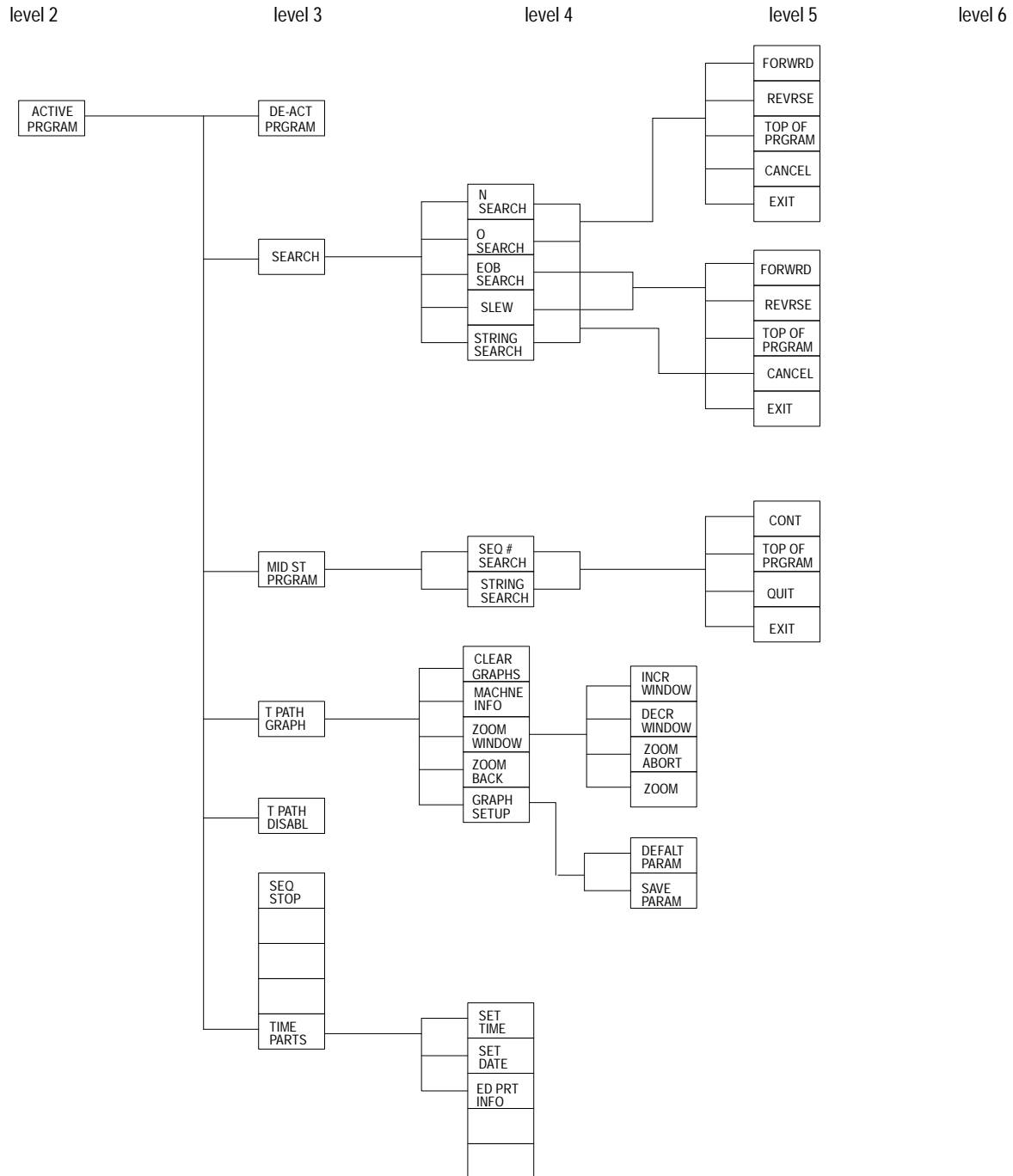
ERROR MESSAGE



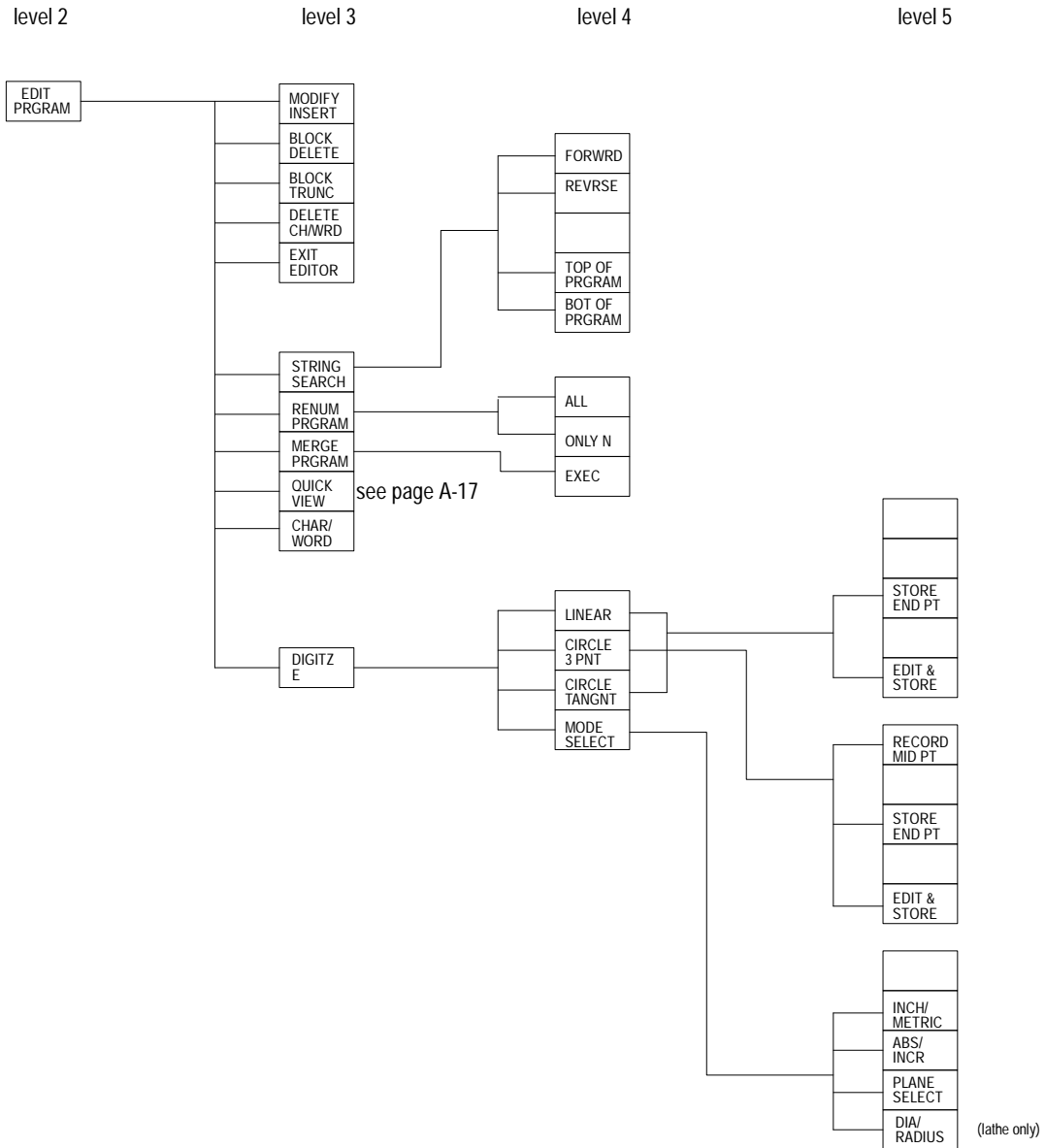
PASSWORD



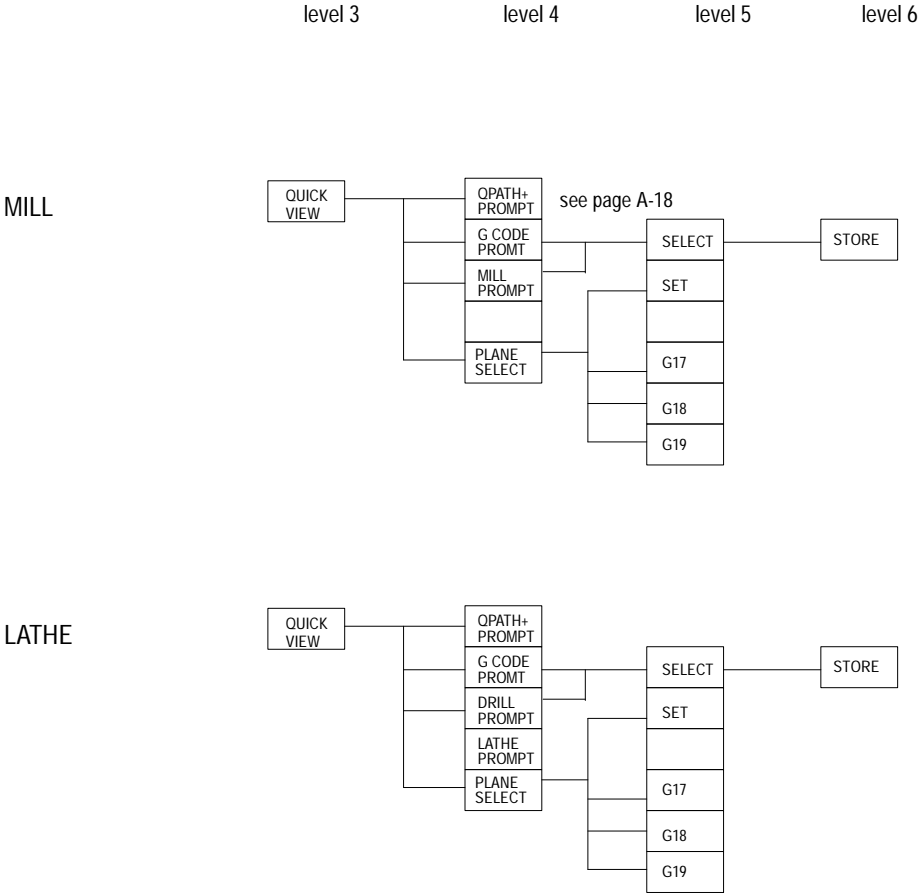
ACTIVE PROGRAM



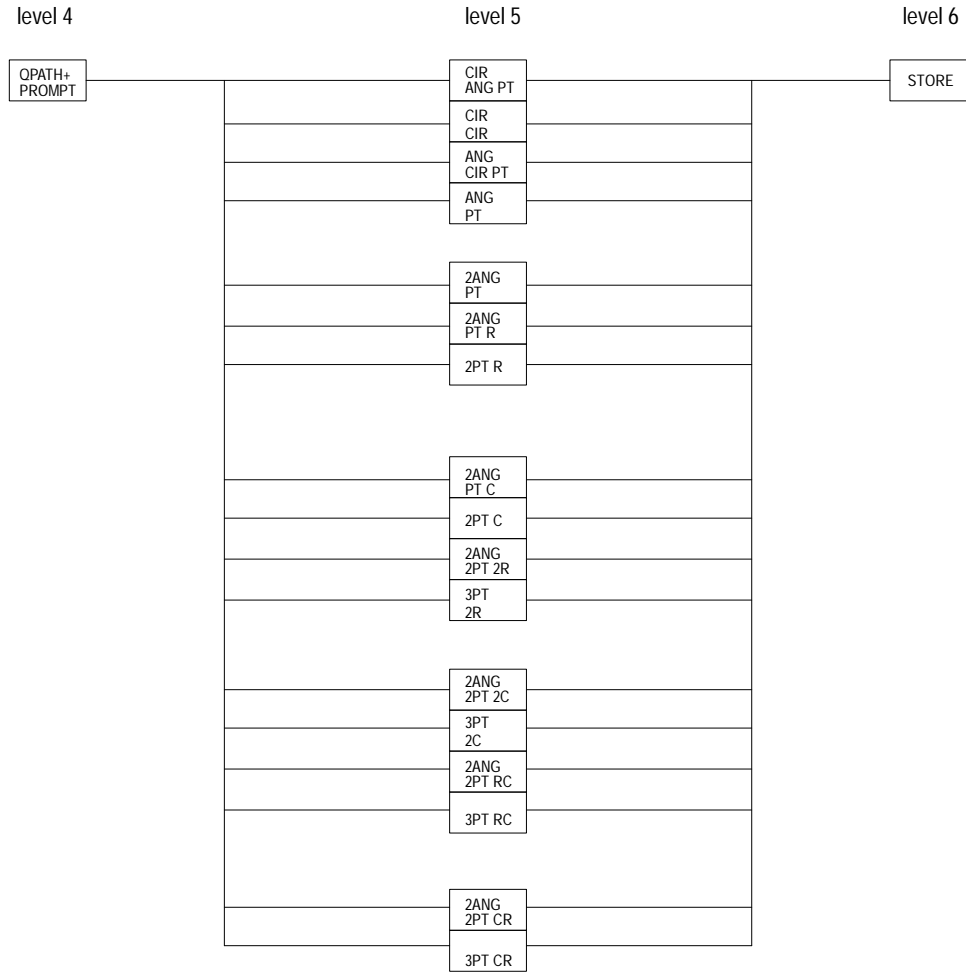
EDIT PRGRAM



QUICK VIEW



OPATH+ PROMPT



END OF APPENDIX

Error and System Messages

Overview

This appendix serves as a guide to error and system messages that can occur during programming and operation of the 9/Series control. We listed the messages in alphabetical order along with a brief description.

Important: To display both active and inactive messages, press the {ERROR MESSAGE} softkey found on softkey level 1. For details, see chapter 2.

Important: This appendix covers only error and system messages. PAL-generated operator messages generally appear on lines 21 and 22 of the CRT and should be described in documentation prepared by the machine tool builder.

Message	Description
Symbols	
(+) 15V SUPPLY FAILURE	+ 15 Vdc is not available at the main processor board. Check the wires from the main power supply and connector CN07 on the main processor board.
(-) 15V SUPPLY FAILURE	- 15 Vdc is not available at the main processor board. Check the wires from the main power supply and connector CN07 on the main processor board.
(+) OVERTRAVEL PROGRAM ERROR	If axis motion continues along the programmed path, the indicated axis will reach or exceed the positive software overtravel limit (runtime error).
(-) OVERTRAVEL PROGRAM ERROR	If axis motion continues along the programmed path, the indicated axis will reach or exceed the negative software overtravel limit (runtime error).
(+) OVERTRVL PRGRAM ERROR:	The end-point of the commanded move will cause the indicated axis to reach or exceed the positive software overtravel limit (pre-execution error).
(-) OVERTRVL PRGRAM ERROR:	The end-point of the commanded move will cause the indicated axis to reach or exceed the negative software overtravel limit (pre-execution error).
+/- SIGN ERROR	A + or - sign was found out of place when a numeric value was being decoded. Check the active program block for programming format errors.
1	
1394 AXIS MODULE MISMATCH	At power turn on the system identified an axis module in the 1394 rack that is misconfigured in AMP. If an extra axis module is present in the 1394 rack it should either be fully configured or not configured at all in AMP even if that axis module is not used or detached.
1394 RING COMMUNICATIONS ERROR	At power up the internal communications ring which runs through the front of the 1394 system and drive modules was either not connected, a device on the ring experienced a hardware failure, or a device on the ring was discovered to be misconfigured once a command was sent to the device. Make sure all axis modules and the end terminator are properly connected to complete the communication ring.
1746 RACK CARDS MISMATCH	The I/O configuration for the 1746 I/O rack that was downloaded from ODS, or resides in the PAL PROMs, contradicts what is actually in the rack (devices must match slot-for-slot).
1771 RACK CARDS MISMATCH	The I/O configuration for the 1771 I/O rack that was downloaded from ODS, or resides in the PAL PROMs, contradicts what is actually in the rack (devices must match slot-for-slot).

Message	Description
2	
2MB RAM IS BAD/MISSING	The control has discovered the RAM SIMMs for the two megabyte extended storage option are either damaged or missing. The RAM SIMMs must be installed or replaced. Contact your Allen Bradley sales representative for assistance.
9	
9/SERIES LATHE - CANNOT USE MILL AMP	The control was powered up with a lathe software option chip installed, when the AMP file that was downloaded was configured for a mill.
9/SERIES MILL - CANNOT USE LATHE AMP	The control was powered up with a mill software option chip installed, when the AMP file that was downloaded was configured for a lathe.
7300	
7300 NAMES TABLE IS CORRUPTED	7300 program name doesn't match corresponding name in cross-reference table.
7300 PATTERN NAME TOO LONG	More than 5 digits have been used in the pattern name.
A	
A RETRACE BUFFER WAS DELETED	The control required one (or more) of the block retrace buffers to perform a necessary block look-ahead operation (refer to block look-ahead in the user's manual). When this occurs, less block retrace operations can be performed than AMP is configured to allow. If this error occurs, to improve control efficiency, it is recommended that the number of allowable block retrace blocks set in AMP be lowered or add additional RAM to you system.
ABS POSITION NOT INITIALIZED	This message indicates that axes with absolute encoders have not been homed. These axes require an initial homing operation to establish the absolute position.
ABSOLUTE FEEDBACK FAILURE	The control has detected a loss of feedback from the absolute encoder. The most likely cause of this error would be a broken or disconnected wire. Axis homing may be required after the error condition is corrected.
ACC/DEC CONFIGURATION ERROR	An axis configuration error was detected by the control when manual acc/dec was requested in a program block.
ACCUM. AND EXPECTED LIFE ARE 0	No tool life data was entered for the current tool selected by the tool life management feature. Tool life management will be disabled for this tool.
ACTIVE GROUP CANNOT BE DELETED	An attempt was made to delete a tool group in the tool life management feature that contains an active tool currently in the tool holder.
ACTIVE OFFSET CANNOT CHANGE	An attempt was made to alter a tool offset value of a tool offset that is currently the active tool offset on the control. The active tool offset is indicated with an * on the tool offset table.
ACTIVE TOOL CANNOT BE CHANGED	An attempt was made to edit tool data for the currently active tool. De-activate the tool before editing.
ACTIVE TOOL CANNOT BE DELETED	An attempt was made to delete tool data for the currently active tool. De-activate the tool before editing.
ADAPTIVE FEED MIN LIMIT	This message indicates your are exceeding the programmed desired torque. The actual torque is greater than the desired programmed torque and the adaptive feed axis has reached the programmed minimum feed limit. Either raise the programmed desired torque or lower the minimum feed limit.
ADAPTIVE FEED PROGRAMMING ERROR	E and Q must both be programmed in every G25 block.
ALL DUAL AXES ARE PARKED	An attempt was made, while using dual axes, to move the dual group when all the axes of that group were parked.
AMP FILE SIZE ERROR	The size of the AMP file being downloaded is incorrect. The file cannot be downloaded.
AMP IN BACKUP DOES NOT MATCH AMP IN RAM	This message always appears after a successful AMP download if the downloaded file is different from the one currently stored in backup memory. Its purpose is to remind the user to copy the downloaded AMP into backup memory after testing it.

Message	Description
AMP WAS MODIFIED BY PATCH AMP UTILITY	This message always appears after changes have been made to AMP using the patch AMP utility. Its purpose is to remind the user that the current AMP has not been verified by a cross-reference check normally performed by ODS. It is meant as a safety warning.
AMPED HOLDING OR DETECT TRQ OUT OF RANGE	This message is displayed when you have entered a value in AMP for either the holding torque or the detection torque, for the feed to hard stop feature, that is higher than the value entered for the servos available peak torque. You must change your AMP values.
ANALOG SERVO VOLTAGE FAILURE	A $\pm 15V$ to the servo cards has failed.
ANGLE WORD NOT ALLOWED	An angle word was programmed in a QPP block where it is not allowed, for example, programming an angle word in a circular QPP block.
ANGLED WHEEL AXES, JOG ONE AT A TIME	While in the angled wheel grinding mode you can not jog more than one axis in the angled wheel plane at any one time.
ANGLED WHEEL CONFIG ERROR	The angled-wheel grinder AMP downloaded to the control is not configured correctly. Make sure all necessary angled-wheel parameters are configured correctly and re-download AMP to the control.
ANGLED WHEEL NOT CONFIGURED	You have attempted to program an angled wheel grinder mode function and the angled wheel feature has not been correctly configured for your system. The angled wheel feature must be configured in AMP and is a purchased option for your 9/Series control.
ARCTAN SYNTAX ERROR	An attempt was made to calculate or execute a paramacro block that calculates the arc tangent of an invalid or improperly entered number.
ARITHMETIC OVERFLOW ERROR	An internal math error has occurred; contact Allen-Bradley customer support service.
ARITHMETIC UNDERFLOW ERROR	An internal math error has occurred; contact Allen-Bradley customer support service.
AUX FB NOT ALLOWED WITH DEPTH PROBE	Your AMP file has a depth probe configured for an axis that also is configured to use an optional feedback device. A depth probe can not be configured to use any feedback device other than its depth probe for that depth probe axis. If a second feedback device is used it is configured in AMP as a separate logical axis.
AUXILIARY FEEDBACK DISCONNECTED	The digital servo module provides the capability to use two different feedback encoders with one servo (in the case where two encoders are used, the auxiliary encoder is used for the position feedback). If the servo processor detects that the auxiliary encoder has been disconnected, this message is displayed.
AUXILIARY FEEDBACK QUADRATURE FAULT	The digital servo module provides the capability to use two different feedback encoders with one servo (in the case where two encoders are used, the auxiliary encoder is used for the position feedback). If the servo processor detects a quadrature fault on the auxiliary encoder, this message is displayed.
AUXILIARY SPINDLE 2 NOT CONFIGURED	For aux spindle 2 to be programmable, it must be configured in AMP; a decode error.
AUXILIARY SPINDLE 3 NOT AVAILABLE	AMP configuration error; aux spindle 3 can be configured only on a 9/290.
AUXILIARY SPINDLE 3 NOT CONFIGURED	For aux spindle 3 to be programmable, it must be configured in AMP; a decode error.
AXES COLLISION	Two processes have collided. Interference checking has stopped all motion.
AXES CONFIGURED ON INACTIVE PROCESS	An AMP was loaded that contains an axis that was configured for an inactive process. Set the process axis in AMP to a process that has been configured.
AXES DATA MISSING	Expected axis data is missing in a program block.
AXIS AMPED AS NON-SCALING AXIS	The user attempted to scale an axis that was AMPed as non-scaleable.
AXIS ASSIGNED TO PAL AXIS MOVER	The user attempted to move the axis configured as the PAL axis mover axis by some means other than PAL.
AXIS DISPLAY DISABLED BY PAL	The position display for a selected axis has been turned off using the \$NODP flag.
AXIS IN PLANE DOES NOT EXIST	At least one of the axes assigned to a plane that was defined in AMP does not exist. An example of when this error would occur is if an axis was renamed in AMP, but that new name was not entered into the AMP plane definition. Another example would be if an unfitted axis was assigned to that plane.

Message	Description
AXIS INVALID FOR G24/G25	The programmed axis was not AMPed for software velocity loop operation, and can not be used in a G24 or G25 block. To use these features the axis programmed must be configured for tachless operation (or be a digital servo).
AXIS IS HARD STOPPED, CANT ADJUST SERVO	The torque limit of the servo can not be adjusted because, either the axis is in a hard-stopped state, or some other axis on the same servo card is in a hard-stopped state.
AXIS MODULE POWER FAULT	The current through the power output transistors is monitored. If the current exceeds a fixed level (greater than 300% of controller rating) this fault will appear. Typical causes are a shorted lead, motor malfunction, or malfunctioning power IGBTs.
AXIS MODULE OVER CURRENT	One of the axis modules of your 1394 drive has been requested to provide too much current. This is typically caused by Accel/Decel command from the CNC is requiring peak current for an excessive amount of time, the machine friction or inertial/viscous load is excessive, the motor has been improperly sized, a short circuit exists across the drive output terminals, logic supply circuits have malfunctioned, or AC input is incorrectly wired.
AXIS MODULE BUSS VOLTAGE LOSS	The DC bus supply was lost to the axis module. Check slider connections/termination strip or there could be a blown link fuse.
AXIS MODULE OVER TEMP	The 1394 contains a thermal sensor which senses the internal ambient temperature. Causes could be: that the cabinet ambient temperature is above rating. The machine duty cycle requires an RMS current exceeding the continuous rating of the controller. The airflow access to the 1394 is limited or blocked. This does not necessarily indicate a motor over temperature. Motor over temperture sensors should be wired directly into the E-Stop string.
AXIS MOVER CONFLICT WITH G16.3/G16.4	You have requested a PAL axis mover function on an angled wheel grinder. You can not use the PAL axis mover in one of the angled wheel modes unless the PAL axis mover has control of both the axial and the wheel axes.
AXIS NAME DUPLICATE	Two or more axes have been assigned the same name in AMP.
AXIS NOT IN PROCESS	You attempted to read/write a paramacro parameter for an axis that is not currently in the process requesting the data. To access paramacro parameter data for an axis, that axis must be in the process making the request.
AXIS POSITION INCORRECT	Using the mid-start program function, you have searched to a block that does not create the programmed contour if started from your current axis position. Be aware the mid-start operation may have searched thru a offset operation that is not readily apparent to determine your axis position. The mid-start operation is aborted. You must re-perform the mid-start operation and either position the axes to the correct axis position, or use the {MOVE & EXIT} softkey to find the correct axis position.
AXIS SELECT NOT ALLOWED	The {AXIS SELECT} softkey was pressed when no axis select option is available. Axis select is only available on large screens and normal character size screen for systems that contain more then 9 axes total or dual process systems with more than 8 axes in a process. It is not available when the small screen (showing all system AMPed axes) is being viewed.
AXIS TYPE-POSITION LOOP ERROR	In patch AMP, an axis was assigned a position loop type that is illegal for the axis type assigned to that axis.
B	
BACKUP VERSION OF AMP WAS COPIED TO RAM	The AMP in RAM was erased (battery backup failed) or corrupted, so the control automatically copied the version of AMP in backup memory into RAM memory. (The control stores AMP in backup, but works from the copy of AMP in RAM memory.)
BAD DAC MONITOR PATCH AMP ENTRY	An invalid value was entered into patch AMP parameter #86 or #87. Either parameter allows the axes to be monitored through the servo module (DAC) analog output. See documentation provided by Allen-Bradley on patch AMP, or contact Allen-Bradley customer support service.
BAD FIRST POCKET BLOCK	When performing an irregular pocket cycle, the first pocket block should be away from the pocket start/end corner, not toward it. The move to the start/end corner is generated based on the coordinates programmed in the pocket definition block itself.
BAD PAL PROM	One of the PAL PROM chips (plugged into the main processor board) has failed or is not plugged in properly.

Message	Description
BAD RAM DISC SECTOR CHECKSUM ERROR	A RAM disk sector error was detected during the RAM checksum test at power-up. Attempt to power-up again. If the error remains, contact Allen-Bradley customer support services.
BAD RECORD IN PROGRAM	This indicates a serious problem with the program. Attempt to open the program a second time. If retry doesn't work, you may have to delete the program. Typically this error is not caused by a programmer or operator action. It is typically caused by an internal software error in the program.
BAD STATE/TOKEN COMBINATION (PROGRAM ERROR)	While attempting to decode the current block, a combination of characters caused a decode error to occur. Check the characters in the current block for an illegal combination.
BATTERY FAILURE	The battery that provides backup of the RAM memory is not functioning; the voltage may be low. The battery may be dead, removed, or poorly connected.
BLK DELETE CHG IGNORED ON PREPARED BLKS	A block-delete was activated while a program was executing. This change is ignored by the control for blocks that have already been read into the control's set-up buffer (see block look-ahead in user's manual).
BLOCK LENGTH ERROR	A block that exceeds the allowable maximum block length was programmed.
BLOCK RETRACE ABORTED	The block retrace operation being performed has been canceled. When <CYCLE START> is pressed, the control will return the tool along a linear path back to the start-point of the block retrace operation.
BOOT DIRECTORY IS MISSING	The update utility failed to properly create the system boot directory. Retry the update. If the error occurs again, contact your local Allen Bradley service.
BOOTSTRAP FAILED TO START	The bootstrap code did not send the "bk" signal to the main processor within the specified time.
BOTH AXES IN QPP PLANE NOT PRGMD	The second block of a currently executing QuickPath Plus two-block set does not contain both required axis words in the current plane. Both axis words are required to correctly identify the end-point of the second move.
BOTH LINES ARE PARALLEL	Both blocks of a two-block QPP sequence are parallel, and no mathematical intersection can be computed.
BOTH PORTS ARE BUSY	An attempt was made to use or monitor communication ports A or B when neither were available.
BUSY, REQUEST IGNORED	You have requested an operation while the control is currently executing some other higher priority function. The control must first complete the higher priority task before your new task can be performed.
C	
CALLED 7300 PATTERN NAME IS BAD	The 7300 pattern name that is called by a part program does not exist.
CANCEL/REMOVE OFFSET BEFORE AXIS CHANGE	You have attempted to change the active tool length axis while a length offset is currently active on that axis. You must cancel tool length offsets before you are allowed to change the active tool length axis.
CANNOT (GOTO) TO INSIDE A (DO)	A (GOTO) command cannot transfer execution to a block which is located within a (DO) loop.
CANNOT ACCESS REMOTE VARIABLE	Variable name is invalid. Check the fields for CNC name and remote name, and make sure they are in the correct format.
CANNOT ACTIVATE - OPEN PROGRAM	An attempt was made to activate a program for execution when it was still open for an editing operation. Before it can be activated for automatic execution, it is necessary to press the {EXIT EDITOR} softkey from the edit menu to close a program being edited.
CANNOT ACTIVATE RAM PARTITION	The RAM disk has been corrupted. Attempt to perform a "REFORMAT" operation. If this is unsuccessful, consult Allen-Bradley customer support services.
CANNOT ASSIGN IN CURRENT MODE	An attempt was made to modify a paramacro parameter that cannot be modified when the cutter compensation or TTRC feature is active.
CANNOT CALCULATE - PROMPT PRESENT	An attempt to perform a calculate operation was made when some other prompt was present on line 2 of the CRT. Before the control will allow a calculation to be made, it is necessary to remove any prompts from line 2.

Message	Description
CANNOT COPY	The requested copying task cannot be performed due to an internal problem in the file or RAM disk. Contact Allen-Bradley customer support service.
CANNOT DELETE - OPEN PROGRAM	The selected program is either active or open for editing and cannot be deleted.
CANNOT DELETE ALL PROGRAMS	An attempt was made to delete all part programs or to reformat RAM while a program was being edited or was currently selected as the active program for execution.
CANNOT DELETE PROGRAM	The file selected cannot be deleted. This is caused by a major error being detected in the actual software file of the program. It may be necessary to "REFORMAT" RAM to remove the program. If this is unsuccessful, contact Allen-Bradley customer support service.
CANNOT DIVIDE BY ZERO	An attempt was made to divide a quantity by zero, either using the CALC functions or in an executing program with a paramacro operator.
CANNOT EDIT - FILE UPLOADING	The file you've tried to open is already open and is in the middle of a part program upload or download operation with ODS.
CANNOT EDIT - MUST BE IN CYCLE OR E-STOP	An attempt was made to edit a part program while another part program was currently being executed.
CANNOT EDIT - OPEN PROGRAM	The program that you have selected for editing is currently open for another feature.
CANNOT EDIT - OTHER FILE IS BEING EDITED	An attempt was made to edit a part program while another part program was currently being edited.
CANNOT EDIT ACTIVE PROGRAM	An attempt was made to edit a program that is currently selected as the active program for execution. Before it can be edited, the program must first be disabled.
CANNOT EXIT IN CYCLE	You cannot exit in the middle of a roughing cycle because it executes at runtime, not during setup.
CANNOT FIND CORRECT POSITION	The program-restart feature cannot locate the correct program block in the program at which automatic execution was interrupted. To position the program at the correct block, it will be necessary to perform one of the other search operations. The operator must know what this correct block is as the control has failed its recover operation.
CANNOT FIND PAL PAGE	PAL requested a PAL display page to be displayed that does not exist in the display page file.
CANNOT FORMAT - OPEN PROGRAM	A program was selected for automatic execution or was still in the edit mode when a request to format memory was made. The active program must be disabled by pressing the {CANCEL PROGRAM} softkey, and any program being edited must be closed by pressing the {EXIT EDITOR} softkey before formatting memory.
CANNOT FORMAT RAM PARTITION	The control is unable to format memory due to open file conditions indicating a more serious problem. Consult Allen-Bradley customer support services.
CANNOT JOG - ALL AXES ARE PARKED	An attempt was made to jog a dual group when all the axes were parked.
CANNOT MERGE WITH SAME PROGRAM	An attempt was made to merge the same program that is being edited with itself. If this is desirable, first copy the original program, then merge the copy into the original.
CANNOT OPEN DIRECTORY	This indicates a serious RAM disk problem. If retry doesn't work, you may have to reformat.
CANNOT OPEN PROGRAM FOR READ	This indicates a serious problem with the program. If retry doesn't work, you may have to delete the program.
CANNOT OPEN PROGRAM FOR WRITE	An error occurred while attempting to open a file on the RAM disk. Either the RAM disk is full, or there is an internal problem with the file. The file may need to be deleted.
CANNOT OPEN SUBPROGRAM	An attempt to call a sub-program has failed. This is usually caused by the sub-program name (programmed in the calling block with a P-word) not existing in the current program directory.
CANNOT READ A WRITE-ONLY PARAMETER	An attempt was made to use the value of a paramacro system parameter that is a write-only parameter. This parameter may have only its value written to. It cannot be read.
CANNOT READ DIRECTORY	This indicates a serious RAM disk problem. If retry doesn't work, you may have to reformat.
CANNOT READ PROGRAM	This indicates a serious problem with the program. If retry doesn't work, you may have to delete the program.

