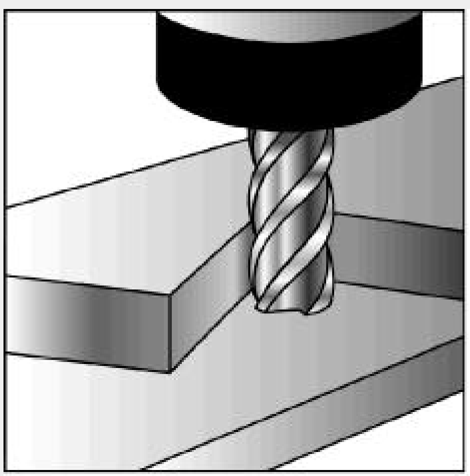
Allen-Bradley

9/Series CNC
Grinder



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.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attention statements help you to:

- identify a hazard
- avoid the hazard
- recognize the consequences

Important: Identifies information that is critical for successful application and understanding of the product.

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 4
Manual/MDI Operation Modes

Chapter 5
Editing Programs On Line

Chapter 6
Editing Part Programs Off Line (ODS)

Chapter 7
Running a Program

Chapter 8
Display and Graphics

Chapter 9
Communications

Chapter 10
Introduction to Programming

Chapter 11
Coordinate Control

Chapter 12
Axis Motion

Chapter 13
Wheel Length Offsets

Chapter 14
Angled-Wheel Grinding

Chapter 15
Dresser/Wheel Radius Compensation

Chapter 16
Surface Grinding Fixed Cycles

Chapter 17
Cylindrical Grinding Fixed Cycles

Chapter 18
Turning Operations

Chapter 19
Skip and Gauge Probing Cycles

Chapter 20
Paramacros

Chapter 21
In-process Dresser

Chapter 22
Program Interrupts and Dressing Interrupts

Appendix A
Softkey Tree

Appendix B
Error and System Messages

Appendix C
G-Code Table

Chapter 1

Using This Manual

1.0 Chapter Overview

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

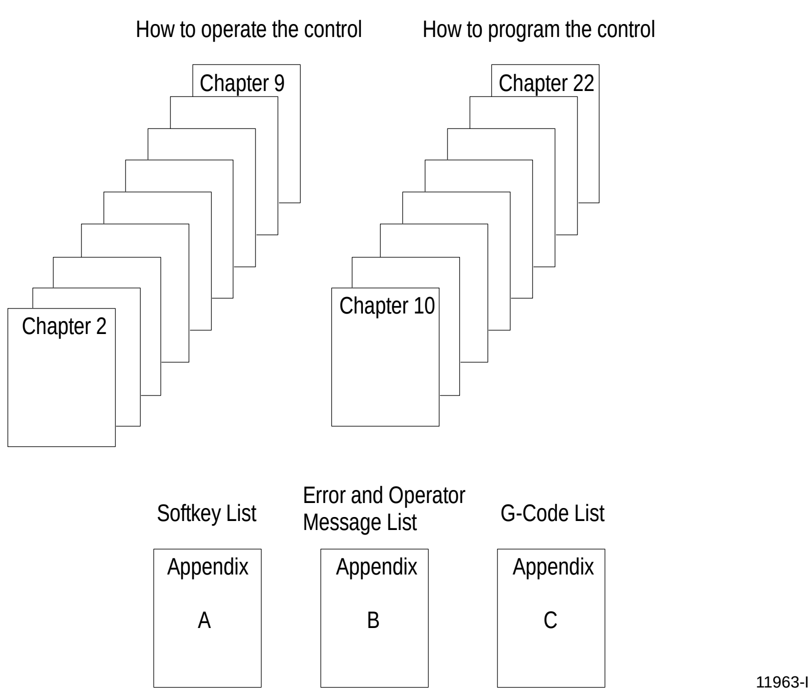
- how the manual is written and what fundamentals are presumed to be understood by the reader
- how the manual is organized and what information can be found in it
- definitions for certain key terms

1.1 Audience

We wrote this manual for operators and programmers who use Allen-Bradley controls. We assume that you are familiar with the basic operation and programming of a CNC.

1.2 Manual Design

This manual has a basic operation section, a programming section and three appendices:



1.3 What This Manual Contains

This table contains a brief summary of each chapter.

Chapter	Title	Summary
1	Manual Overview	Manual overview, intended audience, definition of key terms, how to proceed.
2	Operating the Control	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	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 program 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 and 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 Control	The different coordinate systems and offsets to them. Also sections on plane selection, overtravels, absolute/ incremental and inch/metric modes.
12	Axis Motion	G-words that define how the wheel is positioned to the endpoint of a move. Also sections on spindle control, QuickPath Plus™ programming, dwell, and mirroring.
13	Wheel Length Offset	Selecting a wheel. Activating and deactivating wheel length offsets.
14	Angled Wheel Grinding	Description and use of cylindrical grinders with an angled wheel. The angled wheel is a non-orthogonal wheel axis.
15	Dresser/wheel Radius Compensation	A description of the dresser/wheel radius compensation feature that offsets for different dresser/wheel diameters.
16	Surface Grinding Cycles	Description and use of the fixed cycles (canned cycles) for surface grinders and the G-words and parameters used to define them.
17	Cylindrical Grinding Cycles	Description and use of the fixed cycles (canned cycles) for cylindrical grinders and the G-words and parameters used to define them.
18	Turning Operations	Rough contouring routines and threading including tapered and multi-lead threading.
19	Skip and Gauging Cycles	Description and use of the 9/Series probing features. Included in this is the wheel measuring gauge feature.
20	Paramacros	Description and use of paramacros, including calling, arithmetic functions, looping, decision making, etc.
21	In-Process Dresser	Description on how the in-process dresser operates. The in-process dresser interaction with features such as offset tables and the in-process dresses table are also covered.
22	Program interrupts / Dressing Interrupts	A description of the program interrupt and dressing interrupt features. These features are used to call a subprogram or paramacro program whenever a signal corresponding to that program is sent to PAL by the operator or requested by a grinder cycle.
Appendix A	Softkeys	A description of 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 error/warning messages, each with a brief description.
Appendix C	G-Code Tables	A listing of the G-codes used to program the 9/Series controls.

1.4 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 can differ from the basic information provided in this manual depending on the configuration of your grinder machine controlled by the CNC. For details, see the manuals prepared and supplied by your system installer.
- You can purchase some of the softkey functions and features as options on the 9/Series control. This manual assumes you have all of the optional features.
- Explanations and illustrations assume a 2-axes cylindrical grinder configuration. This means the movement of the grinding wheel on a rotating part or the movement of the grinding wheel on the fixed dresser. Explanations and illustrations were not specifically written for an angled wheel cylindrical grinder though most will apply to the cylindrical grinder configuration.
- The control accepts several different alphabetic characters for expressing numerically controlled axes. This manual uses Z and X for the first and second axes on the basic coordinate system. The integrand name for these axes is K and I respectively. Cylindrical angled wheel grinders should also assume Z (real) and X (virtual) axes for the first and second axis on the basic coordinate system with W (real) however, being the actual physical wheel axis.
- The term AMP is an abbreviation for Adjustable Machine Parameters. These parameters are used to match the control to a specific machine. AMP configuration is usually done by your system installer.
- Program examples are given as radius values. Assume the control is in radius programming mode unless stated otherwise.
- Names between the [ ] symbols are <u>keys</u> found on the operator panel.
- Names between the { } symbols are <u>softkeys</u> found below the CRT.
- Names between the < > symbols are <u>switches and buttons</u> found on the standard MTB panel.
- The term PAL is an abbreviation for Programmable Application Logic. This is a ladder logic program that processes signals between the CNC and the grinder. It is usually programmed by your system installer.