Message	Description
CANNOT RENAME	When performing a rename of a program name, the new program name has not been correctly entered. The format is OLD PROGRAM NAME,NEW PROGRAM NAME.
CANNOT REPLACE START POINT	An illegal attempt was made to change the axis calibration start-point using the online AMP feature.
CANNOT RESTART G24 HARD STOP	An attempt was made to restart a part program on a block which would have an axis at the hard stop. You cannot restart or mid start a part program after if (at that blocks execution) any axis would be holding against a hard stop. You must either re-start/mid-start to a block before the G24 hard stop block or to a block after the hard stop is released.
CANNOT SEND AVAILABLE COMMAND	This is displayed when a non-programmed communications command is executed from 'send' softkey.
CANNOT SET DATA WHEN TOOL IS ACTIVE	An attempt was made to manually (using the softkeys) change tool management data for the currently active tool. Tool management data can be changed only for a tool that is not currently selected as the active tool.
CANNOT TAP IN CSS	You must disable the CSS feature before you begin a tapping operation. Disable CSS using a G97 command.
CANNOT TAP IN VIRTUAL-C MODE	You attempted to use the solid tapping feature while cylindrical or end-face milling was active.
CANNOT UPLOAD - PAL NOT IN PROM	PAL can be uploaded only from the PAL PROMs. PAL in RAM memory cannot be uploaded.
CANNOT UPLOAD - PAL SOURCE NOT LOADED	When the source is loaded, PAL can be uploaded in the 9/240 only . The 9/260 and 9/290 always have PAL in flash.
CANNOT USE COPY WITH ACTIVE TOOL OFFSET	An attempt was made to copy offset data from one axis to another using the {COPY OFFSET} softkey. You cannot use this softkey if the tool offsets are active.
CANNOT USE EXIT - BLOCK NOT FOUND	An attempt was made to {EXIT} while searching for a block for a mid-program start. You cannot use {EXIT} until the block has been found. To abort the search, use {QUIT}.
CANNOT WRITE A READ-ONLY PARAMETER	An attempt was made to assign a value to a PAL or system paramacro parameter that is a read-only parameter. The value of these parameters can be used only by the programmer; they cannot be altered in the program.
CANNOT WRITE TO PROGRAM	This indicates a serious problem with the program. Attempt to write to program a second time. If retry doesn't work, you may have to delete the program. Typically this error is not caused by a programmer or operator action, but rather by an internal program software error.
CAUTION! YOU ARE IN 7300 TAPE MODE TO RETURN TO STANDARD 9/240 MODE RESET THE 7300-COMPATIBILITY PAL FLAG	The operator is cautioned that the tape being copied is presumed to be a 7300 formatted tape. This message is displayed on the copy-tape set-up screen when the MCU is in 7300 compatibility mode.
CC/TTRC ON, CAN'T ASSIGN TIME DEP. PARAM	An attempt was made to assign a time-dependent paramacro system parameter while dresser/wheel radius compensation was active. Time-dependant parameters are any system parameters that record or reference a current axis position.
CHAMFER LENGTH/RADIUS TOO LARGE	A chamfer or radius value programmed with a ,C or ,R would generate a chamfer or radius that is larger than one or both of the two adjacent tool paths.
CHAMFER/RADIUS NOT ALLOWED	An attempt was made to perform a chamfer or radius cut (programmed with a ,R or ,C) in a block that does not allow these functions to be performed. For example, you cannot do a chamfer or radius cut in a non-motion block, in the last block on an MDI line, or in the last block of a part program.
CHANGE NOT MADE IN BUFFERED BLOCKS	Changes to the offset table did not affect those program blocks that were already in the control's current activation queue. Program blocks that call for offsets and which follow those already in the activation queue will call the updated offset tables.
CHANNEL NAME TOO LONG	There is an error in G05 DH+ communications block.
CHAR MUST BE _ , ., LETTER, DIGIT	You have used incorrect search string syntax in the PAL search monitor utility.
CHAR MUST BE LETTER,DIGIT, UNDERSCORE	You have used incorrect search string syntax in the PAL search monitor utility.
CHARACTERS MUST BE DIGIT	You have used incorrect search string syntax in the PAL search monitor utility.

Message	Description
CHARACTERS MUST FOLLOW WILDCARD	You have used incorrect search string syntax in the PAL search monitor utility.
CHECKSUM ERROR IN FILE	The file (AMP, PAL) being downloaded from a storage device has a checksum error. The file cannot be used.
CIRCLE MID-POINT NOT ENTERED	The center-point of an arc is not entered in a circular programming block. Circular blocks require programming either an R or an I, J, K in the block.
CIRCULAR BLOCK NOT ALLOWED	When activating cutter compensation, you cannot program a circular block as the first block or as the last block prior to deactivating cutter compensation.
CIRCULAR NOT ALLOWED AFTER SKIP	A circular move cannot immediately follow a G31 or G37 series skip block. Only linear moves are permitted as the next move following a G31 or G37 type code.
CIRCULAR PROGRAMMING ERROR	A circular motion was programmed incorrectly. Typically this occurs from incorrectly programming an R or I, J or K value.
CODING ERROR	A system software error has occurred. Consult Allen-Bradley customer support services.
COM COMMAND TABLE IS CORRUPTED	Restore the flash version of the output command table.
COM CONFIGURATION TABLE IS CORRUPTED	Restore the flash version of the communication configuration table.
COMM ERROR WHILE PROCESSING HOST REQUEST	A communication error occurred between your PC and 9/Series while performing an update utility. Retry at a lower baud rate. If that does not work check communication ports, connections and cable wiring.
COMMUNICATION TIME-OUT	The time allowed for a peripheral device to respond has elapsed. Check cable connections and device set-up.
COMMUNICATIONS DISPLAY PAGE ENABLED	When a remote host enables the 9/Series remote operator display screen, this message is displayed.
COMMUNICATIONS LINK IS DOWN	A problem was detected in the communications line. Check the cables and retry the download/upload.
COMPLETED WITH ERROR(S)	A QuickCheck syntax check operation has completed the check of the currently active program and found one or more errors. Some editing of the program is required.
COMPLETED WITH NO ERRORS	A QuickCheck syntax check operation has completed the check of the currently active program and found no syntax errors.
CONFIGURATION EXCEEDS AVAIL MEMORY	This error occurs when the amount of available control memory drops below what is required to maintain a minimum 5 block setup buffer for program execution. The system is held in E-Stop when this error occurs. You may either chose to add more memory to your system or re-configure your system by decreasing the watch list allocation (in AMP) for OCI systems.
CONTINUE NOT ALLOWED	An attempt was made to continue a program search when no character string was entered. This can occur when an error is generated by the program being searched and the control cannot continue the search of the program correctly.
CONTROL RESET NOT ALLOWED	The Control Reset Request was not honored by the control (e.g., a Control Reset Request during Cycle Suspended state).
CORRUPTED PROGRAM FOUND & DELETED	Program was found to be corrupted and not usable. This program was deleted.
CPU #2 DUALPORT RAM FAILED	The DUALPORT RAM memory shared between the 68000 main processor and the Z80 I/O ring processor has failed. (two 98030's instead of the 68000 and Z80 on 9/230, 9/260, and 9/290 controls)
CPU #2 EXEC IS BAD/MISSING	CPU #2 exec is not in flash; you must use update utility to load it (9/290 only). Consult Allen-Bradley customer support services.
CPU #2 EXEC WILL NOT START	CPU #2 is halted and will not start to execute its exec (9/290 only). Consult Allen-Bradley customer support services.
CPU #2 HARDWARE ERROR #2	The 68030 main processor has detected a bus error. Consult Allen-Bradley customer support services (9/290 only).
CPU #2 HARDWARE ERROR #3	The 68030 main processor has detected a spurious interrupt. Consult Allen-Bradley customer support services (9/290 only).

Message	Description
CPU #2 HARDWARE ERROR #4	The 68030 main processor has detected an illegal address. Consult Allen-Bradley customer support services (9/290 only).
CPU #2 HARDWARE ERROR #6	The 68030 main processor has detected a privilege violation. Consult Allen-Bradley customer support services (9/290 only).
CPU #2 HARDWARE ERROR #8	CPU #2 has detected an unassigned vector interrupt. Consult Allen-Bradley customer support services (9/290 only).
CPU #2 HARDWARE ERROR #9	CPU #2 has detected an illegal instruction. Consult Allen-Bradley Customer Support Services (9/290 only).
CPU #2 LOCAL RAM FAILED	The RAM memory supporting the 68030 I/O ring processor has failed (Z80 I/O ring processor on 9/240 only). Consult Allen-Bradley customer support services.
CPU #2 PROM HAS FAILED	The PROM memory supporting the 68030 (Z80 9/240 only) I/O ring processor has failed its checksum test. Consult Allen-Bradley customer support services.
CPU #2 RAM HAS FAILED	The RAM memory supporting the 68030 (Z80 9/240 only) I/O ring processor has failed. Consult Allen-Bradley customer support services.
CPU #2 WATCHDOG ERROR	The 68030 (Z80 9/240 only) I/O ring processor has failed. Consult Allen-Bradley customer support services.
CREATING BACKUP FILE - PLEASE WAIT	A backup file for the current utility is being created. The message will clear when the backup is complete.
CREATING TOOL OFFSET FILE - PLEASE WAIT	The tool offset table (or tables) is currently being backed-up. The control is generating an executable G10 program and entering it into the control's program directory.
CREATING TOOL MGMT. FILE - PLEASE WAIT	The tool management tables are currently being backed-up. The control is generating an executable G10 program and entering it into the control's program directory.
CSS RPM LIMIT AUXILIARY SPINDLE 2	The aux spindle 2 RPM requested by CSS is greater than the maximum CSS RPM limit. This limit is set by the system installer in AMP or can be reduced by programming a G92 block.
CSS RPM LIMIT AUXILIARY SPINDLE 3	The aux spindle 3 RPM requested by CSS is greater than the maximum CSS RPM limit. This limit is set by the system installer in AMP or can be reduced by programming a G92 block.
CSS RPM LIMIT FIRST SPINDLE	The spindle 1 RPM requested by CSS is greater than the maximum CSS RPM limit. This limit is set by the system installer in AMP or can be reduced by programming a G92 block.
CSS RPM LIMIT PRIMARY SPINDLE	The primary spindle RPM requested by CSS is greater than the maximum CSS RPM limit. This limit is set by the system installer in AMP or can be reduced by programming a G92 block.
CSS RPM LIMIT SECOND SPINDLE	The spindle 2 RPM requested by CSS is greater than the maximum CSS RPM limit. This limit is set by the system installer in AMP or can be reduced by programming a G92 block.
CSS RPM LIMIT THIRD SPINDLE	The spindle 3 RPM requested by CSS is greater than the maximum CSS RPM limit. This limit is set by the system installer in AMP or can be reduced by programming a G92 block.
CUR LOOP G/A CLOCK LOST	This error was generated by a servo amplifier error. It can usually be corrected by turning off power to the amplifier, and then back on.
CURRENT FEEDBACK ERROR	The servo module has detected faulty or missing current feedback from the digital servo motor. The most likely cause of this error is be a broken or disconnected wire.
CURSORING NOT ALLOWED	While assigning a {CUSTOM TOOL} in {RANDOM TOOL}, you cannot cursor to select another tool position.
CUTTER COMP/TTRC INTERFERENCE	The cutter radius is too large, reverse motion is required, or some other cutter compensation interference exists. Either an alternate tool or an alternate tool path must be programmed. Another option would be to disable cutter compensation error detection.
CYCLE ALREADY ACTIVE	An attempt was made to start a cycle while another cycle was currently executing.
CYLINDER RADIUS IS ZERO	The cylinder radius was not programmed in a virtual C cylindrical interpolation (G16.1) cycle.
CYLINDRICAL AXIS NOT PRESENT	Cylindrical interpolation was programmed without at least one cylindrical interpolation axes present (rotary, park, or feed axes).

Message	Description
CYLIND/VIRTUAL CONFIGURATION ERROR	An axis configuration error was detected by the control when cylindrical interpolation or end face milling was requested in a program block. Some examples would include: A cylindrical/virtual axis is named same as a real axis or is missing (for example on a lathe A, the cylindrical axis may have been named the same as a incremental axis name). A cylindrical/virtual axis is named the same as another programing command (for example a secondary auxiliary word, the angle word, etc...).
D	
D-WORD IS GREATER THAN TOOL DIA.	The programmed D-word value is greater than the tool diameter of the current tool.
D-WORD IS LESS THAN AMP THRESHOLD	The D-word has been programmed with a value that is too small.
D-WORD OUT OF RANGE	More than 1000 auto-dress operations were specified by the D-word in a grinder fixed cycle.
DAC MONITOR CIPC ON	This message comes up on power-up, after patch AMP has been modified to invoke DAC monitoring of the coarse incremental position command.
DAC MONITOR F. E. ON	The axis-following error is being output to the DAC output port for monitoring and debugging. Turning parameters 86 or 87 ON through patch AMP enables this output.
DAC MONITOR FV ON	This message comes up on power-up, after patch AMP has been modified to invoke DAC monitoring of the fine interpolated final velocity for each fine iteration (20ms).
DAC MONITOR INTEGRATOR ON	This message comes up on power-up, after patch AMP has been modified to invoke DAC monitoring of the velocity error integrator accum.
DAC MONITOR VEL ERROR ON	This message comes up on power-up, after patch AMP has been modified to invoke DAC monitoring of the velocity error.
DAC MONITOR VELOCITY ON	The axis velocity command is being output to the DAC output port for monitoring and debugging. Turning ON parameters 86 or 87 through patch AMP enables this output.
DATA MAY BE OUTPUT TO PRINTER ONLY	The information being output by the control is intended to go to a printer. Make sure that the output port that is selected is properly connected to a printer and try again.
DATA STARVED	The control is waiting for the next program block to set up. Typically, this is the result of the control executing a part program faster than it can be read from a peripheral device such as a tape reader. This error often occurs immediately after the execution of several very short, rapidly executed blocks. To prevent this error from reoccurring, it is recommended that the program be loaded into control memory or to a faster peripheral device.
DECIMAL POINT ERROR	A word or parameter has been programmed with more than one decimal point.
DECIMAL POINT NOT ALLOWED	A word or parameter has been programmed with a decimal point when it can legally exist only as an integer value. For example, the number of repetitions (L) must be an integer value programmed without a decimal point.
DEFAULT AMP LOADED	This indicates that the default AMP values stored in the control's executive memory have been activated. AMP in RAM and AMP in Backup memory were either unavailable or corrupt. This message can also occur if the battery backup fails.
DEFAULTS LOADED	The default device set-up parameters were loaded into the current device.
DEPTH > PROGRAMMED ENDPOINT	This error occurs during a threading cycle when the depth of the cut exceeds the programmed final depth of thread.
DEPTH PROBE AXIS MUST BE LAST	Adaptive depth probe is not AMPed as the last axis in the system. It must be AMPed after all normal axis and after the deskew slave and before any spindles. Refer to your AMP reference manual for details.
DEPTH PROBE AXIS NOT AMPED	A G26 (adaptive depth probe) move was programmed but no adaptive depth probe axis has been specified in AMP. Refer to your 9/Series AMP reference manual.
DEPTH PROBE FB GEARING NOT 1:1	The AMP configured gear ratio for the logical axis used as a depth probe must be a one to one ratio. "Reset Teeth on Motor Gear for Pos. FB" and "Teeth on Lead Screw Gear for Pos. FB".

Message	Description
DEPTH PROBE TRAVEL LIMIT	The adaptive depth probe has moved to its AMPed travel limit. Note the value entered in AMP is the adaptive depth probe deflection from the PAL determined probe zero point. It may not be the actual total probe deflection.
DEPTH PROBE NOT SUPPORTED	A depth probe axis has been AMPed on an axis located on a servo card or a 9/230 that does not support the adaptive depth feature. (analog servo rev < rev 0.10 or 3 axis 9/260 9/290 digital servo cards)
DESKEW OPTION NOT INSTALLED	If the AMPed name specifying the deskew slave servo is not zero, or the AMPed name specifying the deskew master servo is not zero, and the option flag for deskew is zero, then the system is held in E-Stop.
DEVICE ALREADY OPENED	An attempt was made to open a device for download or upload from ports A or B when the device was already opened.
DEVICE NOT OPENED YET	The ready signal was not received when attempting to send data to or communicate with a peripheral device connected to communication ports A or B.
DIAMETER AXIS MISCONFIGURED	An invalid axis has been configured as the diameter axis.
DIRECTORY CHANGED TO MAIN DIRECTORY	When a password is entered that does not have access to the protectable part program directory and the protectable part program directory is currently selected, the control changes the selected directory to the main directory.
DISP SELECT NOT ALLOWED	You can not use the display select functions while the online PAL search monitor utility is active. Leave the search monitor utility before you try to select a display.
DIVIDE BY ZERO ERROR	A system software error has occurred. Consult Allen-Bradley customer support services.
(DO) NUMBER ALREADY USED	When executing a program, an attempt was made to activate a DO loop that has the same loop identifier (DO 1, 2, or 3) as an already active loop in the program. Provided they are not nested loops, the same loop identifier can be used more than once in a program .
(DO) RANGES INTERSECT	DO loops are improperly nested. A DO loop that is nested within another DO loop does not have an END command before the original DO loop END occurs.
DRESS CANCEL DEFERRED TO G40	The in-process dresser cannot be canceled (made inactive) while dresser/wheel radius compensation is active. If an attempt to cancel the in-process dresser is made, the control will postpone the request until dresser/wheel radius compensation is canceled with a G40 (note that M02, M30, and M99 can also cancel compensation).
DRESSER AXIS NOT ALLOWED	An attempt was made to program the dresser axis when the over the wheel dresser feature has been activated through PAL. You cannot program the dresser axis when the over the wheel dresser feature is active.
DRESSER FLANGE LIMIT REACHED	While dressing the grinding wheel the wheel size reached the entered flange limit. You should stop dressing the wheel before damage to the wheel flange occurs.
DRESSER MINIMUM LIMIT REACHED	The current dressing operation would dress the grinding wheel below the minimum wheel diameter as specified on the dresser status screen. This dressing operation will not be performed.
DRESSER MISCONFIGURED	One of the AMP parameters for the dresser axis has not been configured properly. Either the dresser axis, the vertical axis, or some other axis name is not a valid axis in the system. You must re-configure your AMP. Refer to your AMP manual for details.
DRESSER MIS-POSITIONED	Wheel re-enable was requested with IPD active and wheel is more than 4 inch-programming counts (hard-code amount) away from its previously active absolute position. Wheel dressing does not start.
DRESSER NOT INITIALIZED	This error is generated if an attempt is made to activate the in-process dresser before the dresser has been initialized through a wheel calibration operation.
DRESSER NOT/MIS CONFIGURED	The grinder over-the-wheel dresser feature issues this message when a wheel is initialized and the dresser parameters in AMP have been misconfigured. This message is issued when the dresser axis, dresser vertical axis, or dresser other axis has not been selected, or has been AMPed to have common axes, or has been AMPed to be a non-existent axis name.

Message	Description
DRESSER WARNING LIMIT REACHED	The axis specified as the dresser axis has been dressed smaller than the dresser warning limit value as specified on the dresser status page.
DRILL AXIS CONFIGURATION ERROR	The drilling axis is not a currently configured machine axis. On dual processing controls this message may result when the drilling axis is in another process. The drilling axis must be a configured axis in the current process and should not be the slave of a dual axis (drill axis should be the master axis for dual group). On machines with dual axes, this message can mean the axis configured in AMP as the fixed-drilling axis is a slave axis. The drill axis should be the master axis.
DUAL AXES MASTER&SLAVE PROCESS NOT SAME	When configuring a dual axis on a dual processing system, configure AMP so all axes in the dual axis group are in the same source process even if the dual axis group is shared.
DUAL AXES PARK LOGIC CANNOT CHANGE	An attempt was made, using dual axes, to change the current park status. At this point, the request will not be allowed.
DUAL GROUP AXES MUST HAVE SAME ROLLOVER	All rotary axes in a dual axes group must have the same rollover value. These rollover values are set in AMP.
DUAL LATHE-MUST USE PROCESS 1,2	Dual lathe must have the active processes be the first 2 available in AMP; 3 or 4 should not be configured as an active process.
DUAL MASTER&SLAVE RAD/DIAM CONFIG ERR	The slave of a dual group has been defined as a diameter axis. The OEM must define the master to be a diameter axis and the system will change the slave to be a diameter axis. When the group is decoupled the slave will continue to take on the master's rad/diam traits.
DUAL PLANE CONFIGURATION ERROR	In AMP you have defined a plane with an axis and a master and a slave in the wrong order. For example: If the system has 4 axes YXZU and ZU are duals, if an AMPed plane is ZX, then UX can not be and AMPed plane. It must be XU (refer to your AMP reference manual for details).
DUAL SLAVE OR SPLIT AXIS NOT ALLOWED	Neither a dual slave, nor a split axis (deskew axis) may be programmed in a G24, G25, or G26 block.
DUALS CANNOT CHANGE OFFSETS IN CIRCULAR	An attempt was made, using dual axes, to account for an offset change in a circular move. Dual offset changes can only be made in linear blocks.
DUALS ONLY ALLOWING SINGLE AXIS HOME	An attempt was made to home multiple axes in a dual group when PAL only allows one axis at a time to be homed. PAL can be changed to allow homing of multiple axes in a dual group.
DUALPORT PTO TEST FAILED	The Dualport failed the diagnostic test and the bootstrapping operation is skipped. Consult Allen-Bradley customer support services.
DUPLICATE 1394 SLOT	The 1394 rack ID and slot number AMP entries are the same for two or more servos. Each axis module in a 1394 rack must have an individual address.
DUPLICATE 7300 PATTERN NAME	An attempt was made to enter a 7300 pattern name that already exists.
DUPLICATE DUAL MASTER NAMES	Both dual master axes names have the same letter.
DUPLICATE I/O RING DEVICE	Two or more of the same type of device on the I/O ring have the same device address switch setting.
DUPLICATE PROGRAM	An attempt was made to rename a program in control memory using the same program name (or number) of another program already in memory.
DUPLICATE PROGRAM NAME	An attempt was made to store or copy a program in control memory using the same program name (or number) of another program already in memory.
DWELL VALUE NOT PROGRAMMED	A G04 Dwell or a parameter requesting a dwell at hole bottom in a fixed drilling cycle was programmed with no value assigned to the length of the dwell.
E	
(E) AND (F) IN SAME BLOCK	In a G32 block (Lathe A) or G33 block (Lathe B & C), both leads were programmed in the same block.
EMPTY PROGRAM WAS DELETED FROM DIRECTORY	The current program being edited was saved and contained no program blocks. This program was deleted from the control's program directory.

Message	Description
ENCODER QUADRATURE FAULT	An error has been detected in the encoder feedback signals. Likely causes are excessive noise, inadequate shielding, poor grounding, or encoder hardware failure.
END OF FILE	When transferring a file over the serial port, the control has reached the last block in the program.
END OF PROGRAM	When displaying a part program on the CRT, the control has reached the last block in the program.
END OF PROGRAM REACHED	When performing one of the program search features, the control has reached the last block in the program.
ENTER ALL REQUIRED PROMPT DATA	An attempt was made to create a transfer line part program from the quick view screen without entering all the required quick view screen prompt data. Optional data is shown in reverse video.
ENTRY OUT OF RANGE	A parameter value was entered that is larger or smaller than the usable range determined in AMP or allowed on the system.
ERASE PROMPT	The operator has data on the input line (line 2 of the CRT) that must be cleared or entered so that a new prompt can be displayed on the input line.
ERROR ACCESSING PROGRAM	A major software error was generated by the control's internal software when editing the program; the program should be deleted. If the error persists, contact Allen-Bradley customer service support.
ERROR FOUND	A QuickCheck syntax check operation has found an error in the currently displayed program block. This is the block after the block containing the block-completed symbol "@". Press <CYCLE START> to continue the program check.
ERROR IN CIRCLE DATA	This error can occur when digitizing a circular block, typically the result of entering positions that cannot be correctly connected with an arc.
ERROR LOOKING FOR (END) COMMAND	The control has found a paramacro END command that does not match one of the active paramacro DO loop ranges.
ERROR TRANSFERRING PAL TO CPU #2	An error occurred while PAL was being transferred to the I/O CPU at power-up. PAL is transferred to the I/O CPU at power-up on a 9/290. Consult Allen-Bradley customer support services.
EXACTLY 2 DIGITS MUST FOLLOW DECIMAL PT	You have used incorrect search string syntax in the PAL search monitor utility.
EXCESS FOLLOWING ERROR	The following error for an axis exceeds the allowable value as defined in AMP. Most likely cause is AMP servo related parameters are set too stringently for the hardware. Also caused by axis runaway.
EXCESS SKEW ON	The calculated skew is larger than the AMPed maximum allowable skew.
EXEC BOOTSTRAP FAILED	The bootstraper failed to respond within the specified time for any code segment. Consult Allen-Bradley customer support services.
EXPRESSION INCOMPLETE	A syntax problem has been found in a paramacro expression. The control is unable to correctly evaluate the expression as entered.
EXTRA DATA IN INTERRUPT MACRO BLK	An attempt was made to program extra data (such as a G-code) in the M-code block that activates or deactivates an interrupt program. No extra commands can be programmed in this block.
EXTRA DATA IN QPP BLOCK	The QuickPath Plus block has been programmed with too many parameters. For example, you cannot program a G13 block with both axis data and an angle word or with an L or A word in the block.
EXTRA I/O RING DEVICE	An I/O device that has not been defined in the I/O assignment file is physically present on the I/O ring.

Message	Description
EXTRA KEYBOARD OR HPG ON I/O RING	The control detected a keyboard or HPG on the 9/Series fiber optic ring that was not configured as a ring device. The I/O ring will still function and the control will NOT be held in E-Stop. You may also use the keyboard or HPG by selecting it as the active device via the corresponding PAL flags. You should configure the keyboard or HPG with the I/O assigner utility (See your 9/Series PAL reference manual for details).
F	
FCM DUALPORT RAM FAILURE	The FCM detected an error in dualport RAM.
FCM FLASH RAM FAILURE	The FCM detected an error in flash RAM.
FCM LOCAL RAM FAILURE	The FCM detected an error in local RAM at power-up or during the runtime diagnostics.
FCM PLUG CONFIGURE FAILED	The FCM card failed to configure correctly.
FCM PLUG FAULT	The plug on the FCM detected an error.
FCM PLUG NEGOTIATE FAILED	The FCM firmware could not communicate with the plug.
FCM POWER UP SEQUENCE FAILURE	Power-up failed. Try again. If error appears again, contact your Allen-Bradley sales representative.
FCM REVISION CHECK FAILURE	Revision on module is out-of-date. Contact Allen-Bradley sales representative to get latest revision of the module's firmware.
FCM ROM FAILURE	The FCM detected an error in ROM during runtime diagnostics.
FCM SHADOW RAM FAILURE	The FCM detected an error in shadow RAM.
FCM SPURIOUS INTERRUPT	A spurious interrupt occurred on the FCM card.
FCM VRTX ERROR	A call from VRTX from the FCM card firmware returned an error.
FCM WATCHDOG	The watchdog on the FCM card timed out.
FDBK NOT AVAILABLE ON 4TH AXIS OF BOARD	An attempt was made to receive feedback from the axis that is configured as the fourth axis on a servo board. You can only receive feedback from the first three axes on a servo board.
FEEDBACK DISCONNECTED	The control has detected a loss of feedback from the encoder. The most likely cause of this error would be a broken or disconnected wire. Axis homing will be required after the error condition is corrected.
FEEDBACK OPTION NOT INSTALLED	A PTO check determines the legal number of axes.
FEED AXIS DATA NOT PROGRAMMED	Feed axis data required during a grinder fixed cycle was not programmed.
FEED AXIS MOTION NOT ALLOWED	During Virtual C programming, no axis motion is allowed on the axis specified as the feed axis in AMP.
FEED TO HARDSTOP PROGRAMMING ERROR	No axis, or more than one axis, was programmed in a G24 block. Or the programmed axis integrand was not programmed in the block.
FILE CANNOT BE CONVERTED TO EIA FORMAT	The file requested to be output to a device has characters that cannot be converted to EIA.
FILE DOWNLOAD COMPLETE	Status message that means the download has completed.
FILE DOWNLOAD ERROR	Check file download and file download configuration screens to make sure all fields are entered correctly.
FILE DOWNLOAD IN PROGRESS	This status message means a file is being downloaded.
FIXED CYCLE ALREADY ACTIVE	You cannot program a fixed cycle with a fixed cycle already active.
FIXED CYCLE PROGRAMMING ERROR	A fixed cycle has been programmed incorrectly. Verify that the correct parameters have been used and that parameters restricted to integer or positive values are programmed as such.
FLASH IN USE - TRY AGAIN LATER	Only one task is allowed to write flash at a time. If a second task requests a flash write, you will see this message.
FLASH SIMMS ARE NOT INSTALLED	Install the flash SIMMs into the 9/Series mother board. Flash SIMMs must be installed. If a repaired system is being installed, you should have saved your flash SIMMs for re-installation before making the return.

Message	Description
FLASH SIMMS CONTAIN INVALID DATA	Flash SIMMs have become corrupted probably from a communication error during a system update. Retry the system executive update utility. If the situation persists, contact Allen-Bradley support.
FLASH SIMMS U10 AND U14 ARE EMPTY OR MISSING	Make sure your flash SIMMs are installed in the correct tracks. Refer to the 9/Series integration and maintenance manual section covering your processor for details on flash installation. Remove and reseal flash SIMMs.
FLASH SIMM U10 IS EMPTY OR MISSING	Make sure your flash SIMMs are installed in the correct tracks. If they appear to be installed correctly, remove and reseal SIMMs. If problem persists, contact Allen-Bradley support service.
FLASH SIMM U14 IS EMPTY OR MISSING	Make sure your flash SIMMs are installed in the correct tracks. If they appear to be installed correctly, remove and reseal SIMMs. If problem persists, contact Allen-Bradley support service.
FLASH WRITE ERROR	A problem occurred while writing to flash, for example bad flash, no flash, or no voltage.
BACKGROUND OVERLAP	The foreground tasks did not complete execution within the 20-millisecond period allocated. Foreground tasks include PAL foreground, axis interpolation, servo interface, and I/O ring scanning. Correct by reducing PAL foreground program size or removing some devices from the I/O ring.
G	
G10 NOT ALLOWED DURING CYCLE	G10 code is not allowed to be used during the cycle. Cancel the fixed cycle.
G24 NOT ALLOWED	G24 is not allowed when any automatic G coded cycle is active (such as G81).
G24 PLANE INCOMPATIBILITY	The hard stop axis may not be in the active part rotation plane.
G25 NOT ALLOWED	G25 is not allowed when any automatic G coded cycle is active (such as G81).
G25 PLANE INCOMPATIBILITY	The adaptive feed axis may not be in the active part rotation plane.
G26 NOT ALLOWED	G26 (adaptive depth probe) can not be programmed when another modal group is active (such as a G81 fixed cycle).
G26 PLANE INCOMPATIBILITY	A conflict between a plane dependent feature and a G26 (depth probe). For example if part rotation is active and a G26 is programmed on an axis in the part rotation plane this error is generated. Refer to the 9/Series users manual section on G26 for details on incompatible planar features.
G28 BLOCK DOES NOT PRECEDE G29 BLOCK	A G29 block was programmed before a G28 block. During 7300 tape compatibility mode, the first automatic threading block must contain a G28 code; the next block must contain a G29 code.
G29 BLOCK DOES NOT FOLLOW G28 BLOCK	A G28 block was programmed without a following G29 block. During 7300 tape compatibility mode, the first automatic threading block must contain a G28 code; the next block must contain a G29 code.
G40 NOT ALLOWED IN CIRCULAR	An exit move from cutter compensation or TTRC was attempted in a circular block (G02 or G03). An exit move (programmed with a G40) must generate a linear move.
G53 NOT ALLOWED IN G91 MODE	An attempt was made to make an incremental move in the machine (absolute) coordinate system. Only absolute moves (G90) are permitted in the machine coordinate system.
G53 NOT ALLOWED IN INCREMENTAL MODE	A G53 move to absolute position was requested while in incremental mode.
G53 ON AN UNHOMED AXES	An attempt to program a move in the machine (absolute) coordinate system was made before the axis was homed. It is necessary to home the axes to establish the location of the machine coordinate system.
G91 MODE NOT ALLOWED IN QPP	Since QuickPath Plus is generally used to program blocks without knowing the intersection of the blocks, it is impossible to calculate a location for the end-point of the block when the move is incremental. QuickPath Plus must be programmed in absolute mode (G90).
(G-CODE) TABLE ERROR	There has been an internal software fault relative to the G-code table. Consult Allen-Bradley Customer Support Services.
(GOTO) SEQ. NUMBER NOT FOUND	The sequence number (N word) called by a GOTO command does not exist in the currently executing program.

Message	Description
GRAPHICS ACTIVE IN ANOTHER PROCESS	Graphics can only be active in one process at a time. You must turn graphics off in one process before you can activate them in another process.
H	
HARD STOP ACTIVATION ERROR	An attempt was made to (G24) hard stop an axis while a different axis was already holding against a hard stop.
HARD STOP AND/OR ADAPTIVE DATA CONFLICT	An attempt was made to create a transfer line part program from the quick view screen entering data for both hard stop (G24) and adaptive depth features (G26). You can select only one of these features.
HARD STOP AXIS NOT ALLOWED IN INTERRUPT	An axis which is still hard-stopped due to a previous G24 block may not be moved by any block inside an interrupt macro program.
HARD STOP DETECTION ERROR	A hard stop (G24) was detected outside of the programmed hard stop region. Or a hard stop was not detected before the hard stop axis reached its endpoint.
HARD STOP DIRECTION ERROR	The axis currently holding against a hard stop (G24) was programmed with a move further into the hard stop. You must program the move away from the hard stop in the direction opposite to the direction used to place the axis at the hard stop.
HARD STOP EXCESS ERROR	The hard stop axis (G24) was moving too fast when it encountered the hard stop. You must reduce the axis feedrate before contacting the hard stop.
HARDWARE ERROR #1	The 68030 (68000 on 9/240 only) main processor received an interrupt of unknown origin on level 6. Consult Allen-Bradley customer support services.
HARDWARE ERROR #2	The 68030 (68000 on 9/240 only) main processor has detected a bus error. Consult Allen-Bradley customer support services.
HARDWARE ERROR #3	The 68030 (68000 on 9/240 only) main processor has detected a spurious interrupt. Consult Allen-Bradley customer support services.
HARDWARE ERROR #4	The 68030 (68000 on 9/240 only) main processor has detected an illegal address. Consult Allen-Bradley customer support services.
HARDWARE ERROR #5	The PAL program residing in RAM memory has failed a checksum test. Attempt to download your PAL program to the control again. If the error remains, consult Allen-Bradley customer support services.
HARDWARE ERROR #6	The 68030 (68000 on 9/240 only) main processor has detected a privilege violation. Consult Allen-Bradley customer support services.
HARDWARE ERROR #7	The AMP data in Backup memory has failed a checksum test. Attempt to download your AMP program to the control again and again try to store it in Backup memory. If the error remains, consult Allen-Bradley customer support services.
HARDWARE ERROR #10	The servo processor RAM diagnostic test has failed. Consult Allen-Bradley customer support services.
HARDWARE ERROR #12	The servo communications timing diagnostic test between the main processor and the servo processor has failed. Consult Allen-Bradley customer support services.
HARDWARE ERROR #13	The main processor was not ready in time to send data to the servo processor. Consult Allen-Bradley customer support services.
HARDWARE ERROR #14	The servo processor sent an invalid error code to the main processor. Consult Allen-Bradley customer support services.
HARDWARE ERROR #15	The servo communications data echo diagnostic test between the main processor and the servo processor has failed. Consult Allen-Bradley customer support services.
HARDWARE OVERTRAVEL (+)	The indicated axis has reached a travel limit in the positive direction.
HARDWARE OVERTRAVEL (-)	The indicated axis has reached a travel limit in the negative direction.
HIPERFACE COMMUNICATION ERROR	A serial communications error (e.g., CHECKSUM, TIMEOUT) was detected within the SINCOS device during power-up. If this error occurs at PTO, check your feedback device to make sure it is not disconnected.

Message	Description
HIPERFACE PASSWORD FAILURE	During the SINCOS device's alignment procedure, the logic used to set the passwords detects an incorrect password. A section of the code will repeatedly attempt various combinations of each of the passwords to correct the error condition.
HOME REQUEST ON A PARKED AXIS	An attempt was made, while using dual axes, to do a homing operation on a parked axis.
HOMING NOT COMPLETED	An attempt was made to execute a programmed axis move before the axes have been homed. Axes must be homed before they can be moved through part program or MDI commands.
HOME TOLERANCE EXCEEDED	This error condition occurs when the homing operation of the indicated axis is aborted due to the condition of the axis stopping off of the home switch in excess of 3/8 of an electrical cycle of the feedback device away from a null marker. If this problem persists you may need to adjust either the home switch position, or the home switch trip dog, or the feedback position. This may also require you to adjust the AMP Home Calibration parameter for this axis.
I/O RING COMMUNICATIONS ERROR	A communication error has occurred in the fiber optic I/O ring. This is usually caused by a broken or disconnected fiber optic cable.
I/O RING NOISE WHILE IDLE	An illegal character was detected by an optical receiver while the I/O ring should have been idle. The system will try to reset itself. If it cannot reset itself, the system enters E-Stop.
I/O RING NOT CONFIGURED	The control cannot run the I/O ring if it was not configured and downloaded from ODS or resident in the PAL PROMs.
I/O RING TIME-OUT	A very large foreground PAL program, combined with a large number of I/O ring devices, has created timing problems for I/O ring communications. Reduce PAL program size by deleting unnecessary rungs and optimize the execution of others.
ILLEGAL (/) VALUE	A block delete slash value greater than /9 was programmed. There are only 9 block deletes available.
ILLEGAL (G) CODE	An illegal G-code value has been programmed.
ILLEGAL (M) CODE	An illegal M-code value has been programmed.
ILLEGAL ANGLE VALUE	A QuickPath Plus block has defined the angle of the next block incorrectly. There is no possible path that connects the two tool paths to the programmed end-point using the entered angle.
ILLEGAL APPLICATION COMMAND FROM TEACH	A non-recognized SD1-type packet was received in a CMD=61 DF1 packet from the teach pendant interface. Allowable SD1s are 1 - 5.
ILLEGAL AXIS DATA FORMAT	Digitized axis data does not fit within the allowable AMPed axis format. For example, if an axis inch format is set at 2.3 and a digitized position is recorded as 121.0, an error will be generated. The axis display will also show "____".
ILLEGAL CHARACTER	An undefined character was entered in a program block and could not be executed. Certain characters cannot be recognized while in certain modes. Also verify that you are using the correct axis and integrand names as assigned in AMP.
ILLEGAL CODE DURING G41/G42	An illegal code was encountered during G41/G42 programming.
ILLEGAL CODE DURING VIRTUAL C	An illegal code was encountered during Virtual C programming.
ILLEGAL CODES IN RANDOM TOOL BLOCK	An invalid parameter was entered in a G10.1L20 block that loads data into the Random Tool table. P, Q, R, and O are the allowable parameters.
ILLEGAL COMMAND FROM ODS	A command was received from ODS that was not recognized by the control.
ILLEGAL COMMAND FROM TEACH PENDANT	A non-recognized CMD-type packet was received in a DF1 packet from the teach pendant interface. Allowable CMDs are 60 - 63.
ILLEGAL CONTROL TYPE	You have downloaded from a peripheral device an AMP that does not match your control hardware.
ILLEGAL CPU #2 COMMAND	The 68000 main processor sent incorrect data to the Z80 I/O ring processor. (two 98030's instead of the 68000 and Z80 on 9/230, 9/260, and 9/290 controls)
ILLEGAL CYLINDRICAL BLOCK	A G-code not allowed in a cylindrical entry block or during cylindrical programming mode was programmed.

Message	Description
ILLEGAL DUAL CONFIGURATION	Both dual master axes names have the same letter OR when assigning dual groups in AMP, dual groups must be assigned in contiguous order, starting with group 1, 2, 3, 4, and 5. You can not assign axes to dual group 3 without axes having been assigned to dual groups 1 and 2.
ILLEGAL DUAL LINEAR/ROTARY CONFIGURATION	The dual group cannot contain a mixture of linear and rotary axes.
ILLEGAL FILENAME	An attempt was made to create a program using a program name that contains illegal characters. A different program name must be used.
ILLEGAL G40 EXIT BLOCK	An illegal sequence of exit moves was programmed in a G40 exit block.
ILLEGAL G88.5 OR G88.6 PARAMETERS	Illegal parameters were entered in a G88.5 or a G88.6 programming block.
ILLEGAL G99	An illegal G99 was entered in a programming block.
ILLEGAL G-CODE IN INTERRUPT MACRO	An illegal G-code has been programmed in a program called by a program interrupt. G24, G25, G26, G40, G41, G42, G52, G92, and G92.1 cannot be programmed in an interrupt program.
ILLEGAL G-CODE IN POCKET	An illegal G-code was entered in a G88 or G89 pocket-programming block.
ILLEGAL I/O RING DEVICE CODE	There is a device on the I/O ring that cannot be identified.
ILLEGAL I/O RING RACK SLOT CODE	There is a card in the 1771 I/O rack that the I/O ring cannot use.
ILLEGAL INPUT	A number was input from the keyboard instead of a character, or a character was input instead of a number.
ILLEGAL MACRO CMD VIA MDI	A paramacro command that cannot be used in MDI mode was programmed. This also can include an illegal sub-program return M99 code.
ILLEGAL MASTER AXIS NAME	Slave axes that do not have a master have been configured for a dual group OR you have assigned a \$ axis name as a group master. Axis names beginning with a \$ can not be assigned as the master axis for a dual group (first logical axis assigned to the group).
ILLEGAL PAL PRIORITY MESSAGE	This error message displays each time your PAL logic program attempts to save a machine error message that contains dynamic data in the error log. Contact your system installer for assistance with this problem.
ILLEGAL PASSWORD	A password was entered that was not assigned to one of the 8 different password levels. Make sure that no one has changed the passwords by using {ACCESS CONTRL}.
ILLEGAL PLANE - USING SLAVE AXIS	This is a power turn-on message. When using dual axes, one of the slave axes was AMPed as part of the plane configuration. Only master can be used in the plane configuration.
ILLEGAL PLANE DEFINITION	The axis plane assignment made in AMP is incorrect. It can also occur if the two axes assigned to a plane have the same axis name.
ILLEGAL PROGRAMMED RETURN GROUP	The tool group programmed in an M06 block must be the currently active tool group that is being replaced (not the tool group you are changing to). This requirement is configured in AMP by the system installer.
ILLEGAL PROGRAMMED RETURN TOOL	The tool number programmed in an M06 block must be the currently active tool number that is being replaced (not the tool number you are changing to). This requirement is configured in AMP by the system installer.
ILLEGAL RANDOM TOOL TABLE ASSIGNMENT	An attempt was made to program a G10.1L20 block that would assign a tool to a tool pocket that already has a tool assigned to it.
ILLEGAL RECIPROCATION INTERVAL	The programmed reciprocating interval is greater than the total rollover distance.
ILLEGAL ROTATION PLANE SELECTED	When using the external part rotation feature, the external part rotation plane selected on the rotation parameter screen is not the currently active plane in the program block being executed.
ILLEGAL SPINDLE PROCESS NUMBER	An illegal process number was used to indicate a process that uses one of the spindles.
INCOMPATIBLE PAL SOURCE	The PAL search monitor utility can not be accessed. The PAL search monitor utility requires PAL program built with a newer version of ODS.

Message	Description
INCOMPATIBLE TOOL ACTIVATION MODES	This message is displayed and the control is held in E-Stop at power up when the tool geometry offset mode is "Immediate Shift/Immediate Move" and the tool wear offset mode is "Immediate Shift/Delay Move" or when the tool geometry offset mode is "Immediate Shift/Delay Move" and the tool wear offset mode is "Immediate Shift/Immediate Move". These modes are incompatible. You must correct your AMP configuration and re-download AMP.
INCORRECT NUMBER OF SYMBOLS	An error occurred in G05 DH+ communications block.
INPUT DATA TOO LONG	The data input has a number of characters exceeding the allowable number of characters.
INPUT STRING SYNTAX ERROR	An attempt was made to search for an illegal character string, or no character string was entered.
INSUFFICIENT MEMORY FOR PAL PROGRAM	An attempt was made to load a PAL program that exceeded the PAL storage space limit allotted by the flash memory.
INSUFFICIENT MEMORY FOR PART PROGRAM	There is not enough available memory for the current program to be stored. Any attempt to store the program in memory will be aborted by the control.
INTEGRANDS FOR DUALS MUST BE THE SAME	This is a power turn-on message. When using dual axes, all integrands of the dual group must use the same letter.
INTEGRANDS FOR NON-MASTER MUST BE NONE	An axis integrand name was configured in AMP that corresponds to an axis in a dual axis group that is not the master axis of that group. Only the master axis in a dual axis group can have a corresponding axis integrand name.
INTEGRANDS NOT AMPED PROPERLY	The axis integrand names were not configured properly in AMP. Refer to your AMP manual for additional details on axis integrand names.
INTERF CHECKING ZONE TABLE CORRUPTED	The zone tables used by interference checking have an invalid checksum and were cleared.
INTERNAL COMMUNICATIONS ERROR	Communication failed. Contact Allen-Bradley customer support services.
INTERRUPT NOT RECOGNIZED	An interrupt macro was not acted on for some reason. An example would be if an interrupt occurred in the middle of another interrupt.
INVAL LOOP BASE	An attempt was made to configure ports TB2 and TB3 as position/velocity loop or digital or digital spindle.
INVALID AMP-DEFINED G CODE	An attempt was made to assign the same G-code to different macro calls. This message appears after AMP is downloaded and the control does secondary calculations.
INVALID AMP LETTER FORMAT	The programmed word or parameter has an invalid letter format defined in AMP. Since ODS AMP detects and prohibits invalid formats, this error usually indicates that an invalid format was entered through patch AMP. Refer to your AMP reference manual for details.
INVALID ARC-COSINE ARGUMENT	An attempt was made to calculate or execute a paramacro block that calculates the arc cosine of an invalid or improperly entered number.
INVALID ARC-SINE ARGUMENT	An attempt was made to calculate or execute a paramacro block that calculates the arc sine of an invalid or improperly entered number. Change cosine to sine.
INVALID ARGUMENT ASSIGNMENT	An invalid argument assignment was programmed.
INVALID AXIS	The axis programmed in the adaptive depth (G26) or adaptive feedrate (G25) block is invalid. Valid axis names for programming these features are defined in AMP.
INVALID AXIS FOR CSS	The CSS axis (the axis that is perpendicular to the center-line of the rotating part) is not a valid axis on the control. This usually occurs when the CSS axis is changed from the default axis by programming a P-word in the G96 block that selects some other axis.
INVALID AXIS PROGRAMMING RESOLUTIONS	The axis resolutions set in AMP by the system installer are too far apart. The control is incapable of handling large differences in axis resolutions. For example, if the X axis has a resolution that allows .999999 and the Z axis allows a resolution of only .9, the control can have difficulty moving both axes simultaneously.
INVALID CCT INDEX	An error occurred in G05 DH+ communications block.
INVALID CHANNEL NAME	An error occurred in G05 DH+ communications block.
INVALID CHARACTER	A program name has been entered that contains an illegal special character.

Message	Description
INVALID CHECKSUM DETECTED	This error is common for several different situations. Most typically it results when writing or restoring invalid data to flash memory. For example if axis calibration data is being restored to flash and there was an error or invalid memory reference in the axis calibration data file. Typically this indicates a corrupt or invalid file.
INVALID CNC FILENAME	An error occurred in G05 DH+ communications block.
INVALID CODE PROGRAMMED FOR 7300	An invalid G or M code was programmed during 7300 tape compatibility mode.
INVALID COMMUNICATIONS PARAMETER	Parameters in G05 and/or G10.2 communication blocks are incorrect.
INVALID CONTROL FOR DUAL PROCESS SYS	The system executive downloaded to the control does not match the hardware configuration established by your option chip.
INVALID CUTTER COMPENSATION NUMBER	A compensation number (or TTRC number) out of the range of allowable compensation numbers (either too large or too small) was programmed.
INVALID CYCLE PROFILE	The path defining the cycle profile is not valid. This is typically caused by the cutter radius being set to the wrong sign, being set too large, or the values for U, W, I, K, and the cutter radius combined are not valid for the profile to be cut.
INVALID DATA AFTER A MACRO COMMAND	Typically caused by a non-paramacro command following a paramacro command. Macro and non-macro commands cannot exist in the same block.
INVALID DATA BEFORE A MACRO COMMAND	Typically caused by a non-paramacro command preceding a paramacro command. Macro and non-macro commands cannot exist in the same block.
INVALID DATA FORMAT MUST BE MM/DD/YY	An invalid date format was entered. The format must be Month / Day / Year (MM/DD/YY).
INVALID DEPTH PROBE CONTROLLING AXIS	The axis name which is AMPed as the controlling axis for the depth axis is not an axis that has been configured on the system or the adaptive depth controlling axis is configured as the same axis defined to have depth probe feedback. Refer to your AMP reference manual for details on axis configuration.
INVALID DESKEW MASTER	The AMPed name specifying the master deskew servo is not one of the AMPed axes.
INVALID DESKEW SLAVE	The AMPed name specifying the master deskew servo is not one of the AMPed axes, or it has already selected as a master axis.
INVALID DH COMMAND TYPE	An error occurred in G05 DH+ communications block.
INVALID (DO) COMMAND NUMBER	The specified loop number in a paramacro DO command is out of the legal range, or not found. DO commands must be followed by a 1, 2, or 3.
INVALID (END) COMMAND NUMBER	A paramacro END command has been encountered without a matching DO or WHILE, or outside the valid range. END commands must be followed by a 1, 2, or 3, as programmed with the corresponding DO command.
INVALID ENDPOINT IN G27 BLOCK	The position programmed in the G27 block is not the home position. The end-point of a G27 block must be the machine home position.
INVALID EXPECTED LIFE	The data just entered for the expected life of the cutting tool for tool management is invalid.
INVALID EXPECTED TOOL LIFE	The current program is attempting to enter an invalid value for the tool management expected life of a tool. Tool life is programmed in a G10 block with an L-word.
INVALID FB COUNTS	At power up the control checks the AMP configured position and velocity feedback counts per revolution. If either of these parameters are invalid (for this hardware type) this error appears and the control is held in E-Stop.
INVALID FDBK/MTR TYPE COMBO	When changing between an executive from system 9.xx to 10.xx some major changes occurred to how a servo is configured in AMP. When copying this AMP project from 9.xx to 10.xx you must open and reconfigure some of the AMP servo group parameters before saving and downloading to the control.
INVALID FILE TYPE	An error has occurred in a file that has been sent from the ODS workstation to the control. Typically it is the result of ODS sending the wrong file type to the control (for example, an AMP file is sent when a PAL download is taking place, etc.). Attempt to download the file again, making sure that the correct file type is selected when downloaded.

Message	Description
INVALID FIXED DRILLING AXIS	The axis selected as the drilling axis is an invalid axis for a drilling application.
INVALID FORMAT SPECIFIED IN B/DPRNT CMD	Improper format was used in the paramacro command (BPRNT or DPRNT) that outputs data to a peripheral device.
INVALID FUNCTION ARGUMENT	An invalid paramacro argument was used in a paramacro function. The argument contains either bad syntax or an illegal value.
INVALID G10 CODE	The format for a G10 block is not correct. Refer to your user manual for the correct format for the G10 block that is currently being programmed.
INVALID IN ANGLED WHEEL MODE	A feature that is not available in G16.3 mode, or G16.4 mode or both has been programmed. Refer to your grinder users manual angled-wheel grinder section for a description of features not available on an angled-wheel grinder.
INVALID INFEEED (P WORD)	Infeed value (P-word) is not in valid range. The valid range for a P-word during a threading cycle is whole numbers 1 through 4.
INVALID INPUT VALUE	The data entered is invalid for the current operation being performed.
INVALID INTERFERENCE AREA	A G10 block has programmed a zone where the plus value is less than the minus value.
INVALID INTERFERENCE CHECK AXIS	An axis from the wrong process was AMPed. Unless a shared axis is used in the zone, the axis defined to make up an interference area must be in the process the zone is defined for.
INVALID LATHE AXIS	An illegal code was encountered during cylindrical interpolation programming.
INVALID LIFE TYPE	The current program is attempting to enter an invalid tool life type for a tool group in the tool management tables. Valid tool life types are type 0, 1, or 2. Tool life type is programmed in a G10 block following a I-word.
INVALID M99 IN MAIN PROGRAM	An M99 part program rewind and auto start was programmed in the middle of the main program. An M99 can be programmed only at the end of a part program.
INVALID MACRO COMMAND	The IS and IM commands are reserved for use by the control only for program interrupts. They cannot be entered in a part program or MDI program.
INVALID MACRO FROM TAPE	You have programmed a paramacro command that cannot be executed from tape.
INVALID NUMBER OF POCKETS	This error occurs when using G10 L20 to enter random tool data and the number of pockets needed for the tool is invalid.
INVALID OFFSET NUMBER	An offset number out of the range of allowable offset numbers (either too large or too small) was programmed.
INVALID OPERATOR IN EXPRESSION	Check expressions to make sure they are correct.
INVALID OPERATOR IN PARAMACRO EXPRESSION	The control has encountered a non-mathematical operator (character) in a paramacro expression or calculate operation.
INVALID OUTPUT FORMAT	An error occurred in G05 DH+ communications block.
INVALID PARAMACRO ARGUMENT ASSIGNMENT	An argument assignment in a block that calls a paramacro program contains either an invalid argument specification or a syntax error was made in the argument.
INVALID PARAMETER NUMBER	An attempt was made to assign or read the value of a paramacro parameter that does not exist.
INVALID PARAMETER VALUE	An attempt was made to assign an invalid value (typically too large or too small of a value) to a paramacro parameter.
INVALID POCKET NUMBER	An attempt was made to enter a tool pocket number that exceeds the allowable number of tool pockets in the random tool table. This error occurs when a P-word that is too large or too small is programmed in a G10.1L20 block.
INVALID POCKET PROFILE	An invalid pocket profile was programmed in a lathe roughing or finishing cycle.
INVALID POSITION FB TYPE	System was incorrectly AMPed with a Yaskawa type encoder (absolute or incremental) on the position feedback device when separate position and velocity feedback devices are used.

Message	Description
INVALID PROGRAM NUMBER (P)	A program number called by a sub-program or paramacro call is invalid. A P-word that calls a sub-program or paramacro can only be an all-numeric program name as many as 5 digits long. The O-word preceding the numeric program number in control memory cannot be entered with the P-word.
INVALID REMOTE NODE NAME	An error occurred in G05 DH+ communications block.
INVALID REMOTE STATION TYPE	An error occurred in G05 DH+ communications block.
INVALID REPEAT COUNT (L)	An L parameter that programs the number of times a paramacro or other operation is to be repeated was programmed incorrectly or out of the legal range. The L-word for repeat count must be a whole, positive number. Decimal values and negative values are invalid. The maximum value of an L-word is 9999.
INVALID ROUGHING CYCLE (P/Q) WORD VALUE	When executing a roughing cycle, the starting or ending sequence number of the contour defining blocks cannot be found in the currently executing program. The sequence number of the contour blocks is programmed using the P and Q words. These blocks can be anywhere in the program provided they are resident in the same program, sub-program, or paramacro program that contains the calling block.
INVALID SCALE FACTOR (P-WORD)	An invalid scale factor has been specified. The P-word has a range of 0.0001 to 999.99999.
INVALID SERVO HARDWARE TYPE	The AMP servo parameter that selects the servo type does not match the hardware found on the control when the AMP file is downloaded. Either AMP is misconfigured or the servo hardware installed on your system is not correctly installed or not of the correct type.
INVALID SHAFT POCKET	When entering a custom tool in the random tool table, an attempt was made to assign a shaft pocket position that is not in the range of the number of pockets assigned to the tool. The shaft pocket number must be equal to or less than the number assigned for the number of pockets.
INVALID SHAFT POCKET VALUE	A program is attempting to enter a custom tool in the random tool table with a invalid shaft pocket position (not in the range of the number of pockets assigned to the tool). The shaft pocket number must be equal to or less than the number assigned for the number of pockets. The shaft pocket value is assigned in a G10.1 block following the R-word.
INVALID SPCMD VALUE	A invalid special command error typically occurs when the servo PROMs are not compatible with the main processor PROMs. Check the software version numbers and contact Allen-Bradley customer support services.
INVALID SYMBOL NAME	An error occurred in G05 DH+ communications block.
INVALID T-CODE FORMAT	This is an invalid T-Code Format
INVALID_THREAD_ANGLE	An attempt was made to program an angle that is outside the allowable range, which is 0 through 120 degrees.
INVALID THRESHOLD RATE	An invalid threshold percentage was entered for a tool group while setting tool management data. The threshold percentage must range between 0 and 100 percent. Only whole positive numbers can be entered. If using a G10 block, the threshold percentage is entered with a Q-word.
INVALID TIME FORMAT MUST BE HH:MM:SS	An invalid time format was entered. The time format must be hour / minute / second (HH/MM/SS).
INVALID TOOL AXIS	This is an invalid Tool Axis.
INVALID TOOL CUTTER COMPENSATION NUMBER	An attempt was made to enter a tool radius offset number, for cutter compensation or TTRC, in the tool life management table that is larger than the maximum offset number allowed. If the tables are being loaded by a G10 program, the radius offset is entered with a D-word in the block.
INVALID TOOL DIAMETER VALUE	An invalid tool diameter value was entered in a program block.
INVALID TOOL GROUP	An attempt was made to create a tool group greater than 200 in the tool management tables. A maximum of 200 tool groups can be used. If loading the tables using a G10 program, the tool group number is entered using a P-word.

Message	Description
INVALID TOOL LENGTH OFFSET NUMBER	An attempt was made to enter a tool length offset number in the tool life management table that is larger than the maximum offset number allowed. If the tables are being loaded by a G10 program, the length offset number is entered with a H-word in the block.
INVALID TOOL LIFE TYPE	An attempt was made to enter an invalid tool life type for a tool group in the tool management tables. Valid tool life types are type 0, 1, or 2.
INVALID TOOL NUMBER	Either no tool or an invalid tool number was programmed in a random tool G10.1 block. Tools should be programmed with a Q-word in a G10.1 block or within a range determined by the system installer in AMP. An invalid tool number was entered into the tool management tables or was programmed in a part program block.
INVALID TOOL NUMBER FROM PAL	The PAL offset change feature specified an invalid tool number to the control.
INVALID TOOL ORIENTATION	This is an invalid tool orientation.
INVALID TOOL TABLE TYPE	This is an invalid tool table type.
INVALID VALUE ZONE 3	A zone 3 value was entered that is outside of the zone 3 limits.
INVALID VALUE ZONE 3:	The zone listed has values that are outside of the zone limits.
INVALID VELOCITY FDBK TYPE	AMP for your digital drive system has been configured for an invalid velocity loop hardware type. Valid values for digital systems are NO FEEDBACK, ABSOLUTE FEEDBACK, and INC ENCODER ON DIGITAL MODULE. Other selections are invalid on digital systems.
INVALID WHEEL ANGLE	An invalid wheel angle has been entered for the angled wheel grinder. Wheel angles must be entered between 0 and 180 degrees. Also wheel angles that approach 90 degrees are also invalid.
INVALID WORD IN G10L3 MODE	An attempt was made to assign a parameter that is not a legal parameter in the G10L3 mode. G10L3 assigns data to the tool management tables.
INVALID WORD IN G11 BLOCK	An invalid word was programmed in a G11 block that cancels the data setting mode for the tool management tables. The G11 code must be programmed in a block that contains no other data.
INVALID ZONE LIMIT	This is an invalid Zone Limit.
INVALID ',' WORD	A word other than a chamfering C-word, a radius R-word, or QPP angle word was programmed in a block with a comma ",". Only the radius and chamfer words can be preceded with a "," in a block.
IPD AND G16.3/G16.4 CANNOT BE CONCURRENT	This error message is issued when in-process dressing is on and a block containing a G16.3 or G16.4 is activated on a cylindrical grinder in angled wheel configurations.
J	
JOG WILL CAUSE (+) OVERTRAVEL	An attempt was made to execute an incremental jog that would move the indicated axis beyond its positive software overtravel limit.
JOG WILL CAUSE (-) OVERTRAVEL	An attempt was made to execute an incremental jog that would move the indicated axis beyond its negative software overtravel limit.
JOGGED HOME TOO FAST:	The speed selected for the move to the home limit switch is too fast and the homing operation has failed. Move the axes back to the other side of the limit switch (the side before the homing operation began), and re-execute the homing operation, this time slowing the speed using the <SPEED/MULTIPLY> switch or the <FEEDRATE OVERRIDE> switch.
L	
L VALUE OUT OF RANGE	An L-word repeat count was programmed larger than the system is capable of performing (typically a maximum L of 9999 is permitted). A second block will need to be programmed to duplicate the commands again. Enter a smaller L-word for both blocks.
L-WORD CANNOT BE GREATER THAN TOOL RADIUS	The programmed L-word value in a G88.5 or G88.6 hemispherical pocket cycle is greater than the programmed tool radius. The incremental plunge depth of a hemispherical pocket cycle cannot be greater than the tool radius.
L- WORD OUT OF RANGE	More than 1000 spark-out passes were specified by the L-word in a grinder fixed cycle.