1.5 Terms and Conventions

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

Term	Description
AMP	Adjustable Machine Parameters
backup	Memory storage area in the control that does not require battery power to be maintained
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 9/240, 9/260, or 9/290 Computerized Numerical Control
Dresser/Wheel	applies to processes that are interchangeable between the dresser or grinding wheel
EPROM	Erasable Programmable Read Only Memory
E-STOP	Emergency Stop
HPG	Hand Pulse Generator
I/O	Input/Output
MDI	Manual Data Input
modal	an operating condition that remains in effect on the control until canceled or replaced
MTB	Machine Tool Builder
ODS	Offline Development System
PAL	Programmable Application Logic
RAM	Random Access Memory
softkeys	the row of keys directly below the screen
system installer	the company or contractor responsible for installing this control on the machine

1.6 Warnings, Cautions, and Important Information

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

Information that is especially important is indicated in these ways:

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

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

1.7 Related Publications

For more information about Allen-Bradley controls, see these publications:

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

Chapter 2

Operating the Control

2.0 Chapter Overview

This chapter covers the basics necessary for operation of the Allen-Bradley 9/Series control. Major topics covered in this chapter include:

2.1 Operator Panel Operations

Use the operator panel to:

- display a part program
- display control status and wheel position
- edit a part program
- display and enter wheel offset data
- display the status of input/output signals
- display and enter programmable zone boundaries
- set the level of protection for:
  - part programs
  - wheel offset data
  - AMP data

You can perform other operations by using the operator panel. They are covered in the remaining chapters of this manual.

Figure 2.1 shows the monochrome operator panel.

Figure 2.1
Monochrome Operator Panel

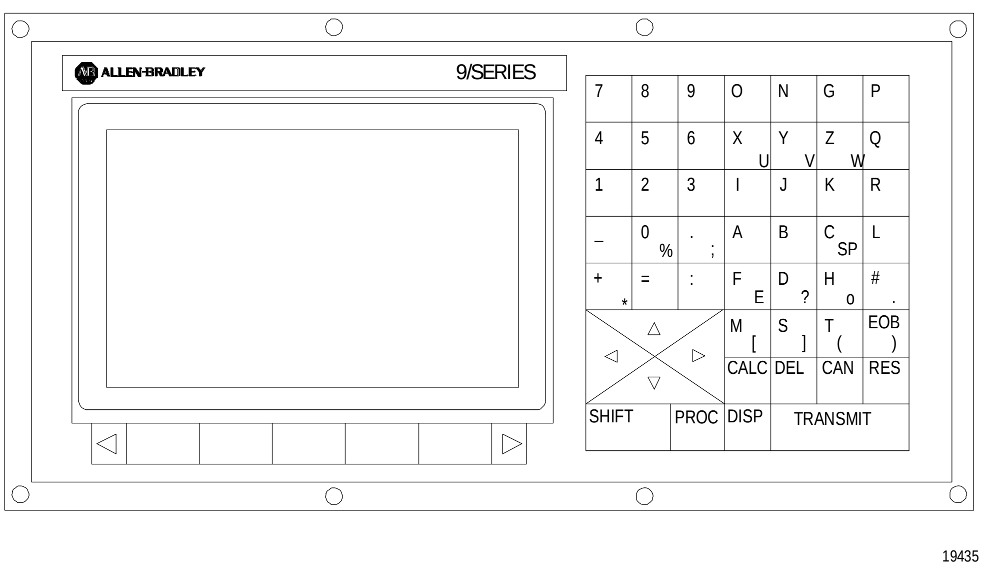
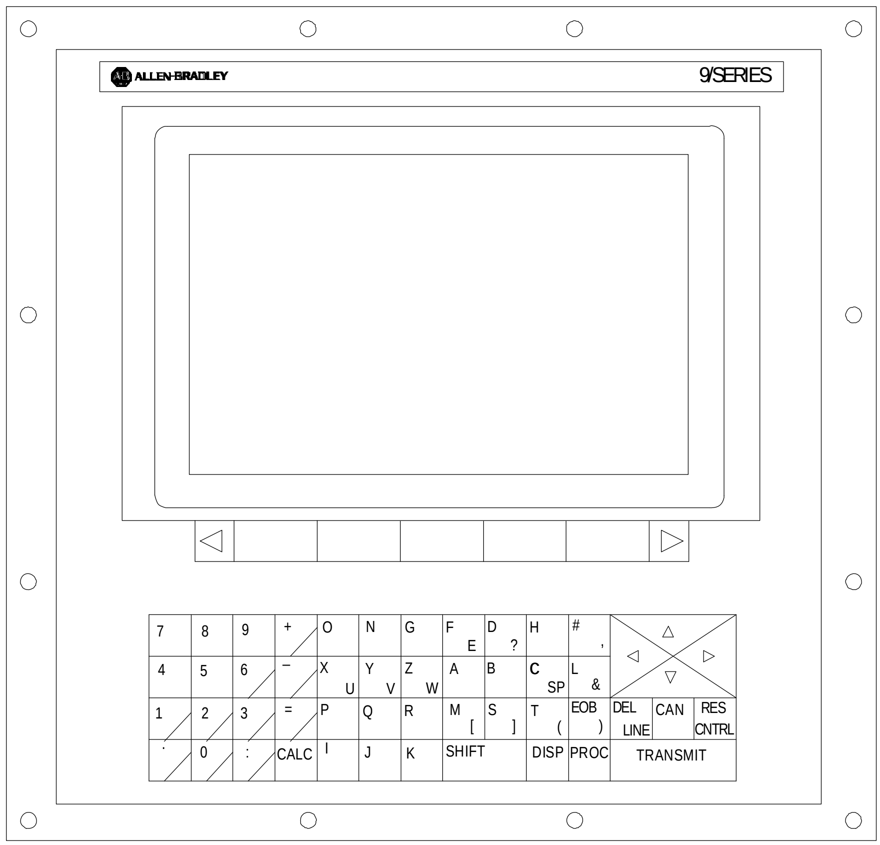


Figure 2.2 shows the color operator panel. It has keys and softkeys identical to the monochrome operator panel in a slightly different configuration.

Figure 2.2
Color Operator Panel



19436

2.1.1 Using the Keyboard

Table 2.A lists the functions of keys on the operator panel keyboard. The names of operator panel keys appear between [ ] symbols.

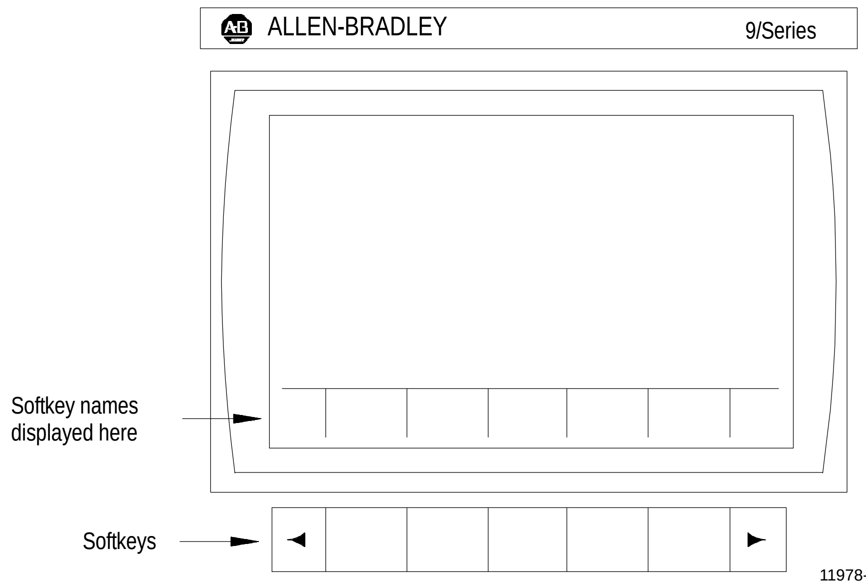
Table 2.A
Key Functions

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. These keys are referred to 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 that data is being input on (normally screen lines 2-3).
[SHIFT] and ↑ or ↓	Press the up and down cursor keys while holding down the [SHIFT] key to roll the display page forward or backward.
Calculation Key [CALC]	Press this key to enable a calculator-type function on the control. Basic mathematical expressions can be evaluated using this feature. For details on Calculator Function, see page 2-48.
Delete Key [DEL]	Press this key to delete the character to the left of the cursor on input lines.
[SHIFT] and [DEL]	Press this key to delete all keyed-in data currently displayed on input lines.
[CAN]	Clears the most currently generated, active error message.
Transmit Key [TRANSMIT]	The data entered and displayed on the input lines (e.g., a password or a program block) is sent to the control when you press the [TRANSMIT] key.
Block Reset [RESET]	This performs a block reset. For details on Block Reset, see page 2-36.
Control Reset [RESET] + [SHIFT]	Press the reset key while holding down the shift key to perform a control reset. For details on Control Reset, see page 2-36.
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.
Process Select [PROC SELECT]	Use this key to select the different processes.
[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.

2.1.2 Softkeys

You access the various software features and functions of the control through softkeys. Softkeys are the row of 7 keys located at the bottom of the CRT as shown in Figure 2.3. They let you move through the control's software. The control displays the function of each softkey on the CRT directly above the softkey. In this manual, softkey names appear between the { } symbols.

Figure 2.3
Softkey and Softkey Name Locations



In this manual, we often describe softkeys as being on a certain level, e.g., softkey level 3. We use the level of the softkey to determine the location or necessary path to reach that particular softkey. 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 of the softkeys and their respective levels, see 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 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 available on that level.

Important: When the number of softkey functions on that level does not exceed 5, the continue softkey is not available.

For example:

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

When you reach softkey level 1, the above set of softkeys appears. Press the continue softkey {→} to display the remaining softkey functions on softkey level 1.

(softkey level 1)

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

On softkey level 1, the control does not display the exit softkey since the softkeys are already on softkey level 1.

We explain the softkey functions for level 1 and 2 in appendix A, and level 3 or higher 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.

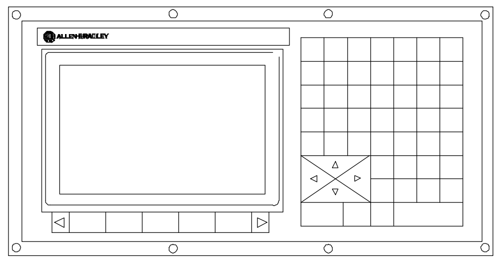
You can purchase some of the softkey functions as optional features. This manual assumes that you purchased all available optional features for your machine. If you did not purchase an option, the softkey is blank.

Some features accessed through a softkey can be password protected. When a feature is password protected, the softkey that accesses that feature is no longer displayed.

2.1.3 Using the CRT

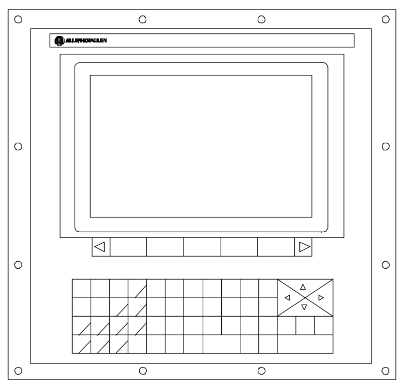
Your control has one of these monitors:

- 9-inch monochrome monitor



19435

- 12-inch color monitor



19436

Both have identical displays and graphics capabilities.

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

Line	Area	Content
1	machine/system message	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.
2 - 3	input lines	When data is entered using the keyboard, the control displays the entered characters on these lines until you press the `[TRANSMIT]` key. Some screens can have only line 2 as an input line.
4 - 20	data display	The control displays axis position data, listing of the part program, wheel length offset data, G/M/H/T/F/S/D codes, graphics, and other data in this area as determined by the selected display.
21 - 22	PAL message	The control displays any messages generated by the control's PAL program in this area.
23 - 25	softkey display	The control displays the currently available softkey functions in this area.

2.2 The MTB Panel

Figure 2.4 shows the MTB panel. Table 2.B lists the selections on this panel. Your system may contain optional or custom MTB panels different than the one shown below. See the documentation prepared by your system installer for details.

We show selection names on the MTB panel between the < > symbols when referred to in this manual. Most selections on the MTB panel are configured by your system installer’s PAL program. This manual assumes that PAL has been written as intended for normal operation. If a selection does not work the way it is described in this manual, see documentation prepared by your system installer.

Figure 2.4
Push-Button MTB Panel

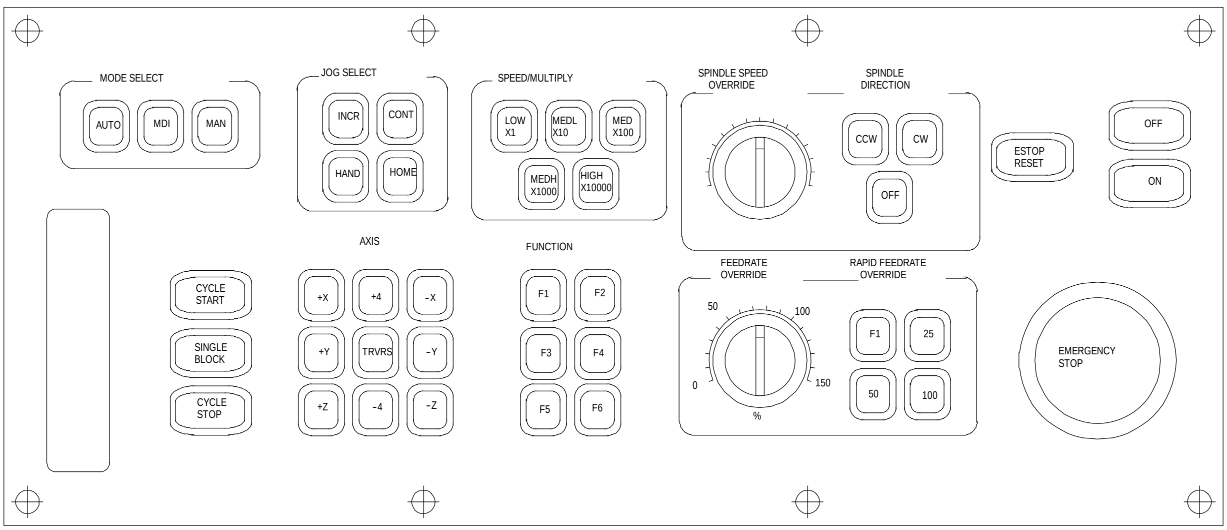


Table 2.B
Selections on the MTB Panel and How They Work

Switch or Button Name	How It Works (shaded = Default for Push-Button MTB Panel)
MODE SELECT	Selects the operation mode AUTO -- automatic mode MANUAL -- manual mode MDI -- manual data input mode
JOG SELECT	Selects the jog method to be active in manual mode HANDWHEEL -- HPG (hand pulse generator) jog INCREMENTAL -- incremental jog CONTINUOUS -- continuous jog HOME -- machine home
SPEED/MULTIPLY	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: • 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 distance by a factor set in AMP by your system installer. Your system installer sets a value for the selections. The incremental jog speed 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. Important: The values for the different <SPEED/MULTIPLY> selections are configured by your system installer.