Message	Description
LARGER MEMORY - REFORMAT	This message typically occurs after a new AMP or PAL has just been downloaded to the control. There is now more memory available for the RAM disk, but you need to reformat to use it. If desired, you do not have to reformat RAM and can continue to run the control with the RAM disk at its current size.
LEAD WORD FORMAT FINER THAN	The word format programmed is requesting a finer resolution than the axis word format for the corresponding axis allows. These word formats are set in AMP.
LENGTH OFFSET AXIS MISSING IN PROCESS	You have configured the tool length axis as a shared axis and it is currently not being controlled by the process requesting to activate a tool length offset. The shared length axis must be returned to the process attempting to activate the tool offset. Or tool offsets were programmed for an axis that is configured in AMP as unfitted.
LESS MEMORY - REFORMAT	This message typically occurs after a new AMP or PAL has just been downloaded to the control. There is now less memory available for the RAM disk, and you must reformat to use the RAM disk.
LETTER OR DIGIT MUST FOLLOW \$, %, !, &, OR #	You have used incorrect search string syntax in the PAL search monitor utility.
LETTER OR DIGIT MUST FOLLOW \$, %, ! OR #	You have used incorrect search string syntax in the PAL search monitor utility.
LETTER OR DIGIT MUST FOLLOW \$, % OR !	You have used incorrect search string syntax in the PAL search monitor utility.
LETTER OR DIGIT MUST FOLLOW \$, % OR !	You have used incorrect search string syntax in the PAL search monitor utility.
LETTER OR DIGIT MUST FOLLOW #	You have used incorrect search string syntax in the PAL search monitor utility.
LIMIT EXTRN DECEL SPEED ON	Dual axes have limited the external decel speed AMP value. An axis in the dual-axis group was AMPed with a lower value.
LIMIT MANUAL DLY CONSTNT ON	Dual axes have limited the manual delay constant AMP value. An axis in the dual-axis group was AMPed with a lower value.
LIMIT MAX CUTTING FEED ON	Dual axes have limited the maximum cutting feedrate AMP value. An axis in the dual-axis group was AMPed with a lower value.
LIMITED ACC/DEC RAMP ON	Dual axes have limited the acc/dec ramp AMP value. An axis in the dual-axis group was AMPed with a lower value.
LIMITED RAPID FEEDRATE ON	Dual axes have limited the rapid feedrate AMP value. An axis in the dual-axis group was AMPed with a lower value.
LIMITED VELOCITY STEP ON	If the velocity step AMP value is not the same for all axes of a dual group, the control will adjust them to the limiting axis.
LOW VOLTAGE ON FLASH STICKS	Call Allen-Bradley Support Services.
LOWER > UPPER	A value entered in the programmable zone table for zone 2 or 3 results in a lower limit value being greater than the upper limit. The upper limit must always be greater than the lower limit.
M	
M02 OR M30 FOUND - REQUEST TERMINATED	This error occurs if an M02 or M30 is found before the requested block while searching during a mid-program start. The search will be terminated at the M02/M30 block.
MACHINE HOME REQUIRED OR G28	An attempt was made to program an axis move before the axes were homed. Axes can be homed manually or by programming a G28 block.
MASTER HAS TO BE AMPED FIRST	The dual master axis has to be configured first in the AMP data base.
MASTER ONLY G-CODE - MUST PARK SLAVES	An attempt was made to program a G-code that is not compatible with a dual axes. The programmed G-code can only be applied to the AMP defined master axis of the dual axis group. All other axes in the dual axis group must be parked.
MATH OVERFLOW	Your paramacro or calculator function is requiring a calculation with an excessively large or illegal value.
MAX SIZE EXCEEDED	The programmed number of symbols is too large (the communication data packet is too large).
MAX SOLID TAP RPM EXCEEDS MAX GEAR RPM	The resulting solid tapping RPM exceeds the spindles current RPM Maximum for the active gear range. Either change gear ranges, or reduce the tapping speed.

Message	Description
MAXIMUM BLOCK NUMBER REACHED	A renumber operation was performed to renumber block sequence numbers (N-words), and the control has exceeded a block number of N99999. Either the program is too large to renumber, or the parameters for the first sequence number, or the sequence number increment, are too large. When this error occurs, the renumber operation stops renumbering at the last block within the legal range of N-words.
MAXIMUM NUMBER OF AXES EXCEEDED	If the COCOM breakout is true, a maximum of 4 concurrent interpolated axes can be used.
MAXIMUM NUMBER OF PROGRAMS	The RAM disk directory for part program storage is full. You can store only 328 files on the system even when memory is available for part program storage.
MAXIMUM RETRACE COUNT REACHED	The limit (defined in AMP) for the amount of retrace blocks allowed was reached. No further retracing will be allowed.
MAXIMUM REVERSE PLANES EXCEEDED	The order that the axes are named in AMP is important. If, for example, axis one's name is assigned as X and axis three's name is assigned as Z, a reverse plane is defined if the G18 plane is assigned in AMP as the ZX plane. The G18 plane defines a plane consisting of axis 3 followed by axis 1, making it a reverse plane (axis 1 followed by axis 3 would be a normal plane since 1 is configured before 3 from the standpoint of ODS). This also pertains to parallel axes. A maximum of four reverse planes is allowed. If your system exceeds this number of reverse planes, you must re-configure your AMP.
MAXIMUM RPM LIMIT AUXILIARY SPINDLE 2	A request was made for the aux spindle 2 speed to exceed the AMPed maximum value. Reduce the programmed aux spindle 2 speed, or use the spindle speed override switch to reduce the RPM.
MAXIMUM RPM LIMIT AUXILIARY SPINDLE 3	A request was made for the aux spindle 3 speed to exceed the AMPed maximum value. Reduce the programmed aux spindle 3 speed, or use the spindle speed override switch to reduce the RPM.
MAXIMUM RPM LIMIT FIRST SPINDLE	A request was made for the spindle 1 speed to exceed the AMPed maximum value. Reduce the programmed spindle 1 speed, or use the spindle speed override switch to reduce the RPM.
MAXIMUM RPM LIMIT PRIMARY SPINDLE	A request was made for the primary spindle speed to exceed the AMPed maximum value. Reduce the programmed primary spindle speed, or use the spindle speed override switch to reduce the RPM.
MAXIMUM RPM LIMIT SECOND SPINDLE	A request was made for the spindle 2 speed to exceed the AMPed maximum value. Reduce the programmed spindle 2 speed, or use the spindle speed override switch to reduce the RPM.
MAXIMUM RPM LIMIT THIRD SPINDLE	A request was made for the spindle 3 speed to exceed the AMPed maximum value. Reduce the programmed spindle 3 speed, or use the spindle speed override switch to reduce the RPM.
MESSAGE PENDING, PRESS A KEY TO DISPLAY	The 9/Series screen saver is engaged and a system error message, PAL error message, E-Stop condition, or PAL display page has been activated. Press any key on the keyboard to disable the screen saver and view the error or PAL display page.
MDI INPUT COMMAND TOO LONG	The MDI input command string exceeds the maximum length allowed.
MDI NOT ALLOWED DURING INTERRUPT MACRO	An attempt was made to halt the execution of an interrupt program and execute a MDI command. MDI commands cannot be executed during the execution of an interrupt program.
MDI NOT ALLOWED DURING POCKET MILLING	An MDI command cannot be programmed while a G88 or G89 pocket milling cycle is executing.
MDI NOT ALLOWED DURING RETRACE	You cannot use MDI while a retrace operation is in progress.
MEASUREMENT POINT OVERFLOW	The user tried to enter more points into online AMP for axis calibration than are permitted.
MEMORY CRASH - REFORMAT	A major error has occurred within the system RAM memory. All part programs stored in memory will have to be deleted by performing a reformat operation. This will not remove the current versions of AMP or PAL from the system.
MEMORY FULL	There is no more RAM memory space for part program storage. If you are in the process of editing a part program, your changes cannot be saved.
MIDSTART NOT ALLOWED FROM TAPE	You cannot perform a mid-program start on a program that is stored on tape. The program must first be transferred to RAM memory.

Message	Description
MINIMUM RPM LIMIT AUXILIARY SPINDLE 2	The commanded aux spindle 2 speed requested by the control is less than the AMPed minimum aux spindle 2 speed for the current gear being used. This requires a gear change operation or a change in the programmed aux spindle 2 speed. In some cases, the <SPINDLE SPEED OVERRIDE> switch may be sufficient.
MINIMUM RPM LIMIT AUXILIARY SPINDLE 3	The commanded aux spindle 3 speed requested by the control is less than the AMPed minimum aux spindle 3 speed for the current gear being used. This requires a gear change operation or a change in the programmed aux spindle 3 speed. In some cases, the <SPINDLE SPEED OVERRIDE> switch may be sufficient.
MINIMUM RPM LIMIT FIRST SPINDLE	The commanded spindle 1 speed requested by the control is less than the AMPed minimum spindle 1 speed for the current gear being used. This requires a gear change operation or a change in the programmed spindle 1 speed. In some cases, the <SPINDLE SPEED OVERRIDE> switch may be sufficient.
MINIMUM RPM LIMIT PRIMARY SPINDLE	The commanded primary spindle speed requested by the control is less than the AMPed minimum primary spindle speed for the current gear being used. This requires a gear change operation or a change in the programmed primary spindle speed. In some cases, the <SPINDLE SPEED OVERRIDE> switch may be sufficient.
MINIMUM RPM LIMIT SECOND SPINDLE	The commanded spindle 2 speed requested by the control is less than the AMPed minimum spindle 2 speed for the current gear being used. This requires a gear change operation or a change in the programmed spindle 2 speed. In some cases, the <SPINDLE SPEED OVERRIDE> switch may be sufficient.
MINIMUM RPM LIMIT THIRD SPINDLE	The commanded spindle 3 speed requested by the control is less than the AMPed minimum spindle 3 speed for the current gear being used. This requires a gear change operation or a change in the programmed spindle 3 speed. In some cases, the <SPINDLE SPEED OVERRIDE> switch may be sufficient.
MIRROR NOT ALLOWED ON ROLLOVER AXIS	You cannot perform mirrored motion using an axis with rollover.
MISSING 1394 I/O RING ADDR	This message indicates that a 1394 amplifier ID has been AMPed but not defined in I/O ring assignment from ODS. The 1394 amplifier must be a defined device on the 9/Series fiber optic I/O ring.
MISSING () AFTER FUNCTION NAME	Paramacro and calculator functions must have their values enclosed in [], for example, SIN[5].
MISSING (])	Paramacro and calculator functions must have their values enclosed in [], for example, SIN[5]. The control has found that a right bracket "]" is missing in the current operation.
MISSING (END) COMMAND	The control has found an end-of-program block (M02 or M30) before it has read the END command for a paramacro DO loop.
MISSING (F) IN INVERSE TIME	An F-word must be programmed in every motion block that is not rapid when in inverse time feed mode (G93). F is not modal in G93.
MISSING (GOTO) COMMAND	An IF paramacro condition does not have a GOTO with a sequence number following the condition.
MISSING A (DO) COMMAND	A WHILE paramacro condition does not have a DO with a loop identifier following the condition.
MISSING ADAPTIVE FEED DATA	An attempt was made to create a transfer line part program from the quick view screen with incomplete adaptive feedrate data.
MISSING COMMA	An error occurred in G05 DH+ communications block.
MISSING COMMA OR RIGHT PARENTHESIS	An error occurred in G05 DH+ communications block.
MISSING CUTTER COMP CODE	Cutter compensation must be activated before initiating a G89 irregular pocket cycle.
MISSING DATA FROM BLOCK	G89 irregular pocket cycle parameters are missing from a the G89 programming block.
MISSING END PARENTHESIS	An error occurred in G05 DH+ communications block.
MISSING G67	An active modal macro (G66 or G66.1) was not canceled by a G67 before the control read an M02 or M30 end-of-program command.
MISSING HPG FROM I/O RING	The I/O assignment file that was compiled and downloaded with PAL defines an HPG that is not physically present in the I/O ring. Verify that the HPG address settings are correct.

Message	Description
MISSING I/O RING DEVICE	The I/O assignment file that was compiled and downloaded with PAL defines an I/O ring device that is not physically present in the I/O ring. Verify that all device address settings are correct.
MISSING INTEGRAND/RADIUS WORD	A circular or helical block has been programmed with axis data and no radius (R) or integrand (I, J, or K) values. A radius or integrand must be programmed in a circular or helical block to define the location of the arc center.
MISSING KEYBOARD AND HPG FROM I/O RING	The I/O assignment file that was compiled and downloaded with PAL defines a keyboard and an HPG that is not physically present in the I/O ring. Also verify that the keyboard and HPG address settings are correct.
MISSING KEYBOARD FROM I/O RING	The I/O assignment file that was compiled and downloaded with PAL defines a keyboard that is not physically present in the I/O ring. Verify that the keyboard address settings are correct.
MISSING L-WORD	The L-word parameter is missing from the G88.5 or G88.6 hemispherical pocket programming block.
MISSING M02 OR M30	The control has executed through to the last block of a program and has not read an end-of-program command (M02 or M30).
MISSING MASTER AXIS NAME	Slave axes that do not have a master have been configured for a dual group.
MISSING OR ILLEGAL L-VALUE	An attempt was made to program an irregular pocket milling cycle (G89) with a missing or illegal L-word.
MISSING PROGRAM NAME	An operation, such as a copy or rename, was performed without the proper program names being specified. The proper format consists of the program performing the operation followed by a comma and the target program (OLD PROGRAM NAME,NEW PROGRAM NAME).
MISSING PROGRAM NUMBER (P)	No sub-program name was specified in a block that calls a sub-program or paramacro. A sub-program name must be programmed with a P-word in the calling block.
MISSING PROMPT DATA	The control is waiting for data to be entered on the input line (line 2 of the CRT) using the keys on the operator panel.
MISSING Q-WORD	The Q-word parameter is missing from the G88 or G89 programming block.
MISSING QPP ANGLE WORD	The second block of a two block QPP set does not contain the necessary angle word to define an intersection with the first block.
MISSING ROUGHING CYCLE (P/Q) WORD	A roughing cycle block was programmed that does not contain both a starting and ending sequence number for the contour blocks as programmed with the P- and Q-words.
MISSING ROUGHING CYCLE DEPTH (D) WORD	A roughing cycle block was programmed that does not contain the D parameter for depth of cut.
MISSING SHADOW RAM	Either your 9/290 control is missing the SIMMS necessary for shadow RAM, or your 9/260 control is not equipped with enough RAM to operate properly. If your 9/260 system contains both the DH+ module and the search monitor utility, additional RAM must be installed. All 9/290 controls must have this additional RAM. Refer to your 9/Series installation and maintenance manual for details on installing SIMMS.
MISSING SLAVE INCREMENTAL AXIS NAME	When using dual axes on Lathe A, all slave axes must have incremental axis names.
MISSING START PARENTHESIS	An error occurred in G05 DH+ communications blocks.
MISSING TOOL ENTRY	This is missing a tool entry.
MODULE(S) WITH INCONSISTENT REVISION LEVEL	Retry the update utility. If this does not work, call Allen-Bradley and request a new update utility that matches your hardware revision level.
MODULE(S) WITH INVALID CHECKSUM	Retry update.
MOTION IN DWELL BLOCK	An attempt was made to program axis motion in the same block that generates a dwell. No axis words can be programmed in a block that generates a dwell.
MOTION NOT ALLOWED	The block includes G-codes that must be programmed in a block without axis motion. For example, the G-codes that convert from inch to metric or metric to inch cannot have axis motion in the same block.
MOTOR SHAFT - LEAD SCREW RATIO TOO HIGH	The motor shaft to lead screw gear ratio is too high to achieve the rapid speed assigned in AMP.

Message	Description
MULTIPLE FUNCTIONS NOT ALLOWED	Multiple functions are not allowed.
MULTIPLE SPINDLE CONFIGURATION ERROR	Each multiple spindle must have a servo board identified in AMP to indicate to which board the spindle is connected. The spindle must be included in the number-of-motors AMP parameter for the board the spindle is on.
MUST ASSIGN TOOL NUMBER FIRST	In random tool, an attempt was made to customize a tool before the tool number was assigned.
MUST BE IN (AUTO)	It is necessary to place the control in auto mode to perform the requested operation.
MUST BE IN (AUTO) OR (MDI)	It is necessary to place the control in Auto or MDI mode to perform the requested operation.
MUST BE IN (CYCLE STOP)	It is necessary to place the control in cycle stop state to perform the requested operation. The control cannot be in cycle suspend, feed hold, or E-Stop.
MUST BE IN (CYCLE STOP) AND (EOB)	The control must be in cycle stop state and at the end-of-program block to perform the requested operation. The control cannot be executing a program, in cycle suspend, feed hold, or E-Stop.
MUST BE IN (E-STOP)	An attempt was made to perform an operation (such as, editing the reversal error parameters in online AMP) that must be performed in E-Stop. Place the control in E-Stop by pressing the <E-STOP> button.
MUST BE IN (LINEAR MODE)	An attempt was made to perform an operation (such as, exiting from cutter compensation) that must be performed in a linear block (G00 or G01).
MUST BE IN (MANUAL)	It is necessary to place the control in manual mode to perform the requested operation.
MUST BE IN (MDI)	It is necessary to place the control in MDI mode to perform the requested operation.
MUST BE IN E-STOP OR CYCLE STOPPED	It is necessary to place the control in E-Stop or cycle stop state to perform the requested operation. Place the control in E-Stop by pressing the <E-STOP> button. Place the control in cycle stop state by pressing the <SINGLE BLOCK> button. Simply pressing <CYCLE STOP> will not guarantee the control to be in cycle stop mode. Most likely a cycle stop request while executing a program will place the control in cycle suspend mode. If you get this error using the CALC function it indicates you may be asking the calculator function to access a paramacro variable (using the # sign) when a program is executing. You can not use a paramacro variable # sign in a calculator function when any part program is executing or suspended.
MUST BE IN MANUAL MODE TO HOME	To do a jog home operation (from jog retract) the control must be in manual mode.
MUST COMPLETE ACTIVE HOME OPERATION	An attempt was made to jog a dual group when one of the axes of the dual was homing.
MUST DISABLE RUN-TIME GRAPHICS	An attempt was made to call up one of the QuickView prompting options while the active graphics option was currently executing. Active graphics must be disabled before QuickView prompting can be performed.
MUST HOME ANGLE SOURCE AXIS FIRST	Before you can enter angled wheel grinding mode both the axial and wheel axes must be homed.
MUST HOME AXIS	An attempt was made to perform axis calibration before the axes were homed. Axes can be homed manually or by programming a G28 block.
MUST SETUP THE ENCRYPTION ARRAY	An attempt was made to encrypt a part program while uploading it to ODS or the mini-DNC package. The encryption array must be set up before you can encrypt a part program.
MUST START WITH \$, %, !, #, +, -, LTR, DIGIT	You have used incorrect search string syntax in the PAL search monitor utility.
MUST START WITH \$, %, !, #, OR LETTER	You have used incorrect search string syntax in the PAL search monitor utility.
MUST START WITH \$, %, !, OR LETTER	You have used incorrect search string syntax in the PAL search monitor utility.
MUST START WITH \$, !, OR LETTER	You have used incorrect search string syntax in the PAL search monitor utility.
MUST START WITH A LETTER	You have used incorrect search string syntax in the PAL search monitor utility.
MUST SWITCH PROCESS FOR SCREEN RESTORE	An attempt was made to 'restore screen' while the system was in Digitize, Graphics, Search, or while PAL was selecting a process. Any attempt to switch processes results in this message.

Message	Description
N	
NEED SHADOW RAM FOR ONLINE SEARCH	Your system contains the DH+ module and you have not installed the extra RAM SIMMS that are required to run the PAL online search monitor with the DH+ module installed. You must buy additional RAM for a system equipped with both of these features. Contact your Allen-Bradley Sales representative to purchase these SIMMS. Refer to your 9Series integration manual for details on installing additional SIMMS.
NEED SPINDLE FEEDBACK	You attempted to use the solid-tapping feature or synchronization feature with a spindle that does not have feedback.
NEGATIVE DWELL VALUE	An attempt was made to execute a dwell with a negative value. Dwell values must be positive values.
NEGATIVE F-WORD PROGRAMMED	A negative feedrate was programmed in a program block. Negative feedrates are illegal.
NEGATIVE TO UNSIGNED LONG ERROR	Internal math error has occurred; contact Allen-Bradley customer support services.
NEGATIVE VALUE NOT ALLOWED	The minus (-) sign was used for an address which cannot be programmed with a negative value.
NET CORRECTION IS NOT ZERO	For a rotary axis, the net amount of correction for axis calibration should be zero for one complete revolution.
NET PICK/PLUNGE AWAY FROM ENDPOINT	The primary and secondary pick/plunge amounts, when added together, are in the direction away from the programmed endpoint.
NETWORK COMMUNICATION DISABLED	When editing or restoring communications configuration data, this message is displayed.
NETWORK PASSTHRU COMMUNICATIONS FAULT	A communication error has occurred between the controls ethernet module and the ODS passthrough device (typically a PLC).
NEVER OPENED THE PROGRAM	An attempt was made to edit a program that was not currently open.
NEW TOOL OFFSET SETUP BUT NOT ACTIVATED	The tool offsets for the active tool were changed, but not activated in the current block. These new tool offsets will not be activated until the set-up blocks are cleared of the old tool offsets and refilled with the new tool offsets.
NO ACTIVE PROCESS CONFIGURED	The AMP has been loaded into a multi-processing system that has no processes configured as active.
NO ACTIVE PROGRAM	An attempt was made to do a search when no part program is active.
NO AXIS CONFIGURED	The current active version of AMP does not have any axes configured as usable. All axes are configured as unfitted.
NO CHARACTERS ENTERED FOR SYMBOL	You have used incorrect search string syntax in the PAL search monitor utility.
NO DEPTH PROBE TRIP	A G26 block reached its programmed endpoint without the adaptive depth probe being tripped. The part surface was not detected by the adaptive depth probe before the G26 block completed.
NO FEEDRATE PROGRAMMED	A command for axis motion was executed when there was no active feedrate. Applies to non-rapid moves (G01, G02, or G03).
NO FURTHER RETRACE ALLOWED	The control has reached a block during retrace from which no further retrace is permitted.
NO INTERSECTION EXISTS	There is no mathematical intersection for the QPP blocks as programmed.
NO MARKER FOUND ON :	The encoder marker was not detected when homing the indicated axis. Homing was unsuccessful.
NO MORE MDI BLOCKS	Cycle start was requested during MDI mode when there were no MDI blocks present to be executed.
NO MORE MDI BLOCKS TO RESET	A reset was requested during MDI mode when there were no incomplete or unexecuted MDI blocks reset.
NO OFFSET ACTIVE	An offset must be active before the control will allow the offset to be changed. This check is used so that the control will no the method and direction of the offset will be the same as the previous offset.
NO OPTIONAL FB PORT ON ANALOG SERVO	The system was incorrectly AMPed with optional feedback module on an analog servo module.

Message	Description
NO PROGRAM TO RESTART	There is no program to restart. The previous program was either completed or cancelled.
NO RECIPROCATION DISTANCE	A reciprocation interval of zero (0) was programmed for a grinder reciprocation fixed cycle.
NO RECIPROCATION FEEDRATE	The reciprocation feedrate, E-word, required during a grinder reciprocation fixed cycle was not programmed.
NO SPINDLE ASSIGNED TO THIS PROCESS	A process attempted to activate virtual or cylindrical mode and that process has no spindle assigned to it via AMP.
NO STRING INPUT	A program search operation was requested and no string to search for has been entered. Key in the required search string, and press the [TRANSMIT] key to enter a search string.
NO TOOL GROUP PROGRAMMED	A block that loads data into the tool management table does not contain the parameter that determines the tool group number corresponding to the other data in the block. The group number is programmed using the P-word.
NO TOOL NUMBER PROGRAMMED	A block that loads data into the tool management table does not contain the parameter that determines the tool number corresponding to the other data in the block. The tool number is programmed using the T-word.
NO UNEXPIRED TOOL AVAILABLE	A request for a tool group was made, and all of the tools in that tool management group have expired their tool lives. Either reset the tool life for the tools, or install new tooling.
NON_CONSECUTIVE/TOO MANY FITTED AXES	More than the allowable number of axes may have been assigned in AMP or an unfitted axis was assigned between two fitted axes. You can assign only fitted axes consecutively in AMP.
NON-LINEAR AXIS IN PLANE DEFINITION	The current axis plane is illegal because a non-linear axis (rotary) has been assigned to the plane in AMP.
NOT ALIGNED	During the power-up alignment procedure, either the 1326 motor (connected to a 9/440HR) is misaligned or the SINCOS device's memory is corrupt.
NOT ALLOWED IN ANGLED WHEEL MODE	The axes can not be moving when you change to angled wheel mode. Also the axes involved in angled wheel motion must be homed before you can enter angled wheel mode. Other features, such as block retrace or jog retract also must not be active when changing mode.
NOT ALLOWED - G41/G42 ACTIVE	An attempt was made to perform some operation or program some feature that cannot be performed when cutter compensation or TTRC is active (G41 or G42). Cancel compensation by programming a G40 block before performing the operation.
NOT ALLOWED FROM MDI	Certain programming commands are not allowed from MDI (GOTO, WHILE, etc.).
NOT ALLOWED ON DUAL/SLAVE AXIS	A G26 was programmed on a dualed axis. The G26 feature is incompatible with the dual axis feature.
NOT ALLOWED - THREADING ACTIVE	An attempt was made to perform some operation (typically a spindle speed adjustment) that is not allowed when cutting a thread. This includes all forms of threading, including single pass or multiple pass threads.
NOT IN G10L3 MODE	A G11 block was programmed that cancels G10L3 data setting when the control is not in the G10L3 data setting mode. G10L3 is used to set the tool management table data.
NUMBER IS OUT OF RANGE	An attempt was made to perform a calculation using the paramacro features or the calculator features that contains a number longer than 11 characters.
NUMBER OF MOTORS/SPINDLE CONFIG ERROR	This is error indicates AMP is incorrectly configured for the 9/Series hardware. Typical AMP configuration errors that generate this error include: You have AMPed more motors than the current hardware supports. You have indicated there are servo motors attached to servo boards that don't exist (the 9/230 and 9/440 are configured as if they have only one servo card). You have configured too many spindles (1 on 9/230, 2 on 9/260 and 9/440, 3 on 9/290). Too few axes were configured for the indicated number of motors on the boards or too few servos were configured for indicated number of motors on the boards.
NUMERIC VALUE MISSING	The numeric value associated with the programmed word is missing. There is an AMP parameter that determines whether a missing numeric is assumed to be zero or if it will generate this error.

Message	Description
0	
OBJECT NOT FOUND IN PROGRAM	The object you are searching for in the search monitor utility does not exist in the current module, or does not exist in the program in the direction you are searching.
OCI ETHERNET CARD NOT INSTALLED	An OCI dual-process system has a standard CRT installed. The OCI Ethernet card has not been installed. This may happen if a dual-process OCI executive is loaded into a non-OCI system.
OCI SYSTEM ERROR	VRTX error. Contact Allen Bradley Support.
OCI PROCESSING TASK OVERLAP	The amount of time to process a new OCI request is taking longer than expected. This is an informational warning only. It is not critical to the CNC.
OCI WATCH LIST TASK OVERLAP	This message indicates that the watch list task was not running to completion in the AMPed allotted amount of time. This typically occurs when a large task is requested by an OCI station and the CNC takes longer than expected to complete.
ODS & 9/SERIES REVISIONS DIFFER	The version of AMP or PAL on the peripheral device does not match the control version.
ODS RUNG MONITOR ACTIVE	The online PAL search monitor utility can not be accessed. The online PAL search monitor utility requires the offline ODS PAL search monitor utility to not be running.
OFFSET EXCEEDS MAX CHANGE	You have attempted to modify an offset table by an amount that is larger than the allowable change to an offset table. Refer to your AMP reference manual for details on Maximum wear and geometry offset change.
OFFSET EXCEEDS MAX VALUE	You have attempted to modify an offset table by entering an offset amount that is larger than the allowable maximum offset selected in AMP. Refer to your AMP reference manual for details on Maximum offset table values.
OFFSET MOTION PENDING ON CYCLE START	After changing the active offset this message identifies that the control will move the axis to the new offset location the next time cycle start is pressed (this may or may not occur on a non-motion block depending on the AMP offset configuration).
OFFSET TABLE(S) CORRUPT/CLEARED	A bad offset table checksum value was detected by the control during PTO.
ONLY ONE DEPTH PROBE PER SERVO BOARD	The 9/Series servo card firmware only supports one adaptive depth probe on each servo card. If your system requires more than one adaptive depth probe they must be attached to different servo cards (9/230 and 9/440 controls can only have one adaptive depth probe). AMP must be configured to indicate which port the adaptive depth probe is attached to.
ONLY ONE M19 ALLOWED PER BLOCK	For system configured with multiple spindles, only one spindle orient M-code (M19) is allowed per block.
ONLY REQUEST THE DUAL MASTER FOR JOGS	An attempt was made to jog a slave axis; you can jog a slave axis only when the master axis is parked.
OPTION NOT INSTALLED	An attempt was made to program an optional feature that has not been purchased from Allen-Bradley.
OPTION NOT INSTALLED (PAL DISPLAY PAGE)	The PAL display page option is not installed on your control.
OPTIONAL FEATURE IS NOT PROVIDED	An attempt was made to program an optional feature that has not been purchased from Allen-Bradley.
OPTIONAL RAM SIMM BAD/MISSING	The control has discovered the RAM SIMMs for the extended storage option are either damaged or missing. The RAM SIMMs must be installed or replaced. Contact your Allen Bradley sales representative for assistance.
OTHER PROCESS G CODE CONFLICT	On a dual processing system, one process has a conflicting G code active when you attempted to activate a G26 depth probe cycle. For example, process one executes a G26 while process two has an axis in feed to hard stop which is on the same servo card as the depth probe.
OVER SPEED	A servo motor is turning at an RPM that is greater than the maximum RPM allowed for that servo as defined in AMP by the system installer. For digital spindles this error can result from maximum RPM gear range 1 being set higher than your AMPed allowed Maximum Motor Speed.

Message	Description
OVER SPEED IN POCKET CYCLE	The programmed feedrate for an irregular pocket cycle (G89) was too high for the cycle to keep up. The part program stops at the endpoint of the block in which the error occurred. The cycle must be executed with a lower feedrate.
OVERTRAVEL (+)	The indicated axis has reached the positive software overtravel limit during an axis jog. This message can appear prior to reaching the overtravel limit in certain instances. For example, if a single pulse from the handwheel will result in a large incremental move beyond the overtravel limit, this error message will appear before the axis moves up to the limit.
OVERTRAVEL (-)	The indicated axis has reached the negative software overtravel limit during an axis jog. See OVERTRAVEL (+) for details.
P	
P VALUE OUT OF RANGE	An attempt was made to call a macro or sub-program using a program number, following the P-word, that is out of the valid range. Valid range for a P-word is 1 to 99999.
PAL & 9/SERIES REVISIONS DIFFER	Either the overall revision number of PAL does not match the software revision on the control, or the revision number of system symbols in PAL and the revision number of those on the control do not match.
PAL ANALOG PORT ILLEGAL CONFIGURATION	This is a power turn-on error that occurs when an AMP configuration error exists in the PAL analog port configuration.
PAL ANALOG PORT/SERVO F-W INCOMPATIBLE	PAL-controlled analog output port feature requires the servo firmware (f-w) revisions: Analog servo f-w rev 0.06 or greater Digital servo f-w re. 2.03 or greater Consult Allen-Bradley customer support services about servo firmware updates.
PAL AXIS STATUS CANNOT CHANGE	You attempted to change the status of a PAL axis (either to PAL axis or to a system axis) when it is not allowed. Examples of when the transition is not allowed are when the axis is jogging, performing jog retract, performing block retrace, etc...
PAL BACKGROUND TOOK TOO LONG	Background PAL was not completed in the time allocated to it in AMP. Background PAL will continue on to completion before restarting. If and when background PAL does complete in the allocated time, this message will disappear. If this message appears continuously, the PAL program should be rewritten, or else the AMP defined background PAL execution time should be increased. Refer to the AMP and PAL reference manuals for more details.
PAL DIVIDE BY ZERO ERROR	The PAL program tried to divide a value by zero. Check the PAL program for errors.
PAL DOES NOT EXIST	There is no PAL program in the system, either on EPROM or in RAM memory. EPROMs must be installed, or else PAL must be downloaded to RAM from ODS.
PAL INITIATED MOTION POSSIBLE	While in QuickCheck mode it is possible for PAL to physically move axes. This includes any motion generated by PAL including the PAL axis mover, or jogs that can occur in automatic mode such as jog on the fly or manual gap elimination. This message is cleared after the first block is executed in QuickCheck mode.
PAL OVERWRITING G54 • G59.3	PAL is overwriting the current G54 - G59.3 offset through PAL offsets.
PAL PAGE WAITING - EXIT DISPLAY SELECT	A PAL display page is being overwritten by the current screen. Pressing the {DISPLY SELECT} softkey will display the display page.
PAL PAGE WAITING - EXIT MONITOR	A PAL display page is being overwritten by the current screen. Exit the search monitor utility to see the screen PAL is attempting to display.
PAL PAGE WAITING - SCREEN HAS PROMPT	A PAL display page is being overwritten by the current screen.
PAL PROM CHECKSUM ERROR	Checksum error in the PAL PROM memory. This indicates PAL has been loaded successfully however it has failed to pass verification. Check if your flash sticks are installed properly and are not damaged. Attempt to download a copy of the same PAL image from another project.
PAL SOURCE NOT DOWNLOADED TO CNC	The PAL search monitor utility can not be accessed. The PAL search monitor utility requires the PAL source code be downloaded with the built PAL program.
PAL SOURCE NOT LOADED	The copy of PAL in flash does not contain source programs.

Message	Description
PAL SOURCE REV. MISMATCH - CAN'T MONITOR	PAL source code in the control does not match the revision of the CNC executive. The PAL code may execute if all of the PAL system flags exist but the monitor cannot be used.
PAL USING MEMORY - REFORMAT	The AMP parameter allowing PAL to be stored in RAM memory has been enabled. This changes the amount of RAM memory available for part program storage, requiring the RAM disk to be reformatted. Part programs should have been backed up prior to this.
PARAMETER ASSIGNMENT SYNTAX ERROR	A block that assigns Paramacro parameters has been entered incorrectly.
PARAMETER NUMBER NOT FOUND	The AMP parameter number being searched for through the control's patch AMP utility does not exist in the system.
PARAMETER VALUE OUT OF RANGE	The value entered for the selected AMP parameter or paramacro parameter is less than or greater than the allowed legal value.
PARENTHESIS INPUT ERROR	Parentheses have been entered incorrectly in a program block or calculation operation. Correct the use of the parenthesis; verify they are in matched pairs.
PARITY ERROR IN PROGRAM	A serial communications error has occurred. A data parity error occurred while sending or receiving data. This can result in a corrupted file, or the entire data transfer operation may be aborted by the control.
PARK AXIS MOTION NOT ALLOWED	Axis motion was programmed for a parked axis in a dual axis group. When both master and slave axes are parked, no axis motion is allowed on a parked axis in a dual group.
PART PROGRAM NOT SELECTED	An attempt was made to execute a program or check a program before a program was selected for execution.
PART ROTATION FORMAT ERROR	In part rotation blocks (G68, G69), only plane changes and mode changes including inch/metric and absolute/incremental are permitted. Any commands other than normal motion commands and the motion G-codes (G00, G01, G02, and G03) are not permitted.
PASSWORD PROTECTED	When assigning password protectable features to an access level, an attempt was made to assign a feature to a different access level when the currently active password does not have access to the feature. You can assign features to other access levels only when you have access to that feature yourself.
PEAK CURRENT NOT 300%	The axis for a 1394 or 9/440 is not AMPed to have the PEAK CURRENT set to 300%. This misconfiguration forces the control into E-Stop.
PERIPHERAL DEVICE ERROR	An illegal communication attempt was made with a peripheral device, for example, attempting to output to a tape reader or input from a tape punch.
PLANE SELECT ERROR	An attempt was made to change planes during cutter compensation (TTRC), between QPP blocks, or between chamfer and corner rounding blocks. This error also will occur if G17 or G19 planes are selected on a lathe.
PLEASE WAIT FOR CLEARING OF PAL MEMORY	PAL is being erased in preparation for a PAL download.
PLUNGE MOTION NOT ALLOWED	The final plunge position must be different from the start point of the cycle. This message can occur if the plunge axis is not programmed in the entry block to G89 mode, or if the plunge axis increment is zero, or if the final plunge axis position is the same as the start point of the cycle block during G89 mode.
PLUNGE MOTION NOT PROGRAMMED	In your pocket cycle you have either not programmed a final depth, or the final depth you have programmed is equal to the depth of the cutting tool at the starting point of the cycle. The location of the cutting tool when the pocket cycle is programmed must be at a different depth than the final programmed depth of the cycle.
PLUNGE NOT ALLOWED	A plunge that will cut into the pocket wall was requested in a G89 irregular pocket cycle.
PLUNGE STEPS MIS-PROGRAMMED	The rough, medium, and fine-feed depths in the cycle block are not programmed correctly. This is possible if the data in the block is incorrect or if the data in the modal values of the parameter not programmed in the block are incorrect.
POCKET END NOT SAME AS START	A pocket end-point that is not the same as the pocket start-point was programmed in a G89 irregular pocket cycle.

Message	Description
POCKET IS PART OF CUSTOM TOOL	An attempt was made to assign a tool to a tool pocket that is already used by a custom tool. Custom tools are assigned to tool pockets that are shown with an XXXX next to the pocket number on the random tool table.
POCKET MILLING SHAPE IS INVALID	A parameter is missing in the G88 programming block.
POINT ALREADY EXISTS	The point that you are trying to enter is already in the axis calibration table.
PORT B IS BUSY	This message appears when you press {SYSTEM SUPPORT}, {MONITOR}, or {SERIAL I/O} and port B is busy.
PORT IS BUSY - REQUEST DENIED	An attempt was made to output or input information to or from a serial communications port that is already being used by some other device or is selected as the port that an active program is coming from.
PREVIOUS ABORT COMMAND NOT COMPLETE	This message is displayed when the communications "abort" key is entered before the last abort requested has completed.
PROBE/CONTROLLING AXIS CARD DIFFERENT	Both the adaptive depth probe and the adaptive depth probe controlling axis (typically the axis that positions the probe) must be attached to the same servo card. You must re-AMP your system and re-arrange your servo wiring so that the adaptive depth probe and its corresponding servo are on the same servo card.
PROBE CYCLES CALCULATION ERROR	The servo module was unable to compute the probe position when the probe is fired. Make sure that all measurement points are within the programmed range entered for the probe cycle. Lower the feedrate during the probing operation and try again.
PROBE CYCLES PROGRAMMING ERROR	Either not enough or too many axes are programmed in a probing cycle block.
PROBE ERROR	A probing cycle has reached the outer limits of the tolerance band without firing the probe, or the probe has fired before entering the tolerance band.
PROBE IN USE BY OTHER PROCESS	On a dual processing control only one probing function is allowed at any one time. Probing can not be performed by both processes simultaneously. You must wait for probing to complete in one process before probing in the other process.
PROBE IS ARMED, CAN'T ADJUST SERVOS	With the probe armed through a probing operation, until the probe fires or the probe is disarmed, other online AMP servo parameters like torque, feedforward percentage, gain, etc., are not allowed to be changed.
PROBE TRIP DURING DECEL	An adaptive depth probe trip occurred after the program block reached endpoint. The trip was made while the control was waiting for the following error to collapse after interpolation is complete. Avoid this error by reducing axis speed (thus reducing following error) or by moving the adaptive depth block endpoint further into the part.
PROCESS SWITCH NOT CURRENTLY ALLOWED	On a dual-processing system, you cannot switch processes while in graphics or in digitize.
PROGRAM ACTIVE	An attempt has been made to delete or perform some other operation to a program that was activated for automatic execution. The program must be deactivated using the {CANCEL PROGRAM} softkey.
PROGRAM ACTIVE IN ANOTHER PROCESS	This dual lathe error appears when one process attempts to open a file for edit, deletion, etc., while that file is active in another process.
PROGRAM BEING EDITED	An attempt has been made to copy, verify, or perform some other operation on a program that is still in the edit mode. It is necessary to press the {EXIT EDITOR} softkey from the edit menu to properly end an editing operation.
PROGRAM BLOCK TOO LONG	More than 128 characters were entered into a single block.
PROGRAM CURRENTLY IN USE	A subprogram or paramacro program was called that is currently being used to perform some other operation (such as editing or copying). Typically, this message is the result of attempting to edit a program that was not properly closed. A program remains in the edit mode until the {EXIT EDITOR} softkey is pressed from the program edit menu.
PROGRAM NAME TOO LONG	An attempt was made to create a program with a program name longer than 8 alphanumeric characters. If a large, descriptive program name is desired, a comment may be added to the right of the program name using the {PROGRAM COMMENT} feature.

Message	Description
PROGRAM NOT FOUND	The program cannot be located in memory. Check to make sure the program name was correctly entered.
PROGRAM OPEN FOR EDIT IN ANOTHER PROCESS	On a dual-processing system, you cannot edit a program that is active in another process. You will need to switch processes if you want to edit the other program.
PROGRAM REWIND ERROR	An attempt to rewind the tape was not successful. Check to be sure that the tape reader is functioning properly and the tape is on the drive sprockets.
PROGRAM SHOULD START HERE	When performing a {MID ST PRGRAM} operation to restart a program, the control has found the block that the program execution should begin at, and selected that block as the next block to be executed. That block is the block immediately following the one containing an @.
PROGRAMMED AXIS IS OFF OR DETACHED	Part program blocks are attempting to program motions on an axis that has its servos either off or configured as detached in AMP.
PROGRAMMED G26 DEPTH < TRIGGER TOLERANCE	A G26 block is programmed with an integrand less than or equal to the AMPed Adaptive Depth Trigger Tolerance amount. A block decode error is given and the block will not execute until the integrand in the block is made larger or AMP is modified to reduce the trigger tolerance.
PROGRAMMED SPINDLE UNAVAILABLE	The programmer attempted to program the follower spindle independently (M03, M04, M05, or M19) while spindle synchronization was active.
PROGRAMS ARE DIFFERENT	A program verify operation has determined that the two selected programs are not identical.
PROGRAMS ARE IDENTICAL	A program verify operation has determined that the two selected programs are identical matches.
PROGRMABLE ZONE 2 VIOLATION	An attempt was made to move the indicated axis into the area defined by programmable zone 2.
PROGRMABLE ZONE 3 VIOLATION	An attempt was made to move the indicated axis into (or out of) the area defined by programmable zone 3.
PROGRMD G26 DEPTH < TRIGGER TOLERANCE	The programmed adaptive depth deflection (hole depth) is less than the probe tolerance value. You must either increase the programmed block depth, or decrease the AMPed probe tolerance value.
Q	
QPP ANGLE WORD SAME AS AXIS NAME	AMP has downloaded an angle word for QuickPath Plus that is the same as an axis name. AMP must be reconfigured; the angle word cannot be the same as an axis name.
QPP BLOCK FORMAT ERROR	Data is incorrectly entered or insufficient data is entered for the control to correctly execute a QuickPath Plus block or pair of QuickPath Plus blocks.
QPP MDI BLOCK LOOKAHEAD ERROR	Only one of two necessary blocks was programmed in MDI using QuickPath Plus commands that require two blocks for proper execution.
QPP NOT ALLOWED DURING POLAR MODE	With polar coordinate programming active, you cannot use QPP.
R	
R WORD FORMAT FINER THAN	The word format programmed is requesting a finer resolution than the axis word format for the corresponding axis allows. These word formats are set in AMP.
RAPID SPEED TOO HIGH FOR AMPED CONFIG	AMP configuration error. The axis resolutions and feedback device resolutions will not permit the rapid and maximum feedrates assigned in AMP.
RADIUS TOO SMALL	An arc (or helix) was programmed (G02 or G03) that defines a radius that is too small to connect the start-point of the arc to the end-point. The value of R is too small.
RAPID TOO HIGH FOR AMPED CONFIG	AMP configuration error. The axis and feedback device resolutions will not permit the rapid feedrates assigned in AMP.
RAPID TRAVERSE ERROR :	An attempt was made to jog an axis using rapid traverse when it is not permitted. Typically, to use the TRVRS function while jogging, the control must be in manual mode; continuous jog must be selected; and, if the axis being jogged has an overtravel value, that axis must first have been homed.
READ ERROR	An attempt to read a program from a tape or disk drive has failed.

Message	Description
RECIP AXIS IN WRONG PLANE	The reciprocation axis specified in a G81 or a G81.1 programming block is not in the currently selected plane.
RECIP AXIS NOT PROGRAMMED	No reciprocation axis was specified in a G81 or a G81.1 programming block.
RECIPROCATION NOT STOPPED	An attempt was made to deactivate the current part program while reciprocation is still active. You must deactivate reciprocation before deactivating the current part program.
REMOTE I/O COMMON RAM FAULT ON RESET	The RIO module tests the common RAM after reset and detects an error. The Interboard Communications Fault LED is turned ON.
REMOTE I/O CTC CHIP TEST FAULT	The RIO module tests the CTC chip after reset and detects a fault. The Processor Fault LED is turned ON.
REMOTE I/O DENIED COMMON ACCESS ON RESET	The RIO module was denied access to CRAM for more than 1 second after reset. The Interboard Communications Fault LED is turned ON.
REMOTE I/O EPROM INTEGRITY FAULT	The checksum test over the RIO program area in the EPROM chip found a fault. The Processor Fault LED is turned ON.
REMOTE I/O INCORRECT USER BT DATA AMOUNT	The RIO module attempted to read a block of data from one of the user output block transfer data buffers in common RAM and found the word count of the data to be outside of the range of 1 to 64. The Interboard Communications Fault LED is turned ON.
REMOTE I/O INITIALIZATION ERROR	Remote I/O hardware or network has failed to initialize. Cycle power to try to restart or check remote I/O hardware (9/290 only).
REMOTE I/O INTERNAL RAM FAULT	The RIO module tests its internal RAM chip after reset and during operation. A fault has been detected. The Processor Fault LED is turned ON.
REMOTE I/O INTERRUPT HARDWARE FAULT	The RIO module detects that its CPU was not interrupted by any expected external interrupts. This condition indicates a problem in recognizing interrupts. The Processor Fault LED is turned ON.
REMOTE I/O INVALID RACK ADDRESS SET UP	The RIO module's rack address is illegal. This fault is the result of the user setting the rack address, via the dip switches, to an invalid rack size and/or starting module group number.
REMOTE I/O INVALID USER BT DATA CHECKSUM	The 16-bit 2's complement checksum calculated by the RIO module using data from a user output block transfer data buffer does not match the checksum placed in the buffer by the user device. The Interboard Communications Fault LED is turned ON.
REMOTE I/O INVALID USER DATA CHECKSUM	The 16-bit 2's complement checksum calculated by the RIO module using data from the user output data table in common RAM does not match the checksum placed by the user in the user output data table in common RAM. The Interboard Communications Fault LED is turned ON.
REMOTE I/O MISSING USER OPERATIONAL CODE	The RIO module did not detect the user operational code after reset. This fault is displayed when the RIO module does not detect the user operational code in the user status register in common RAM within 100ms after the RIO module has set its operational code and released control of common RAM back to the user device. The Interboard Communications Fault LED is turned ON.
REMOTE I/O RIO DENIED COMMON RAM ACCESS	The RIO module was denied access to CRAM for longer than the specified interval. The RIO module failed to gain access to common RAM after attempting for the Accessing Time-out time period. The time-out is due to either the user device maintaining access for more than the Accessing Time-out interval. or to a hardware failure. The Interboard Communications Fault LED is turned ON.
REMOTE I/O SERIAL COMMUNICATIONS FAULT	The RIO module cannot communicate with the PLC processor. Either the PLC processor's power is OFF, the blue hose is not connected, or the PLC processor is in Edit mode.
REMOTE I/O SIO CHIP TEST FAULT	The RIO module tests the SIO chip after reset and detects a fault. The processor fault LED is turned ON.
REMOTE I/O UNABLE TO FIND BT DATA BUFFER	The RIO module was unable to detect the user block transfer data buffer. The interboard communications fault LED is turned ON.
REMOTE I/O UNRECOVERABLE ERROR	Remote I/O hardware or network has catastrophic failure. Cycle power to try to restart or check remote I/O hardware (9/920 only).

Message	Description
REMOTE I/O USER FAULT OCCURRED	The RIO module detected that the user fault bit was set. The interboard communications fault LED is flashing.
REMOTE I/O WATCHDOG TIMEOUT	The watchdog mechanism on the RIO module timed out, indicating that the RIO module has not operated in an expected manner for possibly 17ms. The processor fault LED is turned ON.
REMOTE IO INTERPROCESSOR HANDSHAKE FAULT	The RIO module failed to detect the complement of the user-handshake word, in the complement user-handshake word in common RAM, within the handshake interval. The user device has not shook hands with the RIO module. The interboard communications fault LED is flashing.
REPLACE ABSOLUTE FB BATTERY	The battery that attaches to the servo module and supplies power for the absolute encoders is under-voltage and must be replaced.
REPLACE MEMORY BACKUP BATTERY	The battery that attaches to the main processor board and supplies power for the control's RAM memory is under-voltage and must be replaced. If not replaced, AMP data cannot be copied to backup memory and part program data may be lost.
REQUESTED DATA TOO LARGE	The data you are trying to send or receive is too large.
REQUIRES AT LEAST TWO AXES	A transfer line quick view prompt was selected for a cycle which requires two or more axes. Your system is currently configured as a single axis system.
RESETTING E-STOP	Once you push the E-Stop Reset button to clear the E-Stop state, the Resetting E-Stop message displays to alert you that the control is attempting to come out of E-Stop. After the system is out of E-Stop and the drives are enabled, the control clears this message. If the error condition is not cleared, this message clears, but the "E-STOP" message continues to flash as the control remains in E-Stop state.
RETRACE NOT ALLOWED	A retrace is not allowed from the point in program execution.
RIGHT OPERAND MUST BE POSITIVE	The right operand of a logical operator must be a positive value. Negative values are illegal; for example, 1AND-2 is illegal because of the -2.
RING I/O RECOVERY DISABLED	This message is activated when the {RECVRY DISABL} softkey is pressed on the I/O RING MONITOR page. It is a visual indicator that the operator has disabled the I/O Ring retry functionality.
RIO COMMON RAM ACCESS NOT ACKNOWLEDGED	The control's request to use the RIO module was denied. The RIO module lost power, or the control was restarted, but the RIO module was not.
ROLLOVER/OVERTRAVEL INCOMPATIBLE	Overtravel limits were specified in AMP for an axis that is configured as a rollover axis. Rollover axes do not have overtravel limits.
ROTARY AXIS CANNOT BE SCALED	A rotary axis cannot be scaled.
ROTARY WORD OUT OF RANGE	A rotary axis was programmed to move to an absolute position that is greater than or equal to 360 degrees. In absolute mode, a rotary word must range between 0 and 360 degrees.
ROUGHING CYCLE NESTING ERROR	The contour blocks called by a roughing cycle to define the finished contour of a part contain a block that likewise calls for a roughing cycle. Contour blocks for a roughing cycle cannot contain a block that likewise calls for a roughing cycle.
ROUGHING CYCLE PROGRAMMING ERROR	A syntax error has been found in a roughing routine block (G72, G73, G74, or G75).
RUNG NUMBER NOT FOUND	The rung number you are searching for in the search monitor utility does not exist in the current module, or does not exist in the program in the direction you are searching.
S	
S-CURVE ACC/DEC CONFIGURATION ERROR	An axis configuration error was detected by the control when the programmed acc/dec ramp was out of range. An attempt to program an acceleration ramp value of 0 in a G48.3 or G48.4 block. An attempt was made to program another G-code in a block with a G48.x.
S-CURVE MIN PROG JERK TOO SMALL	An attempt was made to select a jerk value below the allowable AMPed value.
S-CURVE MODE NOT ALLOWED	This message displays when an attempt was made to use a feature that is illegal in S-Curve Acc/Dec mode. The following can not be used with S-Curve Acc/Dec: 7300 Series Tape Compatibility, PAL Axis Mover, Circular Interpolation Mode (G02, G03), Feed to Hard Stop (G24), jogging, threading, and solid tapping.

Message	Description
S-CURVE OPTION NOT INSTALLED	An attempt was made to select S-Curve Acc/Dec (G47.1) when the S-Curve option bit was set to false. Make sure your system includes the S-Curve option.
S NOT LEGAL PROGRAMMING AXIS NAME	This is displayed at power-up when the letter "S" is assigned to linear or rotary axis. Only the spindle(s) can be AMPed with "S" as the name; it cannot be assigned to a programmable axis.
S OVER SPEED	A servo motor is turning at an RPM that is greater than the maximum RPM allowed for that servo as defined in AMP by the system installer. For digital spindles this error can result from maximum RPM gear range 1 being set higher than your AMPed allowed Maximum Motor Speed.
SAVE COMPLETED	The changes made to the current device set-up have been saved.
SCALE FACTORS MUST BE EQUAL FOR PLANE	When performing circular motion or motion in certain cycles, keep the scale factors for the axes of the active plane equal.
SCALING INVALID DURING POLAR	Scaling cannot be used during polar programming.
SEARCH ALREADY IN PROGRESS	You cannot request a search operation while one is currently running. Complete or abort the current search before attempting another search.
SEARCH MONITOR SELECT NOT ALLOWED	You can not use the online PAL search monitor utility while the display select function softkeys are active. Leave the display select screens (press DISP SELECT) before you try to access the search monitor utility.
SEARCH REQUIRES AN ACTIVE PROGRAM	An attempt has been made to perform a search operation when no program was selected for execution. A program must be selected for automatic execution before a program search can be performed.
SEARCH STRING NOT FOUND	The character or character string designated in the search operation was not found.
SECOND SPINDLE NOT CONFIGURED	For spindle 2 to be programmable, it must be configured in AMP; a decode error.
SECONDARY AUX. WORD SAME AS AXIS NAME	The secondary auxiliary word (usually B) is the same as an axis name, causing an interpretation conflict for the control. This word and all axis names are assigned in AMP.
SEE (MESSAGE) IN PROGRAM BLOCK	The programmer has assigned a system parameter that generates this message, telling the operator to read the comment in the current part program block. Program execution will resume when cycle start is pressed.
SEQUENCE NUMBER OUT OF RANGE	A sequence number beyond the range of 1 - 99999 was programmed.
SEQUENCE STOP NUMBER FOUND	A sequence stop number has been activated, and that sequence number has been found in the currently executing program. Execution will stop after the block containing the sequence number corresponding to the sequence stop number is executed. Execution will resume when cycle start is pressed.
SERIAL COMMUNICATIONS BUFFER OVERFLOW	A peripheral device communication error (such as a tape reader). The 512 character input (receive) buffer has overflowed. Data may have been lost. Check your configured communications protocol (flow control) and check for proper cabling/pin connections.
SERIAL COMMUNICATIONS ERROR #1	This is an internal software error. The control is unable to access DF1 Driver.
SERIAL COMMUNICATIONS ERROR #2	This is an internal software error. Check cables and try again.
SERIAL COMMUNICATIONS ERROR #3	This is an internal software error. This is an unknown DF1 Driver error.
SERIAL COMMUNICATIONS ERROR #4	This is an internal software error. The control is unable to access the serial communications port.; check cables and try again.
SERIAL COMMUNICATIONS ERROR #5	Serial communications port has not received the expected response in the time allowed.
SERIAL COMMUNICATIONS FRAMING ERROR	An incorrect number of bits was encountered during a read operation. Check your device setup.
SERIAL COMMUNICATIONS PARITY ERROR	Incorrect parity of data was received. Check your device setup.
SERIAL PORT IN USE	This message will appear if a serial communications port is busy when checked prior to transmission.
SERIAL UART BUFFER OVERFLOW	The 2 character buffer on the UART receiver has overflowed. A character has been lost. Check communications setup.

Message	Description
SERVO AMP C LOOP GAIN ERROR	One of the following AMP parameter errors exist: Current Prop. Gain + Current Integral Gain < 4096 or Current Prop. Gain - Current Integral Gain > 0.
SERVO AMP ERROR	There is an error in one or more of the AMP parameters relative to servo control or an absolute feedback encoder failed to initialize.
SERVO AMP FDBK PORT ERROR	The feedback port assignments in AMP are wrong; for example, two servos are using the same feedback port on the same servo module.
SERVO AMP FE LIMITS CORRECT	One or more of the following AMP parameters were changed to satisfy the following equation: Inposition Band • Gain Break Point • Feedrate Suppression • Excess Error The servo module would have disabled control operation if these parameters were not changed.
SERVO AMP ID SPEED CORRECT	One or more of the following AMP parameters were changed to satisfy the following equation: $o/ \leq \text{Motor speed at starting Id} \leq \text{Motor speed at Id Break Point} \leq \text{Max. Motor Speed}$ The servo module would have disabled control operation if these parameters were not changed.
SERVO AMP OUTPUT PORT ERROR	The output ports as assigned in AMP are wrong; for example, two servos on the same board are assigned to the same output port.
SERVO AMP V LOOP GAIN ERROR	One of the following AMP parameter errors exist: Velocity Prop. Gain + Velocity Integral Gain < 65536 or Velocity Prop. Gain - Velocity Integral Gain > 0
SERVO AMP, AMP TYPE ERROR	The AMP parameters specifying amplifier types and connectors are contradictory.
SERVO AMPLIFIER FAULT	This indicates that a fault signal has been received from a servo amplifier. It can usually be corrected by turning off power to the amplifier, and then back on.
SERVO BUSY DURING HOMING OPERATION	This error indicates that the servo processor was unable to respond during a homing operation. It can occur under the unusual condition resulting from two or more servo axes reaching their home point simultaneously. Generally, the axes can be re-homed with no problems.
SERVO CONFIGURATION ERROR	The AMP servo configuration is inconsistent. An example of this error would be if the downloaded AMP file were configured for only two axes, when the AMP parameter "Number of Motors on First Board" was set for three.
SERVO COMMUNICATIONS ERROR	A communications error occurred between the control and the servo module.
SERVO CURRENT LOOP ERROR	While running an axis, the allowable current loop proportional error or current loop integral error has gone out of range.
SERVO INTERFACE FAILURE	The servo interface diagnostics performed on power-up have failed. Attempt to power up again. If the error remains, contact Allen-Bradley customer support services.
SERVO POS & VEL FB SIGN ERR	This is a power turn-on error which occurs when the signs of the position and velocity feedback devices do not match when a common feedback port is used for both.
SERVO POWER UP SEQUENCE ERROR	The servo processor diagnostics performed on power-up have failed. Attempt to power up again. If the error remains, contact Allen-Bradley customer support services.
SERVO POWERUP DIAGNOSTICS FAILURE	The servo module diagnostics performed on power-up have failed. Possible causes include incorrect servo AMP parameters being downloaded. An example would be configuring AMP for five axes when there is only one servo module installed.
SERVO PROCESSOR ASSIGNMENT ERROR	Too many servos were AMPed or a servo was assigned to a non-existent servo processor. The system is held in E-Stop. The message indicates an error in the total number of fitted axes and spindles, or in the AMPed values of: Number of Motors on 1st board Number of Motors on 2nd board.

Message	Description
SERVO PROCESSOR OVERLAP	The analog version of the servo sub-system provides fine iteration overlap detection. This message is displayed if the fine iteration software on the DSP does not execute to completion in one fine iteration.
SERVO PROM CHECKSUM ERROR	The checksum test on the servo processor software stored in PROM memory has failed. This test is performed on power-up and periodically while the system is running. Contact Allen-Bradley customer support services.
SERVO PTO DIAGNOSTICS FAIL	The servo card has failed its power-up diagnostics. Consult Allen-Bradley customer support services.
SERVO PTO SEQUENCE ERROR	The servo card has failed its power-up diagnostics. Consult Allen-Bradley customer support services.
SERVO TIME-OUT READING ABSOLUTE ENCODER	During power-up initialization of the position registers or during a homing operation, the servo processor has failed to return a read within the required time after the absolute position has been requested by the main processor. Consult Allen-Bradley customer support services.
SERVO TIME-OUT READING FEEDBACK	During a homing operation, if there is an error reading feedback from the servo module, this message appears. This usually occurs when the system scan time is close to the threshold at which PAL execution can just complete and when homing more than 3 axes at a time. This error can be avoided by homing axes individually or increasing the system scan time in AMP.
SET ZERO NOT ALLOWED ON:	A set zero operation on the specified axis is not permitted. Typically this is because either the control is not in manual mode, or the selected axis is in the process of being jogged.
SHAFT VALUE > NUMBER OF POCKETS	An attempt was made to assign a shaft pocket that is greater than the number of pockets assigned for that custom tool. The shaft pocket number must be a value between 1 and the number of pockets assigned to that tool.
SHARED AXIS CONFIGURATION ERROR	Either there are too many shared axes configured, a shared axis has the same name as some other axis in the system, the diameter axes on a lathe are shared axes, or some other miscellaneous configuration error occurred.
SHARED AXIS NOT IN PROCESS	You have attempted to position a shared axis (or recouple a shared dual axis) not currently available to the requesting process. A shared axis can only be positioned by the process currently controlling the shared axis.
SHARED SPINDLE CONTENTION	This is a run-time decode error. A process attempted to activate an exclusive-use spindle mode or change the spindle speed when another process was using it. The process goes into cycle stop.
SHIFT AWAY FROM ENDPOINT	When a cylindrical grinder cycle (G84 or G85) is programmed with a shift and plunge, and the shift increment does not move towards the cycle endpoint, this message is generated. The shift increment must move towards the cycle endpoint.
SHIFT VALUE HAS TOO MANY DIGITS	You have used incorrect search string syntax in the PAL search monitor utility.
SKIPPING SOURCE NOT INCLUDED MODULE(S)	When you downloaded your PAL program the source code for some modules was not included. The ODS software can decide to not include the source on selected modules when it determines there is not sufficient memory on the control to hold both the PAL image and the source code. The PAL search monitor utility will not monitor any PAL modules that do not have their source code downloaded.
SLASH NOT ALLOWED	An error occurred in G05 DH+ communications block.
SLAVE AXIS LETTER CANNOT BE PROGRAMMED	An attempt was made, when using dual axes, to program the slave's axis letter.
SPINDLE CONFIGURATION ERROR	An attempt was made to configure a spindle that did not have a servo board identified in AMP to indicate to which board the spindle is connected. The spindle must be included in the number-of-motors AMP parameter for the board the spindle is on.
SPINDLE ERROR, AMP FIRST SPINDLE 1ST	AMP order of spindles must be spindle 1, spindle 2, spindle 3.
SPINDLE ERROR, AMP SECOND SPINDLE 2ND	AMP order of spindles must be spindle 1, spindle 2, spindle 3.
SPINDLE ERROR, AMP THIRD SPINDLE 3RD	AMP order of spindles must be spindle 1, spindle 2, spindle 3.

Message	Description
SPINDLE IS CLAMPED	An attempt was made to program a block containing a spindle code other than an M05 while the PAL servo clamp request flag for the spindle was set.
SPINDLE MODES INCOMPATIBLE	An attempt was made to enter virtual mode when the spindle that is used for this mode is synchronized as the follower spindle or an attempt was made to perform end face milling during synchronization.
SPINDLE MOTOR SPEED TOO HIGH	When using a 1326 motor as a spindle, feedback resolution combined with your configured maximum spindle speed would return feedback counts faster than the control can reliably decode. Either reduce the maximum configured spindle speed, or reduce the configured feedback counts for the spindle in AMP.
SPINDLE MUST BE THE LAST SERVO	When the system is AMPed, the spindle must be assigned to the first available port after all axes have been assigned.
SPINDLE NOT ASSIGNED	A spindle axis was AMPed, but not assigned to any process.
SPINDLE ORDER ERROR, AMP AUX. 2 SECOND	AMP order of spindles must be primary spindle, aux. spindle 2, aux. spindle 3.
SPINDLE ORDER ERROR, AMP AUX. 3 THIRD	AMP order of spindles must be primary spindle, aux. spindle 2, aux. spindle 3.
SPINDLE ORDER ERROR, AMP PRIMARY 1ST	AMP order of spindles must be primary spindle, aux. spindle 2, aux. spindle 3.
SPINDLE SYNC NOT CONFIGURED	The programmer attempted to enter synchronized spindle mode before it was configured in AMP.
SPINDLE SYNC UNAVAILABLE THIS PROCESS	An attempt was made to enter synchronized spindle mode on a dual-process control when the process was not yet configured for both spindles in the synchronized pair.
SQUARE ROOT OF NEGATIVE ERROR	Internal math error has occurred; contact Allen-Bradley customer support services.
SQUARE ROOT OF NEGATIVE INVALID	An attempt was made to determine the square root of a negative number using the calculator or through a paramacro SQRT command.
STORED PASSWORD LIST TO BACKUP	This message appears after the password list has been successfully stored to the control's backup memory.
STORING TO BACKUP - PLEASE WAIT	This message appears whenever AMP or axis calibration data in RAM is being stored in backup memory.
SYMBOL NAME FORMAT ERROR	Check the remote symbol and CNC symbol to make sure they exist on both remote and CNC. Check the table of the read only or write only variables.
SYMBOL NOT FOUND	Check the remote symbol and CNC symbol to make sure they exist on both remote and CNC. Check the table of the read only or write only variables.
SYNCHRONIZATION DEADLOCK	A synchronization code is activated and caused the activating process to wait on a process that is already waiting.
SYNCH SPINDLES MISCONFIGURED	Causes for this could be: only one spindle (either controlling or follower) was defined in the synchronized spindle pair, you exceeded the simple feedback ratio limitation of 10 (e.g., 11:1 or 2:13), or on a multiprocess system, one (or both) of the spindles in the synch pair is currently not available to the process making the synchronization request.
SYNCH SPINDLES REQUIRE FEEDBACK	One or both of the spindles, configured in AMP as a member of a synchronized pair, did not have feedback. Both spindles in a synchronized pair must be equipped with an AMP configured feedback device.
SYNTAX ERROR (COMMA)	A missing comma or an extra comma was found in the program block.
SYSTEM DIAGNOSTIC #1	An illegal parameter was passed into a switch statement (mid-program start) in the control software. Contact Allen-Bradley customer support services.
SYSTEM DIAGNOSTIC #2	An illegal parameter was passed into a switch statement (ASCII buffer task) in the control software. Contact Allen-Bradley customer support services.
SYSTEM DIAGNOSTIC #3	An illegal parameter was passed into a switch statement (ASCII buffer task) in the control software. Contact Allen-Bradley customer support services.