Table 2.B
Selections on the MTB Panel and How They Work (continued)

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 OVERRIDE> 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 retract 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: You can disable many of the override switch settings by programming the correct M code or by setting a particular paramacro parameter. See 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 your control uses a standard MTB panel (described on page 2-8) or some other custom MTB panel, the requests for operations from the panel take priority. This means that requests from the software MTB panel are ignored if a request from a standard or custom MTB panel is sent.

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

The software MTB panel controls these features:

Feature	Function
Mode Select	Selects either AUTO, MDI, or MANUAL modes as the current operating mode of the control.
Rapid Traverse	Replaces the feedrate when executing a continuous jog move with the rapid feedrate.
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. This switch 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 your system installer.
Spindle Direction	Selects spindle rotation, clockwise (CW), spindle stop (OFF), or counterclockwise (CCW). The front panel 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 control in dry run mode. This replaces feedrates with the dry run feedrates. See page 7-21 for details.
Block Delete	Allows activation of the block delete feature "/ or /1". Block delete 2 - 9 are not available with the `{FRONT PANEL}`. See page 7-2 for details.
M-Function Lock	Allows select M-, S-, T-, and B-codes to be ignored. See page 7-2 for details.
Optional Stop	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 page 10-31 for details on M01.
Single Block	Causes the control to execute or check one block of a part program or MDI entry each time you press the `<CYCLE START>` button when this feature is on. WARNING: Single block execution is not possible during reciprocation.
Mirror Image	Mirrors the axis commands in the part program around the selected axis. See page 12-79 for details.
Axis Inhibit	Prevents axis commands from being executed. The control simulates axis motion on inhibited axes using Acc/Dec and feedrates; however, no actual axis motion for the inhibited axes is generated. See page 7-20 for details.

(continued on next page)

The software MTB panel controls these features: (continued)

Feature	Function
Jog the Axes	Allows manual motions to be performed in any one of the jogging modes. You cannot perform multi-axis jogs using the software front panel feature. See page 4-2 for details.
Set Zero	Changes the wheel's current position in the work coordinate system to 0 for the selected axis. This is done by shifting the work coordinate system. See page 12-89 for details.
Block Retrace	Allows up to 15 part program blocks to be retraced during program execution and allow the wheel to automatically re-execute these blocks. See page 7-30 for details.
Jog Retract	Allows the wheel to be manually jogged away from the part and then automatically returns the wheel to the part by retracing the jogged moves. Up to 15 jog moves can be remembered. See page 7-27 for details.
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. WARNING: During reciprocation, pressing `<CYCLE STOP>` does not necessarily stop the reciprocating axis.

Software MTB Panel Screen

Use the software MTB panel screen to specify the status of the MTB features. To access the software MTB panel screen:

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	SWITCH LANG		→

You see the software front panel screen displaying the current status of the alterable features:

```
       SOFTWARE FRONT PANEL

MODE SELECT:            MDI
RAPID TRAVERSE:         OFF
FEEDRATE OVR:           0%
RAPID FEEDRATE OVR:     0%
SPINDLE DIRECTION:      OFF
SPINDLE SPEED OVR:      50%
DRY RUN MODE:           OFF
BLOCK DELETE:           OFF
M-FUNC LOCK:            OFF
OPTIONAL STOP:          OFF
SINGLE BLOCK:           OFF
MIRROR IMAGE:           X
AXIS INHIBIT:           XZ

       USE CURSOR FOR SELECTION
```

↑	JOG AXIS	PRGRAM EXEC				

2. Press the up and down cursor keys to select the feature to change. The value of the selected feature appears in reverse video.

3. Press the left and right cursor keys to alter the value of the selected feature.

   When you select the mirror image and axis inhibit features, the softkey names change to the axis names. Press the softkey that corresponds to the axis (or axes) to which you want to assign these features.

Jog Axis Screen

After accessing the software front panel screen and selecting the various features for your application, you can use the jog axis screen to:

- jog the axes of the control
- shift the current work coordinate system to force the current wheel position to be the zero point of the work coordinate system

To jog the axes of the control:

1. Press the {JOG AXIS} softkey. The {JOG AXIS} softkey is only available when mirror image or axis inhibit are not in reverse video. If one these features is in reverse video, press the up cursor key.

```
(softkey level 2)
   ↑   | JOG    | PRGRAM |        |        |        |
       | AXIS   | EXEC   |        |        |        |
```

You see the jog axis screen:

```
                                                       E-STOP
     PROGRAM    [INCH]    |   F      0.000 MMPM
       R X       0.000    |   S      0.0
         Z       0.000    |   T 0
                          |   FILENAME
                          |   SUB NAME

      MEMORY        MAN   STOP

        AXIS SELECT:         X
        JOG SELECT:          CONTINUOUS
        SPEED/MULTIPLY       LOW
        HPG NUMBER           0

   ↑   | SET    |  JOG   |  JOG   |        |        |
       | ZERO   | AXIS + | AXIS - |        |        |
```

You can select the:

- axis to jog
- type of jog
- speed multiply value (see manual operating mode on page 4-1)
- HPG number (if HPG has been selected as the type of jog)

2. Use the up and down cursor keys to select a parameter and the left and right cursor keys to alter the value assigned to that parameter.

3. Jog the selected axis in the selected direction by:

If jog type is:	Then:
Not HPG	press the softkey that corresponds to the direction on the selected axis to jog
HPG	the direction of HPG rotation determines the direction to jog

See page 4-2 for details on jogging an axis.

Set Zero

If you want to shift the current work coordinate system to force the current wheel position to be the zero point of the work coordinate system, press the {SET ZERO} softkey.

The current axis position becomes the zero point of the currently active work coordinate system.

The {SET ZERO} softkey does not function when the control is in handwheel mode. See page 11-18 for details on set zero operations.

Program Execute Screen

After accessing the software front panel screen and selecting the various features for your application, you can use the program execute screen to perform a:

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

Important: Before you press the {PRGRAM EXEC} softkey, you should have selected a program for automatic execution as described on page 7-6.

To perform one of these options:

1. Press the {PRGRAM EXEC} softkey.

(softkey level 2)

↑	JOG AXIS	PRGRAM EXEC				

You see the program execute screen:

```
                                                  E-STOP
PROGRAM   [ MM  ]        F       0.000 MMPM

  R X       0.000        S       0.0
    Z       0.000        T 0
                         FILENAME
                         SUB NAME

 MEMORY       MAN        STOP
```

↑	BLOCK RETRCE	JOG RETRCT	CYCLE START	CYCLE STOP		

2. Press the softkey that corresponds to the selected option.

To perform this operation:	Press:
Cycle Start or Cycle Stop	the softkey that corresponds to the desired feature. Details on these features are described earlier in this chapter.
Block Retrace	the {BLOCK RETRCE} softkey; each time you press the {BLOCK RETRCE} softkey the control automatically retraces the previously executed block. Pressing {CYCLE START} returns the wheel to the start point of block retrace. Details on block retrace can be found in chapter 7. (softkey level 3) ↑ | BLOCK RETRCE | JOG RETRCT | CYCLE START | CYCLE STOP | |
Jog Retract	the {JOG RETRCT} softkey. This calls up the jog retract screen. Select the axis, type of jog, speed multiply value (see page 4-1), and HPG number (if HPG has been selected as the type of jog) by using 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. (softkey level 3) ↑ | BLOCK RETRCE | JOG RETRCT | CYCLE START | CYCLE STOP | | the {JOG AXES+} or {JOG AXES-} softkeys to select the direction to jog the axis. Press the exit {↑} softkey when the jog retract is complete. The grinding wheel automatically retraces the jog moves made when the {CYCLE START} softkey is pressed. Details on jog retract can be found in chapter 7. (softkey level 4) ↑ | | JOG AXES+ | JOG AXES- | | |

2.4 Powering the Control

This section describes the procedures for turning on and off power to the control. See the documentation prepared by your system installer for more specific procedures.