Message	Description
SYSTEM MODULE GROUND FAULT	The 1394 system module has detected a ground fault. The system generates a ground fault when there is an imbalance in the DC bus of greater than 5A. This drive error can be caused by incorrect wiring (verify motor and ground wiring), motor malfunction, or an axis module IGBT malfunction.
SYSTEM MODULE OVER TEMP	The 1394 contains a thermal sensor which senses the internal ambient temperature. Causes could be: that the cabinet ambient temperature is above rating. The machine duty cycle requires an RMS current exceeding the continuous rating of the controller. The airflow access to the 1394 is limited or blocked. This does not necessarily indicate a motor over temperature. Motor over temperature sensors should be wired directly into the E-Stop string.
SYSTEM MODULE OVER VOLTAGE	The 1394 system module buss voltage exceeds the maximum operating voltage. The dc power bus is continuously monitored. If it exceeds a preset level (810V dc), a fault is sensed and the power supply is disabled. There are several possible causes for this error. <ul style="list-style-type: none"> • an undersized shunt requirement • a blown shunt regulator fuse • a malfunctioning shunt regulator transistor • the power driver board is malfunctioning and incorrectly sensing the bus voltage • an incorrectly set CNC acc/dec rate • an excessive input line voltage • the system inertia is too high causing excessive energy to be returned to the power supply bus • a vertical axis with insufficient counterbalancing is overdriving the servomotor and causing excessive energy to be returned to the power supply bus • an incorrect power supply is installed in your system. Make sure you are using a <u>CNC</u> power supply
SYSTEM MODULE PHASE LOSS	The 1394 system module has detected a loss of one of the input power phases. The three-phase input line is monitored and a fault will be issued when a phase loss is detected. Typical causes include, one or more input line fuses have opened, contactor malfunction, or incorrect wiring.
SYSTEM MODULE UNDER VOLTAGE	The 1394 system module voltage does not meet the minimum operating voltage. The DC power buss shall activate the under voltage limit when the bus drops to 275 Vdc or less. It will clear at 300 Vdc. Typical causes include low voltage on the three phase input.
T	
(T) WORD IN CIRCULAR MODE	An attempt was made to activate a tool length offset in a block that generates a circular move. Tool length offsets can be activated only in linear blocks (or in non-motion blocks if AMP is so configured).
T-WORD NOT ALLOWED WITH M06	NEXT TOOL IN T WORD was selected as the tool-change type in AMP while a T-word is programmed in an M06 block.
TAN CIRCLE NOT IN 1ST BLOCK	When editing a program, an attempt was made to digitize an arc using {CIRCLE TANGNT} as the first block in the program. To use this digitizing format, the control must first have a tool path programmed to make the arc tangent.
TEMPLATE PROGRAM NOT FOUND	A transfer line quick view item was selected without the correct part program template present in the protected directory. There are 19 transfer line cycles and there must be part program templates QV01 thru QV19 present in the protected directory. Refer to your T-LINE-9 Quick Start guide for details on replacing/restoring these part program templates.
THIRD SPINDLE NOT AVAILABLE	AMP configuration error; spindle 3 can be configured only on a 9/290.
THIRD SPINDLE NOT CONFIGURED	For spindle 3 to be programmable, it must be configured in AMP; a decode error.
THRDS/IN WORD FORMAT FINER THAN	The word format programmed is requesting a finer resolution than the axis word format for the corresponding axis allows. These word formats are set in AMP.
THREAD FEEDRATE TOO LARGE	The lead is too large in threading mode. Program slower spindle speed.
THREAD LEAD ERROR	The thread lead was too large or too small. This commonly occurs when cutting a variable thread lead and before the end of the threading pass is reached. Either the lead goes to zero for a decreasing lead thread, or an axis speed would exceed its maximum allowable cutting feedrate when cutting an increasing lead thread.

Message	Description
THREAD LEAD IS ZERO	No thread lead has been programmed in a block that calls for thread cutting. Thread lead is programmed with either an F- or an E-word.
THREAD PULLOUT DISTANCE TOO LARGE	The programmed threading pullout distance is larger than the programmed distance of the thread departure.
THREAD PULLOUT STOPPED AT I-PLANE	The chamfer block of a threading cycle is shortened so that the combination of pullout angle and pullout distance does not cause the retract in axis 1 to go beyond the I-plane. The AMP pullout angle is still used for the chamfer.
THREADING DISTANCE IS ZERO	A threading cycle has been programmed with no thread. Program an end-point or an end-point different from the start-point.
TIME-OUT OCCURRED WHILE WAITING FOR INPUT	When downloading AMP or PAL from the ODS workstation to the control, the message OKAY TO DOWNLOAD? (Y/N): appears on the control screen. If you do not respond within an allowed time, this error will appear.
TIMER MUST START WITH #	You have used incorrect search string syntax in the PAL search monitor utility.
TOO MANY () IN EXPRESSION	The control has found an unmatched number of [] in a program block or calculator operation. All left brackets “[” must have a corresponding right bracket “]”.
TOO MANY () IN EXPRESSION	The control has found an unmatched number of [] in a program block or calculator operation. All right brackets “]” must have a corresponding left bracket “[”.
TOO MANY 7300 PATTERNS IN MEMORY	An attempt was made to enter a 7300 pattern into the control's memory when the internal cross-reference table of pattern repeat names was full. The internal cross-reference table of pattern repeat names can only hold 20 pattern repeat names.
TOO MANY ACTIVE PROCESSES CONFIGURED	An AMP has been loaded that has too many actively configured processes for this controller model. The 260 series and the dual lathe can have only 2 active processes.
TOO MANY AXES AMPED FOR HARDWARE	An AMP has been loaded that has too many configured axes for this controller model. The 9/440 series can have only up to 6 axes.
TOO MANY AXES PROGRAMMED	Too many axis letters were programmed in a fixed cycle block.
TOO MANY AXES SELECTED FOR DISPLAY	When using the {AXIS SELECT} softkey, you can display only 6 axes. If you attempt to display more than 6 axes, this message is displayed.
TOO MANY CODES IN SYNCH BLOCK	Synch codes must be in a block by themselves, except for an N- or O-word. (9/260-9/290 dual lathe only)
TOO MANY DECIMAL POINTS	A word or parameter value has been programmed with two or more decimal points.
TOO MANY DEVICES ON I/O RING	The I/O ring cannot support the number of devices that has been connected.
TOO MANY EXPRESSION NESTS	The maximum number of nested expressions is 25; for example, [P3+[P4+[P5]]] has 3 expressions nests.
TOO MANY G67'S	A G67 cancel modal paramacro code was executed when no modal paramacro was active. This is typically caused when there are fewer nested modal paramacros than the programmer expected.
TOO MANY I-J-K SETS	An attempt was made to define a local paramacro parameter that is greater than #33 using I,J,K, argument sets. A maximum of 10 different I, J, K, sets may be programmed for each set of local parameters.
TOO MANY MACRO CALLS	The maximum number of nested paramacros was reached. Only 4 paramacros can be active at any one time.
TOO MANY MOTORS AMPED ON 1ST BOARD	The AMP parameter for the number of motors on the first servo board is larger than the number of axes in the system.
TOO MANY NESTED (DO) COMMANDS	More than the allowable number paramacro DO loops are active at one time. A maximum of 3 nested DO loops are allowed.
TOO MANY NONMOTION BLOCKS-DEADLOCK	There were too many non-motion blocks encountered during the look-ahead for cutter compensation or QPP. Consult Allen-Bradley customer support services.

Message	Description
TOO MANY NONMOTION CHAMFER/RADIUS BLOCKS	Too many non-motion blocks separate the first tool path that determines the chamfer or radius size (programmed with a ,R or ,C) from the second tool path. A maximum number of non-motion blocks is set in AMP by the system installer. A non-motion block is defined as any block that does not generate axis motion in the current plane.
TOO MANY POCKETS IN ROUGHING CYCLE	A maximum of 2 pockets can exist in a roughing cycle.
TOO MANY QPP NONMOTION BLOCKS	Too many non-motion blocks separate the first and second tool paths with unknown intersections in QuickPath Plus. A maximum number of non-motion blocks is set in AMP by the system installer. A non-motion block is defined as any block that does not generate axis motion in the current plane.
TOO MANY SHARED SPINDLES	Too many spindles were specified as being shared by two or more processes.
TOO MANY SPINDLES	More than one spindle is configured on the control.
TOO MANY SUBPROGRAM CALLS	The maximum number of nested sub-programs was reached. Only 4 sub-programs may be active at any one time.
TOOL CONFIGURATION WILL NOT FIT	When assigning a custom tool in the random tool table, the number of pockets assigned to the tool relative to the position of the selected shaft pocket will conflict with a different tool already assigned to a pocket. If the custom tool is to be assigned as entered, it must be assigned to a different shaft pocket, or the tool that conflicts with the custom tools location must be moved.
TOOL ENTRY EXCEEDS LIMIT	The selected tool number entered is greater than the AMP'ed maximum tool number entered by the system installer.
TOOL GROUP DOES NOT EXIST	An attempt was made to edit a tool group in the tool life management tables that does not yet exist in the tool directory. A group must be created by using the {TOOL DIR} softkey options.
TOOL OFFSET CHANGES NOT ALLOWED	During certain cycles, G10 tool change operations are not allowed.
TOOL OFFSET REQUIRES MOTION BLOCK	A tool offset cannot be changed in a non-motion block. A non-motion block is any block that does not generate axis motion in the current plane.
TOOL RADIUS TOO LARGE	The programmed tool radius in a G88 or G89 pocket cycle is too large for the pocket contour. A smaller radius tool must be used to machine out the current pocket contour.
TOOL RADIUS TOO SMALL FOR POCKET SIZE	The programmed tool radius in a G88 or G89 pocket cycle is too small for the pocket contour. Either select a larger tool for the pocket contour or reduce the amount of material to be removed each rough cut of the cycle.
TOP OF PROGRAM REACHED	When performing one of the program search operations, the first block in the program has been reached.
TRAVERSE NOT ALLOWED ON :	An attempt was made to move an axis at rapid traverse before it was been homed. This only applies to axes that have software overtravel limits.
TYPE 1 INTERRUPT INCOMPATIBLE WITH G24	This message occurs when returning from a type 1 program interrupt that previously interrupted a G24 block. The interrupt is allowed however the return move is invalid since the axis was previously in the G24 mode. You must manually intervene to continue program execution. We recommend switching to a type 2 program interrupt.
U	
UART PORT IS ALREADY OPEN	The requested serial communications port has already been opened. This message will appear if an attempt is made to send data to a port that is currently being used.
UNABLE TO OPEN PROGRAM	The control cannot find the program that is requested. Make sure the program name is entered correctly or the peripheral device has the correct programs loaded in it.
UNABLE TO OPEN THE UART PORT	A serial communication port error has occurred; retry. The conditions that can lead to this error are unusual and generally will not exist when a second attempt is made to open the port. If this error is generated continuously, it indicates that there may be a communications port hardware failure.

Message	Description
UNABLE TO SYNCH IN CURRENT MODE	The control can not perform the request to synchronize spindles. Possible causes are: synchronization is already active; virtual/cylindrical programming or a threading operation is active on the primary or follower spindle when the synchronization request is made; or on a dual-process system, one of the requesting processes cannot gain control over both spindles.
UNABLE TO WRITE TO FLASH MEMORY	If flash SIMMs appear to be installed correctly, remove and reseat SIMMs. If problem persists, contact Allen-Bradley support service.
UNDEFINED INTERRUPT MACRO/SUBPROG	An interrupt program request was received by the control, but it cannot find the paramacro or sub-program with the corresponding program name in the program directory. The program name is defined in the enable block (M96) with a P-word.
UNEXPECTED DEPTH PROBE TRIP	G26 adaptive depth probe has fired unexpectedly. Either it has fired in a non-G26 block or it has fired before the programmed G26 contact range.
UNSPECIFIED NETWORK ERROR	An error is being sent from another device that the module cannot interpret.
UNUSABLE WORDS IN ZONE BLOCK	An axis word or other data was programmed in a programmable zone block (G22, G22.1 G23, G23.1). These G-codes must be programmed in blocks containing no other data except a block delete /, N word, or comments.
UNRECOVERABLE ERROR	Can occur when updating flash SIMMs with new 9/Series firmware. Retry the update utility. If problem persists, call Allen-Bradley Support Services.
V	
VEL LOOP INVALID WITH DAC OUT	An attempt was made to select the position/velocity servo loop type on a 9/440HR system.
VIRTUAL AXIS NOT ALLOWED	The virtual axis can only be programmed when the control is in a virtual axis mode. You must place the control in G16.3 mode to program a virtual axis.
VIRTUAL C NEEDS SPINDLE WITH FDBK	When the spindle is the virtual C axis in a virtual C application, it must be configured to provide feedback to the servo module.
VIRTUAL/REAL AXIS NAME CONFLICT	The axis configured in AMP as the Virtual C axis was previously configured as a linear machine axis.
W	
WARNING - G10 OFFSETS ALTERED	This message warns that the offsets were changed by a G10 block during execution from a mid-program start.
WARNING - PROGRAM STARTING AT BEGINNING	An active program was edited and then the editor exited. This causes the active program to restart at the beginning of the program.
WARNING - VERIFY MODAL CODES	The MID START PROGRAM feature that activates modal codes for mid-program execution is requesting that these generated modal codes be checked before program execution is started. These modal codes can be checked on the G- and M-code status screens.
WARNING - WATCHDOG JUMPER IS INSTALLED	This error indicates that the watchdog has been bypassed on the 9/Series hardware and your system will not report watchdog errors. Call Allen-Bradley field service.
WHEEL AXIS MOTION INVALID IN G16.3/G16.4	While in the angled wheel grinding mode you have attempted to program the wheel axis directly. Only the virtual axis and the axial axis can be programmed in angled wheel mode.
WILDCARD MUST BE AT START/END OF SYMBOL	You have used incorrect search string syntax in the PAL search monitor utility.
WORK CO-ORD CHANGES NOT ALLOWED	You have attempted to make a change to the work coordinate system at an invalid time. Changes to the work coordinate system can not be performed when some features are active. Disable the offending feature before attempting to change coordinate systems.

Message	Description
Z	
Z-WORD CANNOT BE GREATER THAN R-WORD	The depth (Z-word) of a pocket formed using a G88.5 and G88.6 hemispherical pocket cycle cannot be greater than the radius (R-word) of that pocket.
ZONE 2 PROGRAM ERROR	The next block in the program or MDI entry would cause the specified axis to enter the restricted area of programmable zone 2.
ZONE 2 PROGRAM ERROR:	The current block in the program or MDI entry caused the specified axis to enter the restricted area of programmable zone 2.
ZONE 3 PROGRAM ERROR	The next block in the program or MDI entry would cause the specified axis to enter or exit the area defined as programmable zone 3.
ZONE 3 PROGRAM ERROR:	The current block in the program or MDI entry caused the specified axis to enter the restricted area of programmable zone 3.

END OF APPENDIX

G-code Tables

Appendix Overview

This appendix lists the G-codes for 9/Series Mill controls. They are listed numerically along with a brief description of their use. These G-codes are discussed in detail in the sections within this manual that refer to their specific usage.

The group numbers given in the table refer to modality. Group 00 are not modal and are independent of other G-codes. The remaining G-code groups are modal with other G-codes with the same group number. This means programming a G-code in group 1 replaces any other active group 1 G-code but does not affect any G-codes in the other group numbers.

G-code Tables

A	Modal Group	Function	Type
G00	01	Rapid Positioning	Modal
G01		Linear Interpolation	
G02		Circular/Helical Interpolation (Clockwise)	
G03		Circular/Helical Interpolation (Counterclockwise)	
G04	00	Dwell	Non-Modal
G05		Send Command and Wait for Return Status (for 9/SERIES Data Highway Communication Module)	
G05.1-G05.4		Send Command without Waiting for Return Status (for 9/SERIES Data Highway Communication Module)	
G09		Exact Stop	
G10L2		Setup Work Coordinate Offset Tables	
G10L3		Setup Tool Management Table	
G10L10		Setup Tool Length Values Geometry Table	
G10L11		Setup Tool Length Values Wear Table	
G10L12		Setup Tool Radius Values Geometry Table	
G10L13		Setup Tool Radius Wear Values Wear Table	
G10.1		Setup Random Tool Table	
G10.2L1		Communication Configuration Information (for 9/SERIES Data Highway Communication Module)	
G10.2L2		Output Command Table (for 9/SERIES Data Highway Communication Module)	
G10.2L3		Download Configuration Information (for 9/SERIES Data Highway Communication Module)	
G11		Setup Tool Management Table (Cancel)	

A	Modal Group	Function	Type		
G12.1	21	Primary Spindle Controlling	Modal		
G12.2		Auxiliary Spindle 2 Controlling			
G12.3		Auxiliary Spindle 3 Controlling			
G13		QuickPath Plus (Use First Intersect.)			
G13.1		QuickPath Plus (Use Second Intersect.)			
G14	19	Scaling (Disable)	Modal		
G14.1		Scaling (Enable)			
G15	15	Polar Coordinate Programming (Cancel)	Modal		
G16		Polar Coordinate Programming			
G16.1		Cylindrical Interpolation			
G17	02	Plane Selection	Modal		
G18		Plane Selection			
G19		Plane Selection			
G20	06	Inch System Selection	Modal		
G21		Metric System Selection			
G22	04	Programmable Zone 2 and 3, ON	Modal		
G22.1		Programmable Zone 3, ON			
G23		Programmable Zone 2 and 3, OFF			
G23.1		Programmable Zone 3, OFF			
G24	00	Feed to Hard Stop	Non-Modal		
G25		Adaptive Feedrate (torque mode)			
G26		Adaptive Depth			
G27		Machine Home Return Check	Non-Modal		
G28		Automatic Machine Home			
G29		Automatic Return From Machine Home			
G30		Return to Secondary Home			
G31		External Skip Function 1			
G31.1		External Skip Function 1			
G31.2		External Skip Function 2			
G31.3		External Skip Function 3			
G31.4		External Skip Function 4			
G36		22		Short Block Acc/Dec Check (Enable)	Modal
G36.1				Short Block Acc/Dec Check (Disable)	
G37	00	Tool Gauging Skip, Function 1	Non-Modal		
G37.1		Tool Gauging Skip, Function 1			
G37.2		Tool Gauging Skip, Function 2			
G37.3		Tool Gauging Skip, Function 3			
G37.4		Tool Gauging Skip, Function 4			
G38		Circle Diameter and Center Measurement			
G38.1		Parallel Probing Cycle			

A	Modal Group	Function	Type
G39	20	Cutter Diameter Comp (Linear Generated Block)	Modal
G39.1		Cutter Diameter Comp (Circular Generated Block)	
G40	07	Cutter Diameter Compensation (Cancel)	
G41		Cutter Diameter Compensation (Left)	
G42		Cutter Diameter Compensation (Right)	
G43	08	Tool Length Offset (Plus)	Modal
G43.1		Tool Length Offset Selection (Plus)	
G44		Tool Length Offset (Minus)	
G44.1		Tool Length Offset Selection (Minus)	
G45	23	Disable Spindle Synchronization	Modal
G46		Set Spindle Positional Synchronization	
G46.1		Set Active Spindle Speed Synchronization	
G47	24	Linear Acc/Dec in All Modes	Modal
G47.1		S-Curve Acc/Dec for Positioning and Exact Stop Mode	
G47.9		Infinite Acc/Dec (No Acc/Dec) (AMP-selectable only)	
G48	00	Reset Acc/Dec to Default AMPed Values	Non-Modal
G48.1		Acceleration Ramp for Linear Acc/Dec Mode	
G48.2		Deceleration Ramp for Linear Acc/Dec Mode	
G48.3		Acceleration Ramp for S-Curve Acc/Dec Mode	
G48.4		Deceleration Ramp for S-Curve Acc/Dec Mode	
G48.5		Programmable Jerk Value	
G49	08	Tool Length Offset Cancel)	Modal
G50.1	11	Programmable Mirror Image (Cancel)	Modal
G51.1		Programmable Mirror Image	
G52	00	Offsetting Coordinate Zero Point	Non-Modal
G53		Motion in Machine Coordinate System	
G54	12	Preset Work Coordinate System 1	Modal
G55		Preset Work Coordinate System 2	
G56		Preset Work Coordinate System 3	
G57		Preset Work Coordinate System 4	
G58		Preset Work Coordinate System 5	
G59		Preset Work Coordinate System 6	
G59.1		Preset Work Coordinate System 7	
G59.2		Preset Work Coordinate System 8	
G59.3		Preset Work Coordinate System 9	
G61	13	Exact Stop Mode	Modal
G62		Automatic Corner Override	
G63		Tapping Mode	
G64		Cutting Mode	
G65	00	Paramacro Call	Non-Modal
G66	14	Paramacro Modal Call	Modal

A	Modal Group	Function	Type
G66.1		Paramacro Modal Call	
G67		Paramacro Modal Call (Cancel)	
G68	16	Part Rotation	Modal
G69		Part Rotation (Cancel)	
G73	09	Deep Hole Peck Drilling Cycle (With dwell)	Modal
G74		Left-Hand Tapping Cycle	
G74.1		Left-Hand Solid Tapping Cycle	
G76		Boring Cycle (Spindle Shift)	
G80		Cancel or End Fixed Cycle	
G81	09	Drilling Cycle (No Dwell, Rapid Out)	Modal
G82		Drilling Cycle (Dwell, Rapid Out)	
G83		Deep Hole Peck Drilling Cycle	
G84		Right-Hand Tapping Cycle	
G84.1		Right-Hand Solid Tapping Cycle	
G85		Boring Cycle (No Dwell, Feed Out)	
G86		Boring Cycle (Spindle Stop, Rapid Out)	
G87		Back Boring Cycle	
G88		Boring Cycle (Spindle Stop, Manual Out)	
G88.1	00	Pocket Milling Roughing Cycle	Non-Modal
G88.2		Pocket Milling Finishing Cycle	
G88.3		Post Milling Roughing Cycle	
G88.4		Post Milling Finishing Cycle	
G88.5		Hemispherical Milling (Roughing Cycle)	
G88.6		Hemispherical Milling (Finishing Cycle)	
G89	09	Boring Cycle (With Dwell, Feed Out)	Modal
G89.1	00	Irregular Pocket Milling (Roughing Cycle)	Non-Modal
G89.2		Irregular Pocket Milling (Finishing Cycle)	
G90	03	Absolute Mode	Modal
G91		Incremental Mode	
G92	00	Coordinate System Offset (Using Tool Positions)	Non-Modal
G92.1		Coordinate System Offset (Cancel)	
G92.2		Selected Coordinate System Offsets (Cancel)	
G93	05	Inverse Time Feed Mode	Modal
G94		Feed-per-minute mode	
G95		Feed-per-revolution Mode	
G98	10	Initial Level Return in Milling Cycles	Modal
G99		R-Point Level Return in Milling Cycles	

END OF APPENDIX

Allen-Bradley 7300 Series CNC Tape Compatibility

Appendix Overview

The 7300 Series CNC tape compatibility feature has been developed for customers with an existing library of standard 7320 and 7360 CNC tapes. This feature allows those 7300 tapes to be read and executed by the control. If desired, these 7300 tapes can be copied into the control's memory to allow editing and execution, or they can be executed directly from tape, with the exception of 7300 pattern repeat subprograms.

Important: This feature is not intended to allow you to develop new 7300 part programs using the control. We strongly recommend that for new part programs, you learn to develop them using the control's programming method.

Important: S-Curve Acc/Dec mode is not available in 7300 mode.

To use the 7300 Series tape compatibility feature, the system installer must enable this feature in PAL. Refer to the system installer's documentation and your PAL reference manual for details on how the 7300 Series CNC tape format feature is activated.

G-code Compatibility Considerations

Table D.A lists all 7300 G-codes that the control can execute in 7300 mode. Refer to the System 7320 Programming Manual for details on these G-codes and their operation.

Table D.A
 G-code

G-code:	Function:
G00	Positioning mode
G01	Linear interpolation mode
G02	Circular/Helical ¹ motion CW
G03	Circular/Helical ¹ motion CCW
G04	Dwell
G17	XY plane selection
G18	ZX plane selection
G19	YZ plane Selection
G21	Linear interpolation with delay
G22/G23	Circular interpolation with delay
G40	Cutter diameter compensation off (optional)
G41	Cutter diameter compensation left
G42	Cutter diameter compensation right
G70	Inch mode
G71	Metric mode
G74	Full circle programming off
G75	Full circle programming, EIA standard
G76	Full circle programming, AB standard
G80	Cancel or end fixed cycles
G81	Drilling cycle (no dwell, rapid out)
G82	Drilling cycle (dwell, rapid out)
G83	Deep hole drilling cycle
G84	Tapping cycle
G85	Boring cycle (no dwell, feed out)
G86	Boring cycle (spindle stop, rapid out)
G87	Boring cycle (no dwell, spindle stop)
G88	Boring cycle (dwell, spindle stop)
G89	Boring cycle (dwell, feed out)
G90	Absolute programming mode
G91	Incremental programming mode
G93	V/D feedrate coding
G94	Feed per minute
G98	Insert absolute presets (non-modal)
G99	Cancel G98 (non-modal)

¹ Make sure your arc and linear move complete at the same time. In the event that you programmed your arc to finish first, the control is force to E-STOP.

M-code Compatibility Considerations

Table D.B lists all standard 7300 M-codes that the control can execute in 7300 mode.

Important: In order to provide the same functionality as the 7300 PAL, the system installer has to write a specific application in PAL when interfacing with the control.

Refer to the System 7320 Programming Manual for details on these M-codes and their operation.

Table D.B
 M-code

M-code:	Function:
M00	Program stop
M01	Optional program stop
M02	End of program
M03	Spindle on, CW
M04	Spindle on, CCW
M05	Spindle off
M06	Tool transfer
M07	Mist coolant on
M08	Flood coolant on
M09	Coolant off
M19	Spindle orient
M30	End of program, rewind
M48	Feedrate override, and spindle override, normal
M49	Feedrate override, and spindle override, bypass to 100%

M06 Tool Transfer

Depending upon your 7300 configuration, M06 can be executed in two ways:

- all tool change operations must be handled by the PAL program.

Note: This is the way M06 works on the 9/Series control.

or

- the active tool offset is cancelled and the axes move to a predefined tool change position.

In order for your control to do this, you have to reconfigure your AMP. We recommend that you do the following:

Set This AMP Parameter:	To:
Move Tool to 7300/M06 Position	True.
7300/M06 Home Parameter	home position for each axis.

Refer to your 9/Series CNC AMP Reference Manual for details on these parameters.

Important: When the control is in 7300 mode, tool offsets will always be in delayed/delayed mode, regardless of the AMP setting. When the system is not in 7300 mode, tool offsets will follow the AMP selections (i.e., immediate/immediate, immediate/delayed, or delayed/delayed).

Offset Compatibility Considerations

Tool Length Offset

When the control is in 7300 mode, tool length offsets are activated in the same manner as on the 7300. The control supports up to 4 digit T-words, and through AMP configuration, you have the flexibility of specifying how the control activates offsets.

Important: Since tool changers vary in style, size and function, the system installer is responsible for specific implementations through PAL. Refer to the PAL programming manual and the manual supplied by the system installer for more details.

We recommend that you use this set-up when running your control in 7300 mode:

Set This Tool Length Offset Parameter:	To:	Explanation:
Tool Geometry Mode (AMP [202]):	immediate shift/immediate move	once the offset is programmed, the geometry offset is activated immediately and the coordinate system gets shifted immediately.
Tool Wear Mode (AMP [203]):	immediate shift/immediate move	once the offset is programmed, the wear offset is activated immediately and the coordinate system gets shifted immediately.
Tool Offset Cancel (AMP [204]):	cancel geometry & wear	both the geometry and wear offset are cancelled when the active offset is cancelled.
Tool Number Integer Format (AMP [497]):	integer format	specify the integer format of a T-word that is used to program a tool number.

If the Tool Life Management option is present, sets these AMP parameters for 7300 compatibility:

Set This Tool Length Offset Parameter:	To:	Explanation:
Tool Number/Group Boundary (AMP [131]):	9799	specify the division between ordinary tool numbers and tool life groups
T-word Programming Method (AMP [132]):	next Tool in T-word	for this method, the T-word to be activated is programmed in a block that does not contain an M06

Important: For systems with the Tool Life Management option, the T-word is limited to 4 digits in the range of 1 to 9799.

Refer to your AMP reference manual for details on these parameters.

Important: When 7300 mode is active on the control, automatic tool changing requires the proper PAL interface and may require changing the AMP parameters described above. The Random Tool feature of the control is also available to interface with special tool changers.

The 9/Series control has two offsets tables: geometry and wear table. The sum from these two tables is used to generate tool length data when the tool offset number is programmed. When in 7300 mode, the active offset is also computed as the sum of the geometry and wear offsets. Refer to Chapter 3 for details.

Important: When changing from inch to metric (or vice versa) in 7300 mode, the 9/Series control does automatic conversion on tool offset values. The 7300 did not provide this automatic conversion, and instead, required the operator to re-enter tool offsets.

Important: The 9/Series control allows the Power-Turn-On (PTO) mode of the control to be specified in AMP with respect to inch/metric (G70/G71) and absolute/incremental (G90/G91) etc. For 7300 tape compatibility, these parameters may need to be modified if a certain PTO mode is expected. Refer to your AMP reference manual for details on these parameters.

Fixture Offset

In place of fixture offsets, the control has nine preset work coordinate systems selectable using G54-G59.3. The required work coordinate system can be selected by specifying any of these G-codes in the part program. The default or power-up work coordinate is specified in AMP. For 7300 compatibility, the work coordinates G54-G57 have been assigned to correspond with Pallet A and the work coordinates G58-G59.2 have been assigned to correspond with Pallet B.

Important: The system installer has to write a specific application in PAL to access all 8 preset work coordinates in 7300 mode. H1 to H4 selects G54 to G57 or G58 to G59.2 depending upon how the system installer has written the PAL program. Refer to your PAL reference manual for details.

Important: An activated fixture offset is applied in the next programmed move.

Additional Feature Compatibility Considerations

In this section, we discussed how to copy a 7300 tape to the control. This process involves the following steps:

- store a 7300 part program into the control
- execute the 7300 part program

Storing 7300 Part Programs

A 7300 tape can be stored into control memory using the COPY PROGRAM softkey function. While reading the tape to the control, any TN code, which defined the program name for 7300's, is interpreted as a comment block; this means a new program name must be entered. The control prompts the user to enter a program name for each part program or pattern repeat stored. The name given to the program must adhere to the format of 9/Series program names.

Important: When storing a pattern, the pattern name entered here must be a 1 to 5 digit (maximum) numeric prefixed with the letter O. For example, O10005. Refer to Section 10.5.9 for information on valid subprogram names.

At this time, the control creates an internal cross-reference table for all pattern repeat names. The cross-reference table is generated so that any blocks that call pattern repeat do not need to be rewritten using the new program name. Refer to the section on Pattern Repeat for details.

You can not copy pattern repeat programs from memory to memory. Doing so does not create the necessary cross-reference table.

Pattern Repeat

(CP, name, r)#

where:

Name:	Indicates:
CP	a pattern repeat is called
name	the pattern blocks being called (part program name)
r	the number of times the pattern blocks get executed
#	end of block

Important: The patterns called must be saved as independent part programs and copied into control memory from a peripheral device. You can not create new pattern repeat programs online. A maximum of 20 patterns can be copied into the controls memory.

You can store more than one pattern in the control by using the COPY PRGRAM softkey function. The control requires that a new program name be entered for each pattern. The control automatically creates an internal cross-reference table of 7300 pattern names. This cross-reference table consists of the subprogram file name called by the main program, and the new file name entered for the pattern repeat. The control takes the first program start code from the tape and corresponds it with the file name you have just entered for the pattern repeat; this way, you do not need to modify you main program so that it still calls the correct pattern. This cross-reference table is maintained by the control and it is transparent to you. For example:

You have a 7300 part program that has a pattern named BL-M008-001. When you stored the pattern to the control, the control prompted you for a name, you entered O00001. The part program and the pattern are now stored in the memory. At this point, the control searches for the program start code on the tape and takes the code and matches it with O00001. When this pattern name is called, the control automatically searches the cross-reference table for BL-M008-001 and executes O00001.

Important: The (DP) block is saved in memory as part of the program, and it is treated as a comment block during the execution of the part program.

Executing 7300 Part Programs

The system installer has to write PAL program for control to execute in 7300 tape compatibility mode. Refer to your PAL reference manual for details.

The control allows the Power-Turn-On mode (PTO) of the control to be specified in AMP with respect to inch/metric (G70/G71) mode and absolute/incremental (G90/G91) programming mode. For 7300 tape compatibility, we recommend that you select G70 and G90 for PTO mode. When 7300 tape compatibility mode is enabled, these modal conditions are automatically established and cannot be selected in AMP:

- G36.1 Short block Acc/Dec enabled
- G39.1 Cutter Diameter Compensation circular transition

Important: When programming a G00 in 7300 mode, the 9/Series control uses the <RAPID OVERRIDE> switch to override the rapid feedrate. This applies to any programmed rapid move including moves generated by the fixed cycles or pattern repeats. The operation of the <RAPID OVERRIDE> switch is very PAL dependent, and can be alter by your system installer's PAL logic. Refer to your system installer's documentation for details.

The part program can be executed either from tape or control memory. If a 7300 program makes pattern calls, the patterns must reside in control memory.

The control attains compatible pattern repeat functionality by using an M98 subprogram call. When the control executes the 7300 pattern repeat call, it makes these substitutions:

Control Uses:	To substitute 7300s:	To Indicate:
M98	CP	a pattern repeat is called
Pnnnnn	name	the pattern blocks being called (part program name)
Lmmm	r	the number of times the pattern blocks get executed
#	#	end of block

Important: In order for the control to properly execute a M98 subprogram call, the blocks called by the pattern repeat must be stored under a program name starting with the letter O, followed by up to 5 numeric characters. Refer to Section 10.5.9 for information on valid file names.

The main program, which has the pattern repeat call block “(CP, name, r)”, can be executed from tape or from the control’s memory. However, if you want to make minor editing to your main program, you must copy the program into the control’s memory. Refer to Section 9.2, “Inputting Part Programs,” for details on how to copy a program from tape.

Important: To execute a program from tape, the tape must be positioned at the start of the main program.

For more information on how subprograms are executed in the control, refer to chapter 10, “Subprogram Call (M98).”

9/Series G-codes Applicable
 to the 7300 Series CNC

You may enhance your 7300 part program flexibility with selected 9/Series features. Table D.C lists the control’s G-codes that can be executed in 7300 mode. However, we recommend that you use the 9/Series programming method for optimal performance. Refer to earlier sections of this manual for details on these G-codes.

Table D.C
 Mill G-codes Available in 7300 Mode

G-code:	Description:
G09	Exact stop
G10L2	Set up work coordinate offset table
G10L3	Set up tool management table
G10L10	Set up tool offset geometry table
G10L11	Set up tool offset wear table
G10L12	Set up tool diameter wear table
G10L13	Set up tool diameter geometry table
G11	Set up tool management table cancel
G14	Disable scaling
G27	Machine home return check
G28	Automatic return to home
G29	Automatic return from home
G30	Return to secondary home
G31	External skip function 1
G36	Short block Acc/Dec disabled
G37	Tool gauging skip function 1
G38	Hole probing
G39	Cutter compensation (linear transition)
G43	Tool length plus
G44	Tool length minus

Table D.C (continued)
 Mill G-codes Available in 7300 Mode

G-code:	Description:
G49	Tool length cancel
G52	Offset coordinate zero point
G53	Motion in machine coordinate system
G54	Preset work coordinate system 1
G55	Preset work coordinate system 2
G56	Preset work coordinate system 3
G57	Preset work coordinate system 4
G58	Preset work coordinate system 5
G59	Preset work coordinate system 6
G61	Exact stop mode
G62	Automatic corner override
G63	Tapping mode
G64	Cutting mode
G65	Paramacro call
G66	Paramacro modal call
G67	Paramacro modal cancel
G68	Part rotation
G69	Part rotation cancel
G92	Coordinate offset using tool positions
G95	Feed per revolution mode

7300 Series Features Not Supported

The control does not support these 7300 Series CNC features:

- 7300 Scaling (using “SCF” with G-codes)
- 7300 Part Rotation
- Additional Plane Selection (beyond G17, G18 and G19)
- 7300 Tool Axis Switching
- 7300 Cutter Compensation Entry and Exit Motion

All cutter compensation moves, except for the entry and exit moves, use the same paths for 7300 and control. However, the entry and exit moves of cutter compensation uses the control’s tool paths. Refer to chapter 21 for details.

END OF APPENDIX

Symbols

- ; As End of Block, 10-11
- / Block Delete, 10-10
- / Block Delete Character, 7-1

Numbers

- 7300 Series CNC Tape
 - Compatibility
 - 9/240 G Codes Applicable, D-9
 - Features Not Supported on 9/240, D-10
 - G Code Considerations, D-1
 - M Code Considerations, D-3
 - M06 Tool Transfer, D-4
 - Overview, D-1
 - Tool Length Offset, D-4
 - Tool Life Management, D-5
 - Part Programs
 - Copying into 9/240 Memory, D-6
 - Executing from 9/240, D-8
 - Pattern Repeat, D-7
 - Storing in 9/240 Memory, D-6

A

- Abort Execution at Block N, 7-2
- Absolute Coordinate System, 11-1
- Absolute Home, Manual, 4-8
- Absolute Mode (G90 & G91), 13-12
- Absolute Position Display, 8-6
- Absolute, Motion in, 11-2
- Acceleration/Deceleration, 18-14
 - For short blocks, 18-22
- Access Control, 2-25
 - Assigning Access Levels and Passwords, 2-26
 - Passwords, Entering, 2-31
 - Protection of Passwords, 2-29
- Access Levels, 2-26
- Active Offset, changing, 3-12
- Active Process, Selecting for a Dual-Process, 30-2
- Active Program, de-selecting, 7-9
- Active Program Search, 7-10
- Active Program, Selecting, 7-6
- Adaptive Depth Probe, 27-18
- Adaptive Feed (G25), 18-9
- Address, Definition, 10-6

- All Position Display, 8-19
- Alternate, home return, 14-34
- AMP, Reference Manual, 1-5
- AMP Feedrate, 18-12
- Angles, for polar programming, 14-21
- Angular Jogging, 4-5
- Arcs, circular mode, 14-5
- Area, Protected Zones, 12-5
- ASCII (RS-358-B), 9-7
- Auto Erase, 8-32
- Auto File Name, 9-7
- Auto Size, 8-30
- AUTO, Selecting, 2-36
- Automatic Corner Override (G62), 18-21
- Automatic Homing, 14-29
- Automatic Mode, 7-23
- Automatic Mode, Selecting, 2-36
- Auxiliary Home Return, 14-34
- Axis, setup, see AMP manual, 1-5
- Axis Clamp, 14-39
- Axis Detach, 2-43
- Axis Direction, 2-14
- Axis Inhibit, 7-17, 7-20
- Axis Motion (Min/Max), 10-21
- Axis Mover, PAL, 14-20
- Axis Names, 10-22
- Axis Position, Display Screen, 13-17
- Axis Position Data Display, 8-1
- Axis Select (Large Display Screens Only), 8-15
- Axis Select Switch, 4-3
- Axis Speed, feedrates, 18-1
- Axis, Detach, 4-8
- Axis, Shared, 30-26

B

- B-word, 10-37
- Backing Up Interference Tables, 30-23
- Backing up offset tables, 3-17
- Backing Up Parameter Values, 28-43
- Backing up Part Programs, to ODS, 6-12
- Backup Memory

- Definition, 1-4
 - Password Protection, 2-30
 - Setting
 - Power on Time/After Reset, 2-47
 - Power on Time/Overall, 2-46
 - Base Coordinate System, 11-1
 - Basic Control Operation, 2-1
 - Basic Program Execution, 7-17
 - Baud Rate, Selecting MAX, 9-5
 - Baud Rate, Setup, 9-5
 - Block by Block, 2-14
 - Block by Block Execution, 7-4
 - Block Delete, 7-1, 10-10
 - Block End, 10-11
 - Block Number, 5-13
 - Block Numbers, 10-9, 10-37
 - Block Reset, 2-3, 2-4
 - during MDI, 4-14
 - Block Retrace, 2-14, 7-31
 - Block, Program, 10-6
 - Blocks, adding comments, 10-10
 - Boring Cycles
 - Back Boring Cycle (G87), 26-2, 26-32
 - Boring Cycle, Spindle Shift (G76), 26-2
 - Cancel or End Fixed Cycle (G80), 26-2
 - No Dwell/Feed Out (G85), 26-2, 26-28
 - Spindle Shift (G76), 26-15
 - Spindle Stop, Rapid Out (G86), 26-2, 26-30
 - With Dwell, Feedout (G89), 26-2, 26-36
 - With Dwell, Spindle Stop, Man Out (G88), 26-2, 26-34
- ## C
- C axis, for cylindrical, 14-14
 - C-axis, rotary, 14-12
 - CALC key, 2-4
 - Calculator
 - Mathematical Functions, 2-6
 - Paramacro Variables in, 2-7
 - Valid Operators, 2-5
 - Calculator Functions, 2-4
 - Call for subprogram, 10-34
 - Cancel Fixed Cycles (G80), 26-30
 - Cancel or End Fixed Cycle (G80), 26-18
 - Canceling Coordinate System Offset (G92.1), 11-20
 - Canceling Selected Coord. Sys. Offset (G92.2), 11-21
 - Change Directory, 2-31
 - Changing Languages, 8-23
 - Changing Parameters
 - Auto Erase, 8-32
 - Auto Size, 8-30
 - Grid Lines, 8-30
 - Overtravel Zone Lines, 8-30
 - Process Speed, 8-32
 - Rapid Traverse, 8-29
 - Select Graph, 8-29
 - Sequence Starting #:, 8-31
 - Sequence Stopping #:, 8-31
 - Changing parameters, {GRAPH SETUP}, 8-28
 - Character Encryption, for part programs, 5-46
 - Characters, Definition, 10-6
 - Check, home position, 14-33
 - Checking Part Programs
 - Axis Inhibit, 7-20
 - QuickCheck, 7-19
 - Chinese, Language Display, 8-23
 - Circular Interpolation Mode (G02, G03), 14-5
 - Clamp Axis, 14-39
 - Clamp, axis, 14-39
 - Clearing Error Messages, 2-40
 - Clearing Screen, 8-33
 - Clock, System, 2-43
 - Clockwise, circular interpolation, 14-5
 - Close Editor, 5-16
 - Codes, Definition, 10-6
 - Color, Changing tool graphics, 20-29
 - Color CRT, 2-10
 - Combining Part Programs, 5-15
 - Comm Port, executing from, 7-5
 - Comment Block, 10-10
 - Comment Display {PRGRAM COMENT}, 5-40
 - Comments on Tapes, 10-5
 - Communication Between Processes, 30-6
 - Communications, 9-1
 - Communications Port Configuration, 9-1
 - Communications Port Parameters, 9-3
 - Conditional Operators, 28-7
 - Continue softkey, 2-8
 - Continuous Jog, 4-3
 - Control Panel, 2-12

Control Reset, 2-3, 2-4
Coordinate Offset, on shared axis, 30-29
Coordinate System
 inch/metric, 13-13
 Offset Tables, 3-14
 Offsetting Work Systems, 11-13
 Rotating (G68, G69), 13-2
 Rotating External, 13-6
Coordinate System Offset (G92), 11-13
Coordinate System, Absolute, 13-12
Coordinate System, Machine, 11-1
Coordinate Systems, rotating, 13-1
Coordinate Systems, Work, 11-4
Copying Program {COPY PRGRAM}, 5-41
Copying programs, 5-16
Counter Clockwise, circular interpolation, 14-5
Coupling dual axes, on dual process controls, 30-32
Creating a Part Program, 31-9
CRT, 2-1, 2-10
CRT Displays, 8-1
CRT screen saver, 8-39
Cursor, Moving, 5-5
Cursor Keys, 2-3
Custom Screens, through PAL, 8-22
Cutter Compensation (G40 G42), 21-1
 Block Look Ahead, 21-50
 Changing Cutter Radius During Compensation, 21-43
 Changing Direction, 21-35
 Changing Work Coordinate System, 21-49
 Corner Movement After Generated Blocks, 21-41
 Error Detection, 21-51
 Generated Blocks, 21-7
 MDI During Cutter Compensation, 21-46
 Moving To/From Machine Home, 21-48
 Non Motion Blocks, 21-39
 Type A, 21-10
 Entry Moves, 21-10
 Exit Moves, 21-14
 Type B, 21-20
 Entry Moves, 21-20
 Exit Moves, 21-24
Cutter Compensation (G41, G42), Error Detection
 Circular Departure Too Small, 21-52
 Disabling, 21-53
 Interference, 21-53
Cutter Compensation (G410 G42), Tool Paths During, 21-30
Cutting Feedrate, Maximum, 7-21
Cutting Mode (G64), 18-21

Cutting Speed, 10-39
Cutting Speeds, 10-23
Cutting Torque, G25, 18-9
Cycle Start, 2-14
Cycle Start for Dual-Processing, 30-2
Cycle start using softkeys, 2-19
Cycle Stop, 2-14
Cycle Stop for Dual-Processing, 30-11
Cycle Time (per program), 2-48
Cylindrical Interpolation, 14-14
Cylindrical Interpolation, restrictions, 14-20

D

D-word, 10-22
Data Length, for communications, 9-6
Date, Setting, 2-43
De-coupling duals
 drilling axis, 30-34
 on dual processing, 30-32
 programming de-coupled axes, 30-33
Deep Hole Drill Cycle (G83), 26-2, 26-21
Deep Hole Peck Drilling Cycle with Dwell (G73), 26-9
Default, Resetting to, 2-4
Delete Block Character, 7-1
Delete Key, 2-3
Deleted Blocks, 10-10
Deleting a Program {DELETE PRGRAM}, 5-37
Depth Probe, 27-18
Detached Axis, 2-43
Detaching a Servo, 4-8
Device, for program execution, 7-5
Device Setup, 9-2
Diameter Offsets, entering values, 3-4
Diameter Offsets, entering wear values, 3-5
Digitizing a Program, Teach, 5-28
 Arc (3 Points), 5-32
 ARC Tangent at End Points, 5-34
 Linear, 5-30
Directory, protected part programs, 5-43
Disable Servos, 4-8
Disable/Enable Overrides, 10-34
Display Pages, PAL, 8-22
Display Select, 8-1

- Displaying a Program {DISPLAY PRGRAM}, 5-39
 - Displaying Machine Information, 8-33
 - Displaying Position
 - ABS, 8-6
 - ABS (Large Display), 8-7
 - absolute (Small Display), 8-8
 - ALL, 8-19
 - distance to go (Small Display), 8-14
 - DTG, 8-12
 - DTG (Large Display), 8-13
 - G Code Status, 8-20
 - M Code Status, 8-16
 - PRGRAM, 8-3
 - PRGRAM (Large Display), 8-4
 - PRGRAM (Small Display), 8-5
 - PRGRAM DTG, 8-17
 - program/DTG (Small Display), 8-18
 - Target, 8-9
 - Target (Large Display), 8-10
 - target (Small Display), 8-11
 - Distance to Go Position Display, 8-12
 - Downloading Part Programs from ODS, 6-5
 - Drawings, creating programs from, 5-20
 - Drilling Axis, on dual axis, 30-34
 - Dry Run, 7-17, 7-21
 - Dual Axis, 19-1
 - Configuration, 19-1
 - Homing, 19-4
 - Invalid Operations, 19-6
 - Offsets for, 19-7
 - on Dual Processing, 30-31
 - Parking, 19-3
 - Programming, 19-5
 - programming de-coupled axis on dual processing, 30-33
 - shared drilling axis, 30-34
 - Terms, 19-2
 - Dual-Processing, 30-1
 - Active Process, 30-2
 - Cycle Start, 30-2
 - Cycle Stop, 30-11
 - De-coupling duals, 30-32
 - Definition, 30-1
 - dual axes on, 30-31
 - Dual-horizontal mill, 30-1
 - Editing a Part Program, 30-4
 - Error Messages, 30-5
 - Graphics, 30-4
 - Interference Boundaries, 30-16
 - Interference Checking, 30-12
 - Interference Values, 30-19, 30-21
 - MDI Mode, 30-11
 - Operating a Shared Axis, 30-26
 - Operation, 30-2
 - Part Program Directory, 30-3
 - Preventing Collisions, 30-15
 - QuickView, 30-4
 - Reset Operations, 30-6
 - Shared Axes, 30-26
 - Shared Parameters, 30-6
 - Single-Digit Feedrates, 30-5
 - Spindle Control, 30-12
 - Split Screen Display, 30-3
 - Switching a Shared Axis, 30-28
 - Synchronizing Part Programs, 30-7
 - Dual-Horizontal Mill, 30-1
 - Duplicating Programs, 5-41
 - Dwell (G04), 14-35
 - Seconds, 14-36
 - Spindle Revolutions, 14-36
- ## E
- E-STOP, 2-24
 - Editing a Part Program, 31-12
 - Editing a Part Program for Dual-Processing, 30-4
 - Editing a Program, protectable program directory, 5-43
 - Editing a program, Selecting, 5-2
 - Editing Part Programs
 - with Quickview, 31-13
 - with the part program editor, 31-13
 - Editing Part Programs Offline, 6-2
 - Editing Programs, 5-3
 - Changing and Inserting, 5-7
 - Entering Characters and Blocks, 5-7
 - Erasing Characters and Blocks, 5-11
 - Exiting Edit Mode, 5-16
 - String and Character Search, 5-5
 - Editing Programs Offline, 6-1
 - Editing Programs Online, 5-1
 - EIA (RS-244), 9-7
 - Emergency Stop Operations, 2-14, 2-24
 - Emergency Stop Reset, 2-14, 2-24
 - Enable/Disable Overrides, 10-34
 - Encoder, as a depth probe, 27-18
 - Encryption, part programs, 5-46
 - End of Block key, 2-3
 - End of Block Statement, 10-11
 - End of Program M02, 10-33
 - End of Program M99, 10-14

End of Program Rewind M30, 10-34
End Program on Tape, 10-5
Energizing the Control, 2-21
English, Language Display, 8-23
English/Metric, 13-13
Enlarging, scaling, 13-14
Entering Characters and Blocks, 5-7
Entering Interference Values Manually, 30-19
Entering Interference Values Through Programming, 30-21
Entering Part Programs Offline, 6-1
EOB end of block, 10-11
EOB Key, 2-3
EOB when editing programs offline, 6-4
Erasing a Program, 5-37
Error Detection, 21-51
Error Display, CRT, 2-10
Error Messages, 2-37
 Clearing, 2-40
 Currently Active, 2-38
 Error Log, 2-38
 System, B-1
Error Messages for Dual-Processing, 30-5
Evaluate, math expressions, 2-5
Exact Stop Mode (G61), 18-20
Execute Program from Comm Port, 7-5
Execute program using softkeys, 2-19
Executing a Program, 7-1
Executing Programs, selecting active, 7-6
Execution, Selecting AUTO mode, 2-36
Exit Editor, 5-16
Exit softkey, 2-8
Exponential Acc/Dec, 18-15
External Offset, 11-9
 table for defining, 3-14
External Part Rotation, 13-6

F

F word, single digit feedrate tables, 3-23
F-word, 10-23
F1 F4, 2-14
Feed Per Minute Mode (G94), 18-5
Feed Per Revolution (G95), 18-5

Feed to Hard Stop (G24), 14-40
Feedback, as an adaptive depth probe, 27-18
Feedhold Status, 7-18
Feedrate
 as torque control (G25), 18-9
 dry run, 7-21
Feedrate Limits, 18-8
Feedrate Override, 2-14, 18-7
Feedrate Switch, External, 18-13
Feedrates, 10-23, 18-1
 Applied During Cutter Comp, 18-2
 linear mode, 14-3
 rapid, 14-1
 single digit, 3-23
Feedrates, circular mode, 14-5
File Name to/from Tapes, 9-7
Find, active program search, 7-10
Flip, mirror, 14-36
Force on cutting tool, G25, 18-9
Format, RAM Disk, 2-41
Freeze, clamp axis, 14-39
French, Language Display, 8-23
Function Keys (Softkeys), 2-8
Functions, in calc operation, 2-6

G

G Code Format Prompting, 5-24
G Code Status, 8-20
G Code, Using LZS and TZS, 10-18
G Codes
 G00, 14-1
 G01, 14-3
 G02, 14-5, 14-10
 G03, 14-5, 14-10
 G04, 14-35
 G09, 18-20
 G10, 20-18, 20-29
 G10L2, 11-7
 G11, 20-29
 G15, 14-14, 14-21
 G16, 14-21
 G16.1, 14-14
 G17, 13-11
 G18, 13-11
 G19, 13-11
 G20, 13-13

- G21, 13-13
- G22, 12-5
- G22.1, 12-7
- G23, 12-5
- G23.1, 12-7
- G24, 14-40
- G25, 18-9
- G26, 27-18
- G27, 14-33
- G28, 14-29, 14-30
- G29, 14-32
- G30, 14-34
- G31, 27-2
- G31.1, 27-2
- G31.2, 27-2
- G31.3, 27-2
- G31.4, 27-2
- G37, 27-4
- G37.1, 27-4
- G37.2, 27-4
- G37.3, 27-4
- G37.4, 27-4
- G38, 27-8
- G38.1, 27-12
- G39, 21-7
- G39.1, 21-7
- G40, 21-3
- G41, 21-3
- G42, 21-3
- G43, 20-3
- G44, 20-3
- G47, 18-18
- G48, 18-19
- G49, 20-3
- G50.1, 14-36
- G51.1, 14-36
- G52, 11-16
- G53, 11-2
- G54, 11-4
- G55, 11-4
- G56, 11-4
- G57, 11-4
- G58, 11-4
- G59, 11-4
- G59.1, 11-4
- G59.2, 11-4
- G59.3, 11-4
- G61, 18-20
- G62, 18-21
- G63, 18-21
- G64, 18-21
- G65, 28-45
- G66, 28-46
- G66.1, 28-48
- G67, 28-46
- G68, 13-2
- G69, 13-2
- G73, 26-9
- G74, 26-10
- G74.1, 26-12
- G76, 26-2, 26-15
- G80, 26-2, 26-18
- G81, 26-2, 26-18
- G82, 26-2, 26-20
- G83, 26-2, 26-21
- G84, 26-2, 26-23
- G84.1, 26-25
- G85, 26-2, 26-28
- G86, 26-2, 26-30
- G87, 26-2, 26-32
- G88, 26-2, 26-34
- G88.1, 22-1
- G88.2, 22-15
- G88.3, 23-1
- G88.4, 23-7
- G88.5, 24-1
- G88.6, 24-7
- G89, 26-2, 26-36
- G89.1, 25-2
- G89.2, 25-10
- G90, 13-12
- G91, 13-12
- G92, 11-13
- G92.1, 11-20
- G92.2, 11-21
- G93, 18-4
- G94, 18-5
- G95, 18-5
- G98, 26-6
- G99, 26-6
- G codes, G14.1, 13-14
- G-code Overview, 10-24
- G-code table, 10-26
- G-codes
 - definition of, 10-24
 - table of, 10-26
- G90, with Transfer Line Cycles, 31-25
- G91, with Transfer Line Cycles, 31-25
- Gauge Line, 3-3
- Geometry Offset Table, 3-2, 3-6
- German, Language Display, 8-23
- Graphics
 - Active Graphics, 8-24
 - Auto Erase, 8-32
 - Auto Size, 8-30
 - Changing Parameters, 8-27
 - Clearing Screen, 8-33

Disabling, 8-27
 Grid Lines, 8-30
 Machine Information, 8-33
 Overtravel Zone Lines, 8-30
 Process Speed, 8-32
 Rapid Traverse, 8-29
 Running Graphics, 8-25
 Scale, 8-26
 Select Graph, 8-29
 Selecting a Program, 8-24
 Sequence Starting #:, 8-31
 Sequence Stopping #:, 8-31
 Single-Block, 8-33
 Tool Paths, 8-24
 Zooming, 8-33
 Graphics for Dual-Processing, 30-4
 Graphing Tool Path, 8-24
 Grid Lines, 8-30
 Group Number for M-codes, 10-32
 Groups, for dual axes, 30-32
 Groups, Modal, 10-25

H

H Word, 20-7
 H-word, 10-22
 Hand wheel, Jogging with, 4-4
 Hard Stop, Feed To (G24), 14-40
 Hardware Overtravels, 12-2
 Helical Interpolation Mode (G02, G03), 14-10
 Hemisphere Milling
 Finishing Cycle G88.6, 24-7
 Concave Hemisphere, 24-8
 Convex Hemisphere, 24-10
 Roughing Cycle G88.5, 24-1
 Concave Hemisphere, 24-2
 Convex Hemisphere, 24-5
 Hold, clamp position, 14-39
 Hold, dwell, 14-35
 Hole Probing (G38), 27-8
 Hole Probing Cycle (G38), 27-9
 Home, secondary return, 14-34
 Homing a Dual Axis, 19-4
 Homing, Manual Machine, 4-8
 Homing, the Axis
 Automatic Homing, 14-29
 Automatic Return from Machine Home (G29), 14-32
 Automatic Return to Machine Home, 14-30

Machine Home Check (G27), 14-33
 HPG jogging, 4-4

I

I-word, 10-30
 Inch/Metric Modes (G20, G21), 13-13
 Incremental Jog, 4-3
 Incremental Mode, 13-12
 Incremental/Absolute and G10, 11-8
 Inhibit, Axis, 7-17
 Inhibit, Axis Mode, 7-20
 Input Cursor, 2-41
 Input Device, for part programs, 7-5
 Integrand Words, 10-30
 Integrands, for circular interpolation, 14-8
 Integration Manual, 1-5
 Interfacing the Workstation with the Control, 6-5
 Interference Boundaries, 30-16
 Interference Checking, 30-12
 Activating, 30-14
 Backing up Interference Tables, 30-23
 Entering Interference Values, 30-21
 Interference Values, 30-19
 Measuring Interference Boundaries, 30-16
 on shared axis, 30-13
 Preventing Collisions, 30-15
 Interference Tables, 30-23
 Interference Values, 30-19, 30-21
 Interpolation, linear, 14-3
 Interrupted Program Recover, {MID ST PROGRAM}, 7-25
 Interrupts
 Disabling, 29-1
 Enabling, 29-1
 Program, 29-8
 Request, 29-4
 Types, 29-5
 Inverse Time Feed Mode (G93), 18-4
 Irregular Pocket Milling, 25-1
 Finishing Cycle G89.2, 25-10
 Roughing Cycle G89.1, 25-2
 Italian, Language Display, 8-23

J

J-word, 10-30

Japanese, Language Display, 8-23
Jog Offset Function, 4-6
Jog on the Fly, Offsets, 11-19
Jog Retract, 2-14, 7-28
Jog Select, 2-13
Jog Select Switch, 4-3
Jogging
 Arbitrary Angle Jog, 4-5
 Continuous Jog, 4-3
 HPG Jog, 4-4
 In Auto mode, 4-6
 Incremental Jog, 4-3
 Jogging an Axis, 4-2
 Jogging an Offset, 11-19
 jogging an offset, 4-6
 Jogging at Rapid (TRVRS), 2-14
 on the fly, 4-6
Jogging using softkeys, 2-18

K

K-word, 10-30
Keyboard, 2-1

L

L-word, 10-37
Label, block number, 5-13
Label, stop at block number, 7-2
Labels, for Blocks, 10-9
Languages, Changing, 8-23
Leader Section for Tapes, 10-3
Leading Zero Suppression (LZS), 10-18
Left Hand Solid Tapping Cycle (G74.1), 26-12
Left Hand Tapping Cycle (G74), 26-10
Length Offset
 Activating, 20-8
 Axis Selection, 20-9
Length Offset, Tool, 20-3
Length Offset, Tool data, 3-2
Length Offset, Wear Data, 3-4
Length, Offset Table, 3-1
Limit, G24 hard stop, 14-40
Limits, programmable zone tables, 3-21
Limits, Hardware Overtravels, 12-2

Limits, Overtravels and Zones, 12-1
Limits, Programmable Zones, 12-5
Limits, Resetting, 12-13
Limits, Software Overtravels, 12-3
Line Numbers, 10-9
Linear Acc/Dec, 18-16
Linear Interpolation Mode (G01), 14-3
Local Parameters, 28-12
Lock out M code Functions, 7-2
Log, of errors, 2-38
Lot completed, 2-49
Lot Remaining, 2-49
Lot Size, 2-48
LZS, Using, 10-18

M

M code Ignore, 7-2
M code ignore, 10-41
M Code Status Display, 8-16
M Codes
 M00, 7-18
 M00 program stop, 10-33, 28-44
 M01, 7-18
 M01 optional stop, 10-33
 M02 end of program, 10-33
 M06 Tool Change, 10-33
 M100-M149 simple synchronization, 10-35
 M150-M199 synchronization with Setup, 10-36
 M30 end of program rewind, 10-34
 M48 overrides enabled, 10-34
 M49 overrides disable, 10-34
 M98 subprogram call, 10-13, 10-34
 M99, 10-14
 M99 end of subprogram/program, 10-35
M Codes Status, M96/M97, 29-1
M-code Table, 10-32
M-codes, 10-30
 group number, 10-32
 table of, 10-32
 Transfer Line Values, 31-16
Machine Coordinate System, 11-1
 Motion in the Machine Coordinate, (G53), 11-2
Machine Home Moving to and From, 21-48
Machine Home Return (G27), 14-33
Machine Home, automatic, 14-29

Machine Home, Establishing, 11-2
Machine Home, Manual, 4-8
Machine Information, 8-33
Machine Messages, 2-37
 Clearing Active Messages, 2-40
Macro
 Call Commands, 28-44
 Nesting, 28-52
 Output Commands, 28-54
Magazine, Tool Selection, 10-40
Magnification, scaling, 13-14
Main Program End M99, 10-35
Main Program Sequence Starting #:, 8-31
Main Program Sequence Stopping #:, 8-31
Maintenance Manual, 1-5
Manual (Operator's), Design, 1-1
Manual Homing, 4-8
Manual Mode, Selecting, 2-34
Manual Operating Mode, 4-1
Manual Positioning, jogging, 4-2
Math Key, 2-4
Mathematical
 Function Commands, 28-4
 Operators, 28-2
Mathematical Functions, 2-6
Mathematical Operators, Valid, 2-5
Max Axis Speed, 14-1
Maximum Baud Rate, Setup, 9-5
Maximum Cutting Feedrate, 7-21
MDI, block reset, 4-14
MDI Basic Operation, 4-12
MDI Mode, 4-11
 Synchronization, 30-11
MDI mode, Selecting, 2-35
Measure, Offset Data, 3-9
Measuring Interference Boundaries, 30-16
Mechanical Handle Feed, 4-8
Mechanical Stop, Hard (G24), 14-40
Memory, part program storage capacity, 6-4
Memory Loss, conditions of, 31-17
Memory, search with recall, 7-13
Merging Part Programs, {MERGE PROGRAM}, 5-15
Message Area, 2-10
Message Pending Error, 8-39