2.4.1 Turning On Power

Follow this procedure to turn on power to the control:

1. Visually make sure that the control and the machine are in normal operating condition.

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. Make sure that characters appear on the screen within approximately 10 seconds.

Important: If, after you press the power <ON> button, the screen does not display characters within a reasonable warm up period (about 10 seconds), immediately press the power <OFF> button and contact service personnel.

After power has been turned on, the control displays the power turn-on screen. To activate the main menu, press the [TRANSMIT] key.

You see the main menu screen:

```
                                                  E-STOP
   PROGRAM   [ MM  ]    F  00000.000 MMPM

      R X  00000.000    S

        Z  00000.000    T  12345

                        FILENAME

                        SUB NAME 9999

     MEMORY      MDI  STOP

      (PAL messages)

        PRGRAM  OFFSET  MACRO   PRGRAM  SYSTEM   --->
        MANAGE          PARAM   CHECK   SUPORT
```

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 Off Power

Turn off power to the control when it is not used for an extended period of time.

To turn off power to the control:

ATTENTION: 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. Press the <EMERGENCY STOP> button.

2. Make sure that power is turned off to all peripheral devices (tape reader, tape punch, etc.) that are connected to the control.

3. Press the power <OFF> button.

2.5 Control Conditions at Power-Up

The control assumes a number of initial operating conditions at power-up or at a control reset operation (see page 2-18). These are:

- Initial Password Access is 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, the control defaults to the next lower level that is a power-up level (see page 2-23 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 your system. If either of these is the case, an appropriate error message occurs.

- The control defaults to <u>one</u> 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
12	G54 -G59.3 work coordinate system
13	G61 Exact Stop Mode G62 Auto Corner Override Mode G64 Cutting Mode
15**	G15 Angled Wheel Transform Off G16.3 Angled Wheel Normal Motion G16.4 Angled Wheel 2 Step Motion
17	G96 CSS On G97 CSS Off
18	G07 Radius Programming Mode G08 Diameter Programming Mode
20	G39 Wheel/Radius Comp Linear G39.1 Wheel/Radius Comp Rounding
22	G36 Short Block Feed Clamped G36.1 Short Block Full Feed

* 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.

** Only available on cylindrical grinders

To show the current operating conditions at any time, access the G-code status screen as described on page 9-2. If this is done 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 your machine) to stop operations regardless of the condition of the control and the machine.

ATTENTION: To avoid damage to equipment or hazard to personnel, your 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's subprocessor board. This disables the axis drives and the spindle drive circuits, provided both are connected to this terminal. See your integration manual or the documentation prepared by your system installer for details.

If equipped with the standard MTB panel, the following occurs automatically after pressing 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 be resumed at the point of interruption. See the mid-program start feature described in chapter 7.

2.6.1 Emergency Stop Reset

After the control has entered the emergency stop state, you must perform an E-Stop reset before program execution can continue.

ATTENTION: 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. You can release the locked button 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 resets the program when you perform an E-Stop reset (provided AMP is configured to perform a control reset or E-Stop reset). Assuming that you perform a control reset, program execution begins from the 1st block of the program when you press <CYCLE START>. If the current axis position prohibits this, the operator can manually jog the axes clear, or consider executing a mid-program start (see page 7-13). If you do not perform a control reset or E-Stop reset, the control aborts the remainder of the program block being executed when E-Stop took place, and a <CYCLE START> begins program execution at the next block.

Important: If you do not eliminate the cause of the E-Stop, the circuit connected to the E-STOP STATUS terminals on the sub-basic board remains open, and the emergency stop state is not reset even when you press the <E-STOP RESET> button.

Reciprocation stops when an emergency stop condition occurs. No motion occurs when the emergency stop is reset. If your control’s AMP is configured such that the control is <u>not</u> reset after an E-Stop reset, then typically reciprocation resumes with the next cycle start. This allows part programs to be resumed immediately without having to first restart reciprocation. For details, see the documentation prepared by your system installer.

2.7 Access Control

Access control lets your system installer assign different functions of the control to different users by means of a password. See page 2-28 for a list of the functions that can be protected on the control.

Each protectable function is assigned an access level that is made active when the operator enters his 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. You can assign a total of 8 passwords.

Important: If you do not want to use password protection, 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. You can also disable password protection by assigning a level as the power-up level using the "POWER-UP LEVEL" parameter as described on page 2-29.

2.7.1 Assigning Access Levels and Passwords

This section shows you how to:

- set the functions assigned to a particular access level
- change the functions assigned to a particular access level
- change the password used to activate that access level

Important: You can only assign functions or passwords to another access level if:

- The user attempting to make the change has a higher access level than the access level he is attempting to change. This means that if a user's password is assigned to access level 6, that user can change only 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's functions, access level 6 must have access to any functions that are changed. For example, if an access level 6 user does not have access to {SYSTEM SUPORT}, he cannot assign or remove {SYSTEM SUPORT} from access level 7
- The current user's password must allow access to the {ACCESS CONTRL} 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	SWITCH LANG		→

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}.

2. Press the {ACCESS CONTRL} softkey. This displays the access control screen (Figure 2.5).

(softkey level 2)

↑				ACCESS CONTRL		

Figure 2.5
Access Control Screen

```
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  REFORMAT MEMORY    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 PARAMET    32
   16  AMP               33
   17  PARAMACRO PROG    34
```

	UPDATE & EXIT	01	02	03	04	→

The softkey names change to display the 8 access levels and 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 appear.

3. Press the softkey that corresponds to the access level for which you want to assign access to functions. The pressed softkey appears in reverse video. The password name assigned to that access level is moved to the "PASSWORD NAME."

(softkey level 3)

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

	UPDATE & EXIT	05 (NAME)	06 (NAME)	07 (NAME)	08 (NAME)	→

	STORE BACKUP					→

Important: If you want to change the functions available to an access level that is equal to or higher than the current user's access level, the error message "ACCESS TO THIS LEVEL IS NOT ALLOWED" appears. A user cannot change the features that are assigned to his current access level or any level that is higher than his own.

4. If you want to:

   a. enter or change the password (NAME) for the selected level, edit the password (NAME) next to the "PASSWORD NAME" prompt by using the input cursor as described on page 2-41.

   b. change the functions for this password, move on to step 5.

   c. 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 appear in reverse video on the access level screen.

5. Use the up, down, right, and left cursor keys to select the functions to change for that access level. The selected function appears with a flashing > to the left of the function.

6. Press the [TRANSMIT] key to toggle the function between accessible and inaccessible for that access level.

Important: If you want 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" appears. Only features that are accessible to the current user's access level can be selected as accessible or inaccessible to a lower user's access level.

7. 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)	→

UPDATE & EXIT	05 (NAME)	06 (NAME)	07 (NAME)	08 (NAME)	→

STORE BACKUP					→

2.7.2 Password Protectable Functions

This section describes the functions on the 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 appears in reverse video on the access control screen (see Figure 2.5).

You control access to the functions in Table 2.C by using passwords. Table 2.C describes the function that is enabled (the operator can perform them) if the parameter name appears in reverse video. If the function does not appear in reverse video, the function is protected and cannot be accessed by the operator.

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.C
Password Protectable Functions

Parameter Name	Function becomes accessible when parameter name appears 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. • {SELECT PRGRAM} — 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 PRGRAM} — 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 PRGRAM} — Disable the currently active program from execution. • {SEQ STOP} — Choose a sequence number for automatic program execution to stop at.
2) PROGRAM MANAGE	• {ACTIVE PRGRAM} — All of the functions in 1 above provided ACTIVE PROGRAM (number 1 above) is also selected. • {EDIT PRGRAM} — Edit an existing program or create a new program. • {DISPLY PRGRAM} — Display a program using the display function. • {COPY PRGRAM} — Copy a program to or from memory. • {VERIFY PRGRAM} — Verify that two programs are identical using the verify function. • {COMENT} — Add comments to a program name in the directory. • {RENAME PRGRAM} — Change a program name. • {DELETE PRGRAM} — Delete a single program stored in memory. • {REFORM MEMORY} — Delete all programs currently stored in memory.
3) EDIT PROGRAM	{EDIT PRGRAM} — Edit an existing program or create a new program.
4) COPY PROGRAM	{COPY PRGRAM} — Copy a program to or from memory or storage device.
5) RENAME PROGRAM	{RENAME PRGRAM} — Rename a program name.
6) DELETE PROGRAM	{DELETE PRGRAM} — Delete a single program stored in memory.
7) REFORM MEMORY	{REFORM MEMORY} — Delete all programs currently stored in memory and format RAM.
8) OFFSETS	• {WORK CO-ORD} — Display and alter the preset work coordinate system zero locations and the fixture offset value. • {WHEEL GEOMET} — Display and alter the wheel geometry tables. • {RADIUS TABLE} — Display and alter the dresser/wheel radius and orientations. • {ACTIVE OFFSET} — Change the currently active offset number without requiring the programming of a different offset number. • {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) RADIUS TABLE	
10) WHEEL GEOMETRY	{WHEEL GEOMET} — Display and alter the wheel geometry, radius and orientation values.
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 wheel offset data.
13) QUICK CHECK™	{QUICK CHECK} — Use the syntax and format checker or the {QUICK CHECK} with graphics function.
14) SYSTEM SUPORT	• {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.

Table 2.C
Password Protectable Functions (continued)

Parameter Name	Function becomes accessible when parameter name appears in reverse video:
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	• `{SET TIME}` — Change or reset time parameters • `{SET DATE}` — Change or reset date parameters 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.
22) SI/OEM MESSAGE	• {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.
23) SCALING	When SCALING is not in reverse video, the operator still has access to the {SCALNG} softkey; however values on the screen cannot be modified.
24) CHANGE DIRECTORY	{CHANGE DIR} — Display either the main part program directory or the protectable part program directory.
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} softkey that allows searching and monitoring of the control's PAL program.
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 Storing Password List to Backup Memory

You can save your password list in the backup memory of the control so that it won't be lost if system power and battery-backup fails.

To store the password list, follow these steps:

1. Press the `{PASSWORD}` softkey.

```
(softkey level 1)

        PRGRAM  OFFSET  MACRO   PRGRAM  SYSTEM   ──►
        MANAGE          PARAM   CHECK   SUPORT

        FRONT   ERROR   PASS-   SWITCH           ──►
        PANEL   MESAGE  WORD    LANG
```

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}.

2. Press the {ACCESS CONTRL} softkey. This displays the access control screen (Figure 2.5).

(softkey level 2)

↑				ACCESS CONTRL		

3. Press the {STORE BACKUP} softkey. This softkey appears in reverse video and the message “STORING TO BACKUP – PLEASE WAIT” appears on the CRT until the control has finished storing the password list to its backup memory.

(softkey level 3)

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

	UPDATE & EXIT	05 (NAME)	06 (NAME)	07 (NAME)	08 (NAME)	→

	STORE BACKUP					→

After storing the password list, the control returns to softkey level 2 and displays the message “STORED PASSWORD LIST TO BACKUP” on line 2 of the CRT.

2.7.4 Entering Passwords

When the control is powered-up, the only functions that are available to the operator are functions that are not protectable and functions that are assigned to access level 8 (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)

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

You see the password log-on screen:

```
                                                          E-STOP
    ENTER PASSWORD:

     PROGRAM   [INCH]        F      0.000 MMPM

         Z  00000.000        S      0

       R X  00000.000        T  1

         C     359.99


      MEMORY       MAN  STOP
```

↑				ACCESS CONTRL		

The prompt “ENTER PASSWORD:” appears on line 2 of the CRT.

2. Enter the password. The control displays only * for the characters 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 by using the `{FRONT PANEL}` softkey. This is configurable by your system installer. Both means of selection may not be available. Details about using the `{FRONT PANEL}` softkey are given on page 2-11.

Depending on the current control status, a mode change request may not be honored. Operating modes cannot be changed if any of the following 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 threading or performing a single pass turning cycle

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

(1) Manual mode

To operate the machine manually,

- select MAN or MANUAL under `<MODE SELECT>`

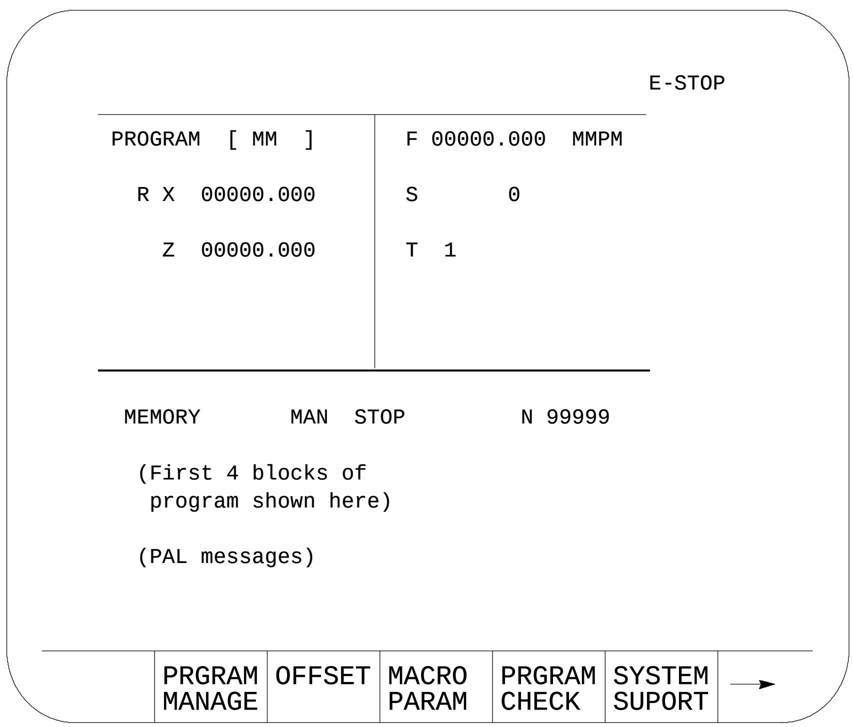
  or

- press the `{FRONT PANEL}` softkey.