Message, at PTO, 8-37
Metric Mode, 13-13
Mill Cycle Format Prompting, 5-26
Millimeters/Inches, 13-13
Milling Fixed Cycles, 26-2
 Milling Cycle Format Prompting, 5-26
 Milling Cycle Operations, 26-8
 Milling Cycle Parameter, Altering, 26-38
 Milling Fixed Cycle Operating Parameter, 26-38
Mirror, 14-36
Mirror Image, manual, 14-38
Miscellaneous Function Lock, 7-2, 10-41
Miscellaneous Function, (B-words), 10-37
Miscellaneous Function, (M-words), 10-30
Modal Paramacro Call
 G66, 28-46
 G66.1, 28-48
Modal, G-codes, 10-25
Mode, Manual Operation, 4-1
Mode Select, 2-13
 Manual, 4-1
Mode, G code Display, 8-20
Monitor, CRT, 2-10
Monochrome CRT, 2-10
Monochrome Operator Panel, 2-2
More softkey, 2-8
MTB Panel
 function of buttons or switches, 2-13
 push-button, 2-12
Multi-Axis Jogging, 4-5
Multi-Level Block Delete, 10-10
Multi-Processing, 30-1

N

N Word, stop at block, 7-2
N word, block number, 5-13
N word, Sequence Numbers, 10-9
N-word, 10-37
Name Program, renaming, 5-38
Names, for Program, 10-8
Nesting Subprograms, 10-16
Number of Workpieces, 2-48
Numbering of Blocks, 10-9

O

- O word, as program name, 10-8
- O-words, 10-37
- ODS
 - Downloading Part Programs, 6-5
 - Uploading Part Programs, 6-12
- ODS, Using to Edit Part Programs, 6-1
- Offset, Length Offset, 20-3
- Offset Data, Measure Feature, 3-9
- Offset Management for Dual Axis, 19-7
- Offset Table, 3-1
- Offset Tables
 - Backing up the Tool Offset Tables, 3-17
 - clearing all, 3-18
- Offset Tables for Dual-Processing, Backing up Tables, 30-23
- Offset, Activation, 10-22
- Offset, Jogging in, 4-6
- Offset, Tool, 20-1
- Offsets
 - canceling (G92.1), 11-20
 - canceling selected (G92.2), 11-21
 - changing active, 3-12
 - external, 11-9
 - from PAL, 11-22
 - gauge line, 3-3
 - jogging in, 11-19
 - range verification, 3-10
 - set zero, 11-18
 - Tool Length Axis Select, 20-9
 - using tool position (G92), 11-13
 - work coordinate system, 11-13
 - work coordinate system tables, 3-14
 - zero points (G52), 11-16
- Offsets, Tool
 - assigning offset numbers, 3-2
 - entering diameter values, 3-4
 - entering diameter wear values, 3-5
 - entering length values, 3-2
 - entering length wear values, 3-4
 - entering through PAL, 3-5
 - entering using G10, 3-5
 - entering using measure, 3-5
 - entering using paramacros, 3-5
 - entering using skip cycle, 3-5
 - manually entering data, 3-5
- Operating a Dual-Processing System, 30-2
- Operating a Shared Axis, 30-26
- Operating Modes, Changing, 2-33
- Operator Messages, B-1
- Operator Panel, 2-1
 - Calculator Function, 2-4
 - Keyboard, 2-3
 - monochrome, 2-2
- Operators, Valid Mathematical, 2-5
- Optional Stop, 7-18
- Optional Stop M01, 10-33
- Output code, for Communications, 9-7
- Outputting Part Programs, Errors, 9-18
- Overflow value, 28-44
- Overrides Disabled M49, 10-34
- Overrides Enabled (M48), 10-34
- Overtravel Zone Lines, 8-30
- Overtravels, 12-1
 - Hardware, 12-2
 - programmable zones, 3-21
 - Resetting, 12-13
 - resetting, 4-6
 - Software, 12-3
- Overtravels and Zones (G22, G23, G22.1, G23.1), 12-1

P

- P-word, 10-37
- PAL, Reference Manual, 1-5
- PAL Axis Mover, 14-20
- PAL Display Pages, 8-22
- PAL Offsets, 11-22
- Panel
 - Front Panel, 2-15
 - MTB Panel, 2-12
- parallel cutting tools, 30-31
- Parallel Probing Cycle (G38.1), 27-12
- Paramacro Commands
 - AMP Defined
 - G Macro Call, 28-6, 28-50
 - M Macro Call, 28-51
 - T, S, B Code Macro Call, 28-51
 - Common Parameters, 28-15, 28-35
 - Control Commands, Transfers, 28-7
 - DO END, 28-10
 - GOTO, 28-8
 - IF GOTO, 28-8
 - Local Parameters Assignments, 28-12
 - Non Modal Paramacro Call (G65), 28-45
 - PAL Parameter, 28-33

- Input Flags, 28-33
- Output Flags, 28-34
- Parameter Value Assignment, 28-36
 - Through Programming, 28-38
 - Through Tables, 28-40
 - Using Arguments, 28-36
- System Parameters, 28-15
- WHILE DO END, 28-10
- Paramacro Variables
 - in CALC operation, 2-7
 - Transfer Line, 31-14
- Parameter Table, 10-19
- Parameters, Changing, 8-27
- Parameters, Definition, 10-6
- Parametric Expressions, 28-2
- Parity, for communications, 9-6
- Parking a Dual Axis, 19-3
- Part Production/Automatic Mode, 7-17, 7-23
- Part Program
 - Error Conditions, I/O, 9-18
 - Inputting, 9-9
 - outputting, 9-13
 - selecting, 31-5
 - Verifying, 9-17
- Part Program Directory for Dual-Processing, 30-3
- Part Programs
 - editing, 31-6
 - Editing Offline, 6-2
 - editing with QuickView, 31-13
 - editing with the part program editor, 31-13
 - Transfer Line, 31-3, 31-4
 - creating, 31-9
 - editing, 31-12
 - editing with the part program editor, 31-16
 - reloading part programs, 31-17
 - Transfer Line, customizing, 31-5
- Part Program, creating, 31-9
- Part Rotation
 - Manual (external), 13-6
 - Programmed, 13-1
- Parts Count Display, 2-43
- Password Functions, 2-25
- Passwords
 - assigning, 2-26
 - Entering, 2-31
 - Features Protected, 2-29
- Patch AMP, see AMP manual, 1-5
- Pause, dwell, 14-35
- Peripheral Setup, 9-2
- Peripheral, executing from, 7-5
- Personal Computers, for part programming, 6-1
- Plane Select
 - for program prompts, 5-19
 - Power up Condition, 2-23
 - QuickView, 5-17
- Plane Selection (G17, G18, G19), 13-11
- Plotting Tool Path, 8-24
- Pocket Milling
 - Finishing Cycle G88.2, 22-15
 - Circular Pocket, 22-19
 - Rectangular Pocket, 22-16
 - Slot, 22-20
 - Roughing Cycle G88.1, 22-1
 - Circular Pocket, 22-10
 - Circular Pocket Enlarging, 22-13
 - Rectangular Pocket, 22-2
 - Rectangular Pocket Enlarging, 22-5
 - Slot, 22-8
- Polar Coordinate Programming (G15, G16), 14-21
- Polar Programming Special Cases, 14-25
- Polar Programming, restrictions, 14-24
- Port A, Configuration, 9-2
- Port B, Configuration, 9-2
- Position Display, 8-1
- Position Displays, disable, 14-39
- Positioning a Dual Axis, 19-5
- Positioning and Axis Through PAL, 14-20
- Positioning and Hole Machining Axes, 26-4
- Positioning Axes, 14-1
- Positioning Rotary Axes, 14-12
- Positioning, at feedrate, 14-3
- Positioning, rapid, 14-1
- Post Milling
 - Finishing Cycle G88.4, 23-7
 - Circular Post, 23-11
 - Rectangular Post, 23-8
 - Roughing Cycle G88.3, 23-1
 - Circular Post, 23-5
 - Rectangular Post, 23-2
- Power Off, 2-14
- Power On, 2-14, 2-21
- Power Switch, Power Off, 2-22
- Power Up Display, 8-37
- Power Up Conditions, 2-23
- Precautions on Corner Cutting, 18-20

- Probing
 - Applications (G31), 27-4
 - Applications (G37), 27-7
 - Hole Probing (G38), 27-8, 27-9
 - Parallel Cycle (G38.1), 27-12
 - Parameter Table, 27-15
 - Skip Function (G31), 27-2
 - Tool Gauging, 27-4
 - Probing for adaptive depth, 27-18
 - Probing Parameters Table, 27-15
 - PROC SELECT key, 2-3
 - Process Speed, 8-32
 - Process, changing axes of, 30-28
 - Process, Selecting for Dual-Processing, 30-2
 - Processing, Multi, 30-1
 - Program
 - execution through peripheral, 9-2
 - Selecting, 8-24
 - Program DTG Display, 8-17
 - Program End, for Tapes, 9-8
 - Program Interrupts During Synchronization, 30-11
 - Program Name on Tapes, 10-4
 - Program Names, 10-8, 10-37
 - Program Position Display, 8-3
 - Program Recover, 7-25
 - Program Search, {SEARCH}, 7-10
 - Program Stop M00, 10-33
 - Programmable Acc/Dec, 18-18
 - Programmable Mirror Image, 14-36
 - Programmable Zone Table, 3-21
 - Programmable Zones
 - resetting, 4-6
 - Zone 2, 12-5
 - Zone 3, 12-7
 - Programming
 - absolute, incremental, 13-12
 - adding blocks manually (MDI), 4-11
 - inch/metric, 13-13
 - Programming Aids, {QUICKVIEW}, 5-17
 - Programming Changes to Work Coordinate Values, 11-7
 - Programming Configuration, 10-6
 - Programming in MDI, 4-11
 - Programming Resolution, (Min and Max Axis Motion), 10-21
 - Programs
 - block definition, 10-6
 - block numbers, 5-13
 - calling subprograms, 10-13
 - comments, 5-40
 - copying, 5-41
 - de-selecting active, 7-9
 - deleting, 5-37
 - displaying, 5-39
 - editing online, 5-3
 - encryption, 5-46
 - end of, 10-34
 - Executing, 7-1
 - exiting editor mode, 5-16
 - help prompts, 5-17
 - Inputting from Peripheral, 9-9
 - memory requirements, 6-4
 - Merging, 5-15
 - outputting to peripheral, 9-13
 - protected directory, 5-43
 - renaming, 5-38
 - Retracing Blocks, 7-31
 - Selecting Active, 7-6
 - Selecting an Input Device, 7-5
 - selecting to edit, 5-2
 - simplified geometry, 5-20
 - stopping at specific blocks, 7-2
 - storing on personal computer, 6-12
 - subprograms, 10-12
 - tape format, 10-1
 - teaching through digitizing, 5-28
 - verifying against source, 9-17
 - Programs, Editing Online, 5-1
 - Prompting
 - G Codes, 5-24
 - Milling Cycles, 5-26
 - QuickPath Plus, 5-20
 - Prompts for Programming, 5-17
 - Protectable Directory
 - protected program encryption/decryption, 5-46
 - storing encryption/decryption table to backup memory, 5-49
 - Protected Areas, creating zones, 3-21
 - Protected Program Directory, 5-43
 - Protocol, for communications, 9-6
 - PTO Message, 8-37
 - Publications, related, 1-5
- ## Q
- Quick Check , {QUICK CHECK}, 7-17
 - Changing Quick Check with Graphics, 8-24
 - Quick View Plane, 5-19
 - QuickCheck, 7-19

QUICKPATH Plus and Radius Chamfer Words, 10-22
QuickPath Plus Prompting Patterns, 5-20
QUICKVIEW, 5-17
QuickView, with Transfer Line Cycles, 31-5
QuickView for Dual-Processing, 30-4
QV01, Drilling Cycle, No Dwell/Rapid Out, 31-2, 31-15, 31-26
QV02, Drilling Cycle, Dwell/Rapid Out, 31-2, 31-15, 31-28
QV03, Deep Hole Drill Cycle, Chip Break, 31-2, 31-15, 31-32
QV04, Deep Hole Drill Cycle, Chip Clear, 31-2, 31-15, 31-30
QV05, Right Hand Tapping Cycle, 31-2, 31-15, 31-34
QV06, Right Hand Solid Tapping Cycle, 31-2, 31-15, 31-35
QV07, Left Hand Tapping Cycle, 31-2, 31-15, 31-36
QV08, Left Hand Solid Tapping Cycle, 31-2, 31-15, 31-37
QV09, Boring Cycle, No Dwell/Feed Out, 31-2, 31-15, 31-38
QV10, Boring Cycle, Spindle Shift, 31-2, 31-15, 31-41
QV11, Boring Cycle, Spindle Stop/Rapid Out, 31-2, 31-15, 31-39
QV12, Back Boring Cycle, 31-2, 31-15, 31-43
QV13, Boring Cycle, Dwell/Feed Out, 31-2, 31-15, 31-44
QV14, Single Axis Lift Cycle, 31-2, 31-15, 31-45
QV15, Single Axis Transfer Cycle, 31-2, 31-15, 31-46
QV16, Two Axis Transfer Bar Cycle, 31-2, 31-15, 31-47
QV17, Single Axis Cross Cycle, 31-2, 31-15, 31-48
QV18, Single Axis Feed Cycle, 31-2, 31-15, 31-50
QV19, Two Axis Cross Feed Cycle, 31-2, 31-15, 31-51

R

Radius Offset Table, 3-1
Radius Offset Word, 10-22
Radius, for polar programming, 14-21
Radius, programming for circular, 14-8
Random Tool, 20-11
Range of Motion, Overtravels, 12-1
Range Verification, tool offsets, 3-10
Rapid Feedrate, 18-6
 Override, 18-7
Rapid Feedrate Override, 2-14
Rapid Positioning Mode (G00), 14-1
Rapid Traverse, 8-29
Recall, search with memory, 7-13
Reducing, scaling, 13-14
Reference Line, 3-3

Reflect, mirror, 14-36
Reformat Memory, 2-41
Reloading Part Program Templates, 31-17
Remaining Workpieces, 2-49
Removing an Axis, 4-8
Removing an Axis, 2-43
Renaming Programs {RENAME PROGRAM}, 5-38
Reset, Emergency Stop, 2-24
Reset Operations for Dual-Processing, 30-6
Reset Work Coordinate System, 11-20
Reset, Block, 2-4
Reset, Control, 2-4
Resetting Overtravels, 12-13
Resolution, for Axis Words, 10-21
Retrace Blocks, 7-31
Retrace, block, 2-14
Retract, Jog, 7-28
Return from Home, automatic, 14-32
Return to Alternate Home (G30), 14-34
Return to Home, automatic, 14-30
Reverse Program Execution, 7-31
Revolutions, in dwell, 14-35
Rewind Tape, 10-5
Right Hand Solid Tapping Cycle (G84.1), 26-25
Right Hand Tapping Cycle (G84), 26-23
Rollover Point, rotary axis, 14-12
Rotary Axes, 14-12
Rotating Coordinate System
 External, 13-6
 Programmed, 13-1
Rotating Coordinate System (G68, G69), 13-2
Rotation, Probing for, (G38.1), 27-12
RPM Spindle Control, 10-38
RS-244A, 9-7
RS-358-B, 9-7
RS232-C, 9-5
RS422A, 9-5
Run Time, 2-47
Running a Program, 7-1

S

S-Curve Acc/Dec, 18-17

- S-word, Spindle Speed, 10-38
- Save CRT, 8-39
- Saving Offset Tables, to a part program or external device, 3-17
- Saving Part Programs, 5-16
- Saving Programs, to ODS, 6-12
- Saving programs, 5-16
- Scaling, 13-14
- Scaling, restrictions, 13-19
- Screen Saver, 8-39
- Screens, Custom through PAL, 8-22
- Search
 - Program Search, 7-10
 - Search With Recall, 7-13
- Secondary Home, return, 14-34
- Seconds, for dwell, 14-35
- Select Graph, 8-29
- Selecting a part program Input Device, 7-5
- Selecting a Program to Edit, 5-2
- Selecting a Tool, 10-40
- Selecting Linear Acc/Dec Modes, Using G47, 18-18
- Selecting Linear Acc/Dec Values, Using G48, 18-19
- Selecting Planes, 13-11
- Sensor, adaptive depth, 27-18
- Sequence Numbers, {RENUM PRGRAM}, 5-13
- Sequence Numbers {RENUM PRGRAM}, 10-9
- Sequence Stop {SEQ STOP}, 7-2
- Servo Detach, 4-8
- Servo Disable, 4-8
- Servo Off, 4-8
- Servo, clamp, 14-39
- Set Zero Offset, 11-18
- Setting Communications, 9-1
 - Port Parameters, 9-1
- Shared Axes
 - changing processes, 30-28
 - configuring, 30-29
 - coordinate offsets, 30-29
 - interference checking, 30-13
 - on Dual Processing, 30-26
 - operating, 30-26
 - programmable zones, 30-26
 - restrictions, 30-28
 - tool offsets, 30-30
- Shared Parameters for Dual-Processing, 30-6
- Shifting Work Coordinate Systems, 11-9
- Short Block Acc/Dec
 - Entry and Exit, 18-25
 - G36 G36.1, 18-22
- Single Block, 2-14, 7-4
- Single Digit F-word, 18-12
- Single Digit Feedrate Table, 3-23
- Single-Block, 8-33
- Single-Digit Feedrates for Dual-Processing, 30-5
- Sizing, scaling, 13-14
- Skip and Gauging Functions, 27-1
- Skipping Selected Blocks, 10-10
- Softkey, COPY OFFSET, 3-7
- Softkeys, 2-8, A-1
 - {NEXT PRGRAM}, 7-11
 - {SWITCH LANG}, 8-23
 - ABS, 8-1, 8-6
 - ACCESS CONTROL, 2-25
 - ACTIVE OFFSET, 3-12
 - ALL, 8-1, 8-19
 - AXIS SELECT, 8-1
 - BACKUP ALL, 28-43
 - BACKUP COM1, 28-43
 - BACKUP COM2A, 28-43
 - BACKUP COM2B, 28-43
 - BACKUP INTERF, 30-24
 - BACKUP OFFSET, 3-18
 - BLOCK DELETE, 5-12, 7-1, 10-10
 - BLOCK RETRCE, 2-19
 - BLOCK TRUNC, 5-11
 - CHANGE DIR, 2-31, 5-44
 - CHAR/WORD, 5-5
 - CLEAR ACTIVE, 2-40
 - COORD ROTATE, 13-8
 - COPY PRGRAM, 5-41
 - CYCLE START, 2-19
 - CYCLE STOP, 2-19
 - DE ACT PROGRAM, 7-9
 - DELETE CH/WRD, 5-11
 - DELETE PRGRAM, 5-37
 - DEVICE SETUP, 9-2
 - DIGITIZE, 5-28
 - DISPLY PRGRAM, 5-39
 - DTG, 8-1, 8-12
 - ED PRT INFO, 2-45
 - EDIT PROGRAM, 5-3, 31-7
 - ENTER MESSAGE, 8-38
 - ERROR MESSAGE, 2-37
 - EXIT EDITOR, 5-16
 - F1 F9, 3-24, 30-19, 30-24
 - FRONT PANEL, 2-15

- G CODE, 8-1
- G CODE PROMPT, 5-24
- G CODE STATUS, 8-20
- GRAPH, 8-25
- GRAPH SETUP, 8-28
- JOG AXIS, 2-18
- JOG AXIS +, 2-18
- JOG AXIS -, 2-18
- JOG RETRACT, 2-19
- M CODE, 8-1
- M CODE STATUS, 8-16
- MACHINE INFO, 8-33
- MACRO PARAM, 28-40
- MEASURE, 3-9
- MERGE PROGRAM, 5-15
- MID ST PROGRAM, 7-13, 7-25
- MILL PROMPT, 5-26
- MODIFY INSERT, 5-7
- MORE OFFSET, 3-7
- NCRYPT MODE, 5-46
- OFFSET, 3-1, 3-12
- PASSWORD, 2-26
- PLANE SELECT, 5-19, 5-29
- PROGRAM, 8-1, 8-3
- PROGRAM CHECK, 8-24
- PROGRAM COMMENT, 5-40
- PROGRAM DTG, 8-17
- PROGRAM MANAGE, 5-43
- PROGRAM PARAM, 3-21, 3-24
- PROBE PARAM, 27-16
- PROGRAM DTG, 8-1
- PROGRAM EXEC, 2-19
- PTOM SI/OEM, 8-37
- QPATH PLUS, 5-20
- QUICK CHECK, 7-19, 8-24
- QUICK VIEW, 5-17
- RANDOM TOOL, 20-11
- REFORM MEMORY, 2-41
- RENAME PROGRAM, 5-38
- RENUM PROGRAM, 5-13
- SAVE PARAM, 8-32
- SCREEN SAVER, 8-40
- SEARCH, 5-5, 7-10
- SELECT PROGRAM, 7-6
- SEQ STOP, 7-2
- SET DATE, 2-45
- SET TIME, 2-45
- SET ZERO, 2-18, 11-18
- SET UP NCRYPT, 5-46
- SPLIT ON/OFF, 8-21
- STRING SEARCH, 7-11, 7-15
- SYSTEM SUPPORT, 3-23, 30-19, 30-24
- T PATH DISABL, 8-27
- TARGET, 8-1, 8-9, 8-10
- TIME PARTS, 2-44
- TOOL GEOMET, 3-1
- TOOL MANAGE, 20-22, 20-27, 20-32
- TOOL WEAR, 3-1
- UPDATE & EXIT, 2-28, 5-48
- VERIFY PROGRAM, 9-17
- WORK COORD, 3-14
- ZONE LIMITS, 3-21
- ZOOM WINDOW, 8-33
- Software Front Panel, 2-15
 - Jogging, 2-18
- Software MTB Panel, 2-15
- Software Overtravels, 12-3
- Solid Tapping Cycle (G74.1), 26-12
- Solid Tapping Cycle (G84.1), 26-25
- Spanish, Language Display, 8-23
- Speed Multiply, 2-13
- Speed of Spindle, 10-38
- Speed, cutting feedrates, 18-1
- Speed, rapid, 14-1
- Speed/Multiply Switch, 4-3
- Speeds, using linear feed mode, 14-3
- Spindle Acceleration, 18-22
- Spindle Button, 2-14
- Spindle Control for Dual Process System, 30-12
- Spindle Control for Dual-Processing, Single-Digit Feedrates, 30-5
- Spindle Direction, 2-14
- Spindle Speed, 10-38
- Spindle Speed Override, 2-14
- Spindles, with Transfer Line Cycles, 31-25
- Spiral Interpolation, helix, 14-10
- Split Screen Display, 8-21
- Split Screen Display for Dual-Processing, 30-3
- Standard MTB Panel, 2-12
- Start Code for Tapes, 10-4
- Start Up Message, 8-37
- Start/Rewind Code for Tapes, 10-3
- Step by Step Execution, 7-4
- Stop Bit, for communications, 9-6
- Stop Program End, for Tapes, 9-8
- Stop, Emergency, 2-24
- Stop, feed to Hard (G24), 14-40
- Stops, Hardware Overtravels, 12-2
- Stops, Software Overtravels, 12-3

- Storage Capacity, Memory, 6-4
- Subprogram Call M98, 10-34
- Subprogram Call, (M98), 10-13
- Subprogram End M99, 10-35
- Subprogram Names, 10-8
- Subprogram Nesting, 10-16
- Subprogram Return, (M99), 10-14
- Subprogram, Using, 10-12
- Suppression, of Zeros, 10-18
- surface speed, 10-39
- Switching a Shared Axis Process, 30-28
- Synchronization
 - Coordinating, 30-10
 - Cycle Stop, 30-11
 - M-codes, 30-7
 - MDI Mode, 30-11
 - Multiple Part Programs, 30-7
 - Program Interrupts, 30-11
 - Simple, 30-8
 - With Setup, 30-8
- Synchronization M Codes, 10-35, 10-36
- System Error Messages, B-1
- System Integrator Message, 8-37

T

- T word Ignore, 10-41
- T-word, Tool Selection, 10-40
 - Programming a T-word, 20-1
- Table, Program Encryption, 5-49
- Table of G-codes, 10-26
- Tables
 - backing up offsets, 3-17
 - external offset, 3-14
 - single digit feedrates, 3-23
 - work coordinate system, 3-14
- Tables, setting offsets using measure, 3-9
- Tape, program end, 10-5
- Tape Device Setup, 9-2
- Tape Format, 10-1
- Tape, program name for, 10-37
- Tape, selecting to execute from, 7-5
- Tapes
 - comments, 10-5
 - Inputting Part Programs, 9-9
 - leader section, 10-3

- outputting to, 9-13
- Program End, 9-8
- program name, 10-4
- rewind/stop code, 10-5
- start code, 10-4
- Stop Program End, 9-8
- tape start/rewind code, 10-3
- Tapes, Auto File Name, 9-7
- Tapping Cycle
 - Left and Right Hand, 26-10, 26-23
 - Left Hand Solid, 26-12
 - Right Hand Solid, 26-25
- Tapping Mode (G63), 18-21
- Target Position Display, 8-9
- Teach, through Digitizing, 5-28
- Templates, transfer line
 - Back Boring Cycle, 31-2, 31-15, 31-43
 - Boring Cycle, Dwell/Feed Out, 31-2, 31-15, 31-44
 - Boring Cycle, No Dwell/Feed Out, 31-2, 31-15, 31-38
 - Boring Cycle, Spindle Shift, 31-2, 31-15, 31-41
 - Boring Cycle, Spindle Stop/Rapid Out, 31-2, 31-15, 31-39
 - Deep Hole Drill Cycle, Chip Break, 31-2, 31-15, 31-32
 - Deep Hole Drill Cycle, Chip Clear, 31-2, 31-15, 31-30
 - Drilling Cycle, Dwell/Rapid Out, 31-2, 31-15, 31-28
 - Drilling Cycle, No Dwell/Rapid Out, 31-2, 31-15, 31-26
 - Left Hand Solid Tapping Cycle, 31-2, 31-15, 31-37
 - Left Hand Tapping Cycle, 31-2, 31-15, 31-36
 - Right Hand Solid Tapping Cycle, 31-2, 31-15, 31-35
 - Right Hand Tapping Cycle, 31-2, 31-15, 31-34
 - Single Axis Cross Cycle, 31-2, 31-15, 31-48
 - Single Axis Feed Cycle, 31-2, 31-15, 31-50
 - Single Axis Lift Cycle, 31-2, 31-15, 31-45
 - Single Axis Transfer Cycle, 31-2, 31-15, 31-46
 - Two Axis Cross Feed Cycle, 31-2, 31-15, 31-51
 - Two Axis Transfer Bar Cycle, 31-2, 31-15, 31-47
- Testing Part Programs, dry run mode, 7-21
- Text, Changing Language, 8-23
- Text, in program, 10-10
- Text, Language Mode, 8-23
- Thread Cutting, Solid Tapping, 26-12, 26-25
- Time and Parts Count Display, 2-43
- Time Out, for communications, 9-6
- Time, for dwell, 14-35
- Time, Power-on Overall, 2-46
- Time, Setting, 2-43
- Tool, Gauging (G37), 27-4
- Tool Change (M06), 10-33
- Tool Color for Graphics, 20-29

- Tool Data, Assigning Detailed, 20-25
 - Tool Directory Data, 20-20
 - Tool Gauging Function (G37), 27-1
 - Tool Length Axis Selection, 20-9
 - Tool Length Offset Function (G43, G44, G49), 20-3
 - Tool Management, 20-19
 - Tool Offset Dimensional Parameters, 3-2
 - Tool Offset Table Setup, 3-1
 - Tool Offset Words, 10-22
 - Tool Offsets
 - Activating, 20-8
 - backing up, 3-17
 - changing active, 3-12
 - entering diameter values, 3-4
 - entering diameter wear values, 3-5
 - entering length values, 3-2
 - entering length wear values, 3-4
 - entering using G10, 3-5
 - entering using measure, 3-5
 - entering using paramacros, 3-5
 - entering using skip cycle, 3-5
 - entering using though PAL, 3-5
 - Length, 20-3
 - manually entering data, 3-5
 - offset numbers, 3-2
 - on shared axis, 30-30
 - range verification, 3-10
 - Tool Path, Displaying, 8-24
 - Tool Position, as offset, 11-13
 - Tool Selection, 10-40
 - Torque, feedrate control (G25), 18-9
 - Trailing Zero Suppression (TZS), 10-18
 - Transfer Line Cycles
 - creating, 31-9
 - customizing, 31-5
 - Description, 31-3
 - editing, 31-12
 - Example, 31-4
 - M-Codes, 31-16
 - Paramacro Variables, 31-14
 - reloading templates, 31-17
 - replacing, 31-17
 - Running, 31-25
 - selecting, 31-6
 - using G17/G18/G19, 31-25
 - using G90/G91, 31-25
 - using G98/G99, 31-25
 - using spindles, 31-25
 - Transfer line cycles
 - Back Boring Cycle, 31-2, 31-15, 31-43
 - Boring Cycle, Dwell/Feed Out, 31-2, 31-15, 31-44
 - Boring Cycle, No Dwell/Feed Out, 31-2, 31-15, 31-38
 - Boring Cycle, Spindle Shift, 31-2, 31-15, 31-41
 - Boring Cycle, Spindle Stop/Rapid Out, 31-2, 31-15, 31-39
 - Deep Hole Drill Cycle, Chip Break, 31-2, 31-15, 31-32
 - Deep Hole Drill Cycle, Chip Clear, 31-2, 31-15, 31-30
 - Drilling Cycle, Dwell/Rapid Out, 31-2, 31-15, 31-28
 - Drilling Cycle, No Dwell/Rapid Out, 31-2, 31-15, 31-26
 - Left Hand Solid Tapping Cycle, 31-2, 31-15, 31-37
 - Left Hand Tapping Cycle, 31-2, 31-15, 31-36
 - Right Hand Solid Tapping Cycle, 31-2, 31-15, 31-35
 - Right Hand Tapping Cycle, 31-2, 31-15, 31-34
 - Single Axis Cross Cycle, 31-2, 31-15, 31-48
 - Single Axis Feed Cycle, 31-2, 31-15, 31-50
 - Single Axis Lift Cycle, 31-2, 31-15, 31-45
 - Single Axis Transfer Cycle, 31-2, 31-15, 31-46
 - Two Axis Cross Feed Cycle, 31-2, 31-15, 31-51
 - Two Axis Transfer Bar Cycle, 31-2, 31-15, 31-47
 - Transfer Line Templates
 - mutually exclusive features, 31-25
 - Optional Fields, 31-25
 - Required Fields, 31-25
 - Transmit Key, 2-3
 - Transpose, mirror, 14-36
 - Travel Limit, hard stop (G24), 14-40
 - Travel Limits, Resetting, 12-13
 - Trigonometric Functions, 2-6
 - TRVRS, 2-14
 - twin axes, 30-31
 - TZS, Using, 10-18
- ## U
- Uniform Acc/Dec, 18-16
 - Uploading Part Programs to ODS, 6-12
 - Using Interference Checking, 30-12
- ## V
- Values, Interpretation of Zeros, 10-18
 - Varying axis feedrate, torque mode (G25), 18-9
 - Verifying Programs, against source, 9-17
- ## W
- Wear Offset Table, 3-2, 3-6
 - Word Descriptions, 10-22

Word Descriptions and Ranges, 10-19
Word Format, Zero Suppression, 10-18
Word Format and Functions, 10-17
Word, Definition, 10-6
Work Coordinate, Changing or Offsetting, 21-49
Work Coordinate System
 Defining, 11-5
 External Offset, 11-9
 Offsetting, 11-13
Work Coordinate System (G54 G59), 11-4
 Altering using G10, 11-7
Work Coordinate System Offset Tables, 3-14
 External Offsets, 3-14
 Setting, 3-15
 Zero Point Parameters, 3-14
Work Coordinate Systems
 altering external offset, 11-11
 canceling offsets on, 11-20, 11-21
 jogging in an offset, 11-19
 offsetting all, 3-14
 PAL offsets, 11-22
Workpieces Completed, 2-49

Workpieces Cut, 2-47

Z

Zero Point, table for defining work coordinate system, 3-14
Zero Point, homing the axes, 14-29
Zero Point, Machine Coordinate, 11-1
Zero Point, Work Coordinates, 11-4
Zero Points, offsetting, 11-16
Zero Set Softkey, 11-18
Zero, Manual Homing, 4-8
Zeros, How Interpreted, 10-18
Zones
 on shared axis, 30-26
 programmable tables, 3-21
Zones, Programmable, 12-5
Zones, Resetting, 12-13
Zones, Resetting Overtravels, 4-6
Zooming, 8-33

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 8520-UM513A-EN-P - October 2000

Supersedes Publication 8520-5.1.3 - August 1998

PN 176957

Copyright 2000 Allen-Bradley Company, Inc. Printed in USA

www.EngineeringBooksPdf.com