Use the left/right arrow keys to change the mode select options if using `{FRONT PANEL}`. Details about using the `{FRONT PANEL}` softkey are given on page 2-11.

For details on manual mode operation, see chapter 4.

Figure 2.6
Manual Mode Screen



(2) MDI mode

To operate the machine in MDI mode,

- select MDI under `<MODE SELECT>`

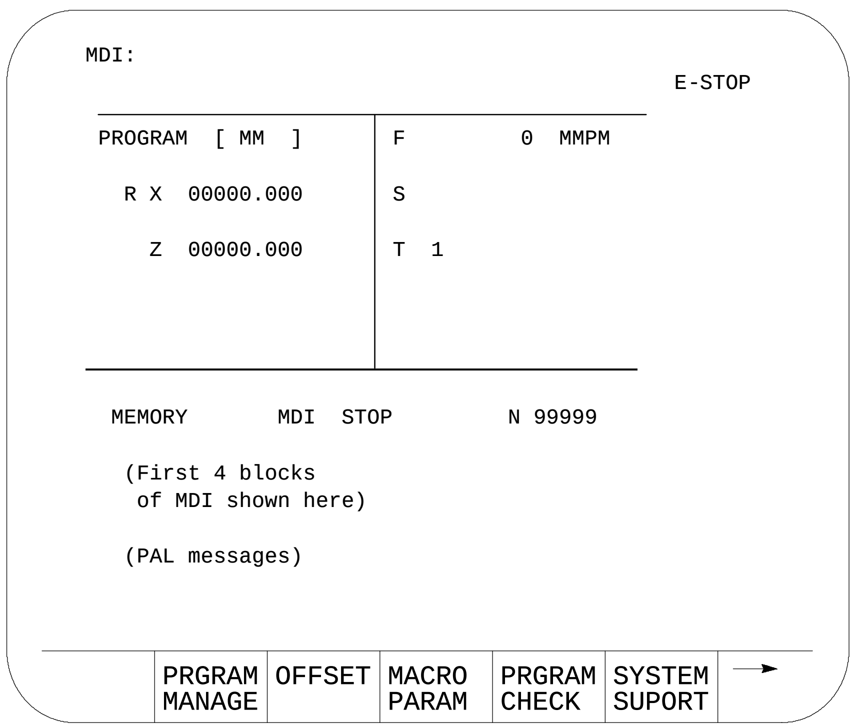
  or

- press the `{FRONT PANEL}` softkey

Use left/right arrow keys to change mode select options if using `{FRONT PANEL}`. Details about using the `{FRONT PANEL}` softkey are given on page 2-11.

For details on MDI operation, see chapter 4.

Figure 2.7
MDI Mode Screen



(3) Automatic mode

To operate the machine automatically,

- select AUTO under `<MODE SELECT>`

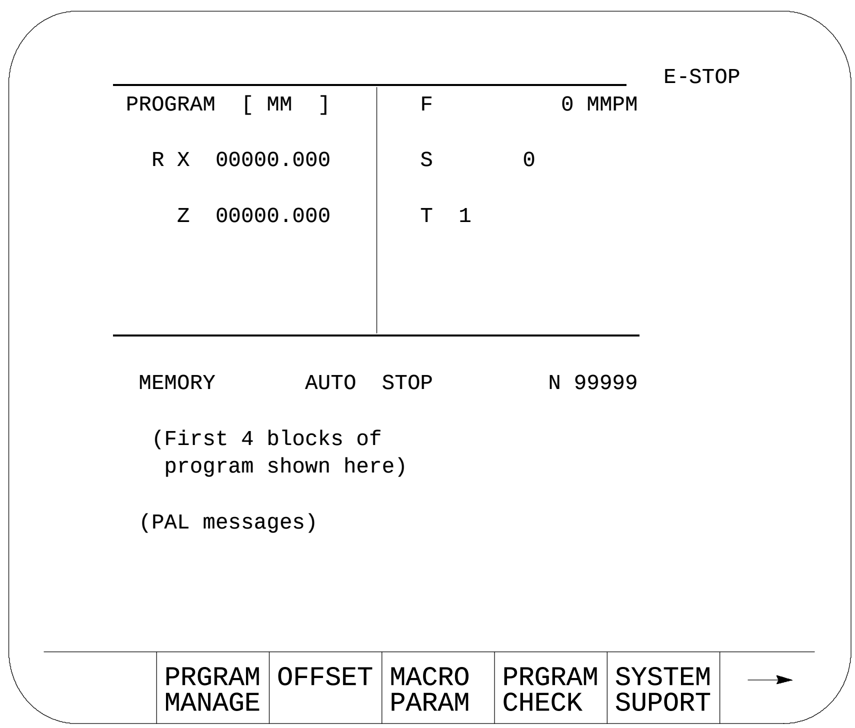
  or

- press the `{FRONT PANEL}` softkey

Use left/right arrow keys to select mode options if using `{FRONT PANEL}`. Details about using the `{FRONT PANEL}` softkey are given on page 2-11.

For details on automatic operation, see chapter 7.

Figure 2.8
Automatic Operation Screen



2.9 Control and Block Reset

Block Reset

Use the block reset feature to force the control to skip the execution of a block. To use the block reset function, you must stop program execution.

If program execution is stopped:	Then:
before the control has completely finished the execution of the block	a block reset aborts any portion of that block that has not been executed.
after the complete execution of a block (as in the case of single-block execution or a M00, etc.)	the control aborts the execution of the entire following block.

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

When executing protected program blocks, the control displays the last non-protected block on the CRT. If you perform a block reset during execution of a protected program block, the protected block is reset but the block displayed on the CRT does not change until the control encounters an M02, M30, or M99.

Control Reset

By executing a control reset, you can:

- return the control to the default parameters (except inch/metric which remains in its last programmed state when control reset is performed).
- clear any programming errors that have recently occurred
- cancel any MDI commands

After the execution of a control reset:

- any active program is reset to the first block
- any programmed offsets or rotations to the coordinate systems are reset to default
- any MDI command is discarded

All of the initial operating parameters return to the standard AMP assigned values including any AMP assigned G codes normally active at power-up.

Press the `[RESET]` key on the operator panel while holding down the `[SHIFT]` key to execute a control reset. Your system installer can also configure AMP to force a control reset when you perform a E-Stop reset.

2.10 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 MESAGE} softkey to enter the MESSAGE ACTIVE screen shown in Figure 2.9.

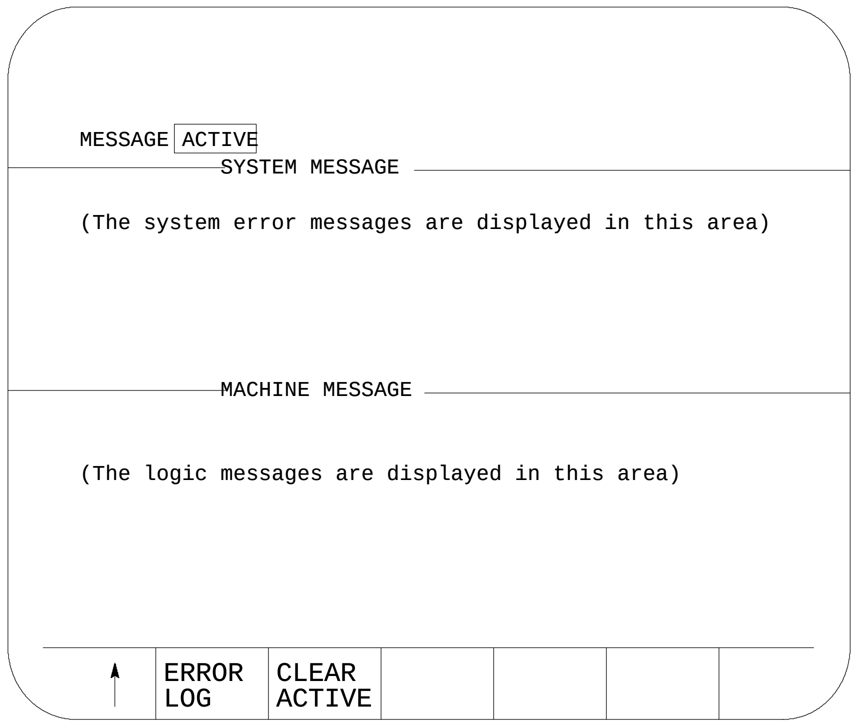
(softkey level 1)

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH 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 2.9
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 2.10.

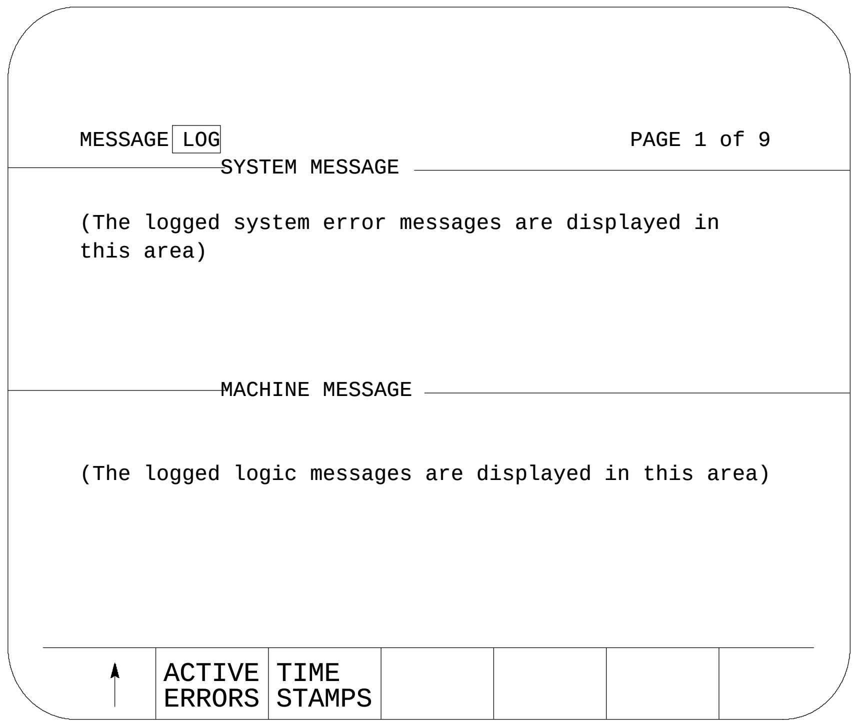
(softkey level 2)

	ERROR LOG	CLEAR ACTIVE				→

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 2.10
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 MESAGE} 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.

2.10.1 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 cleared.

ATTENTION: 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 is of 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 (line 1) of other screens.

Clear active messages from the screen in the following ways:

If you want to:	Then press the:
clear the most recent active messages one at a time	`[CAN]` key
Clear all active messages from the error message display screen	`{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.

2.11 The Input Cursor

The input cursor is the cursor located on line 2 and 3 of the screen. It appears when you must input data using the operator panel (as needed in MDI mode, for example). This section describes how to move the cursor and edit data on the input line by using the keys on the operator panel.

Moving The Input Cursor

To move the cursor left or right in the input area, press and hold the `[SHIFT]` key while pressing the left or right cursor keys.

To move the cursor between line 2 and 3, use the left or 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 line.

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 using the operator panel is inserted 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. Press the `[DEL]` key to delete 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 the input line, press the `[TRANSMIT]` key. All information on the input line is sent to the control.

2.12 {REFORM MEMORY}

Sometimes you must perform a reform memory operation on the control to clear memory. Typically, you do this when:

- 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)
- battery power was lost when the 9/240 control was turned off
- after 5 days' backup by the supercapacitor, the optional lithium battery takes over; this provides 2 years of continuous, accumulated backup for the standard 9/260 or 9/290 control, and 1 year for the 9/260 or 9/290 with extended program storage

Important: You must format any time the control displays the error message "MEMORY CRASH – REFORMAT."

You can also perform this operation if you want to delete all part programs in control memory.

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

ATTENTION: The reform memory function erases all part programs that are stored in control memory. This includes programs in both the main and protected program directories. You can back up these programs to ODS or to another peripheral device before reformatting as described in chapter 6 and 9.

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

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the {REFORM MEMORY} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. Press the {REFORM YES} softkey. All programs that are stored in control memory are deleted.

(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 "disabled" until the format operation is completed.

To abort the operation, press the {REFORM NO} softkey.

(softkey level 3)

↑		REFORM YES	REFORM NO			

2.13 Removing an Axis (Axis Detach)

Use the axis detach feature to remove (detach) a rotary table or other axis attachment from a machine. When activated, the control ignores messages, e.g., servo errors, that occur resulting from the loss of feedback from a detached axis.

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

You enable this feature in AMP. You must select the axis as "detached" to be considered removed. See the documentation supplied by your system installer for the steps involved in detaching an axis or physically removing axis hardware from your machine.

2.14 Time Parts Count Display Feature

The time parts count display logs data relevant to part program execution, including:

- number of parts ground
- cycle time
- lot size
- remaining parts

You display and alter this data through the time parts screen.

Three levels of access are available to the time parts screen. They are listed below in order of most restrictive to least restrictive. See page 2-27 for details on password protection and access control.

Degree of Restriction:	Access Level:	Type of Restriction:
Most restrictive	No access	Restricts operator from Time Parts screen entirely (softkey {TIME PARTS} not accessible). Accomplished by denying access to “Active Program.”
Moderately restrictive	Operator access	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.”
Least restrictive	Supervisor access	Full access to all features of the Time Parts screen.

To access the time parts screen, follow these steps:

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the {ACTIVE PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. Press the {TIME PARTS} softkey.

(softkey level 3)

↑	DE-ACT PRGRAM	SEARCH	MID ST PRGRAM	T PATH GRAPH	SEQ STOP	→

↑					TIME PARTS	→

Figure 2.11
Time Part Screen

```
  PROGRAM          DATE           TIME
  XXXXXXXX       MM/DD/YY       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        :    999999999
LOT COMPLETION FLAG         :            0

  (PAL messages)
```

↑	SET TIME	SET DATE	ED PRT INFO			

Important: All softkeys appear in Figure 2.11 may not appear on your system due to restricted access. See the beginning of this section and page 2-27 for details.

Table 2.D lists the time part screen fields and their definitions.

Table 2.D
Time Part Screen Field Definitions

Field:	Definition:
Program	indicates the currently active part program, displayed automatically by the control.
Date	indicates the current date setting.
Time	indicates the current 24-hour time setting. You can change the time setting provided that you have supervisor-level access.
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.
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, but M02 or M30 does).
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.
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. It is restored to 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 you clear the “Run Time” field to zero. You cannot clear this field 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 the cycle-start button to initiate program execution, turn off power to the control, or follow the clear/reset procedure.
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 you clear the “Run Time” field to zero. You cannot clear this field independently.
Lot Size	indicates the number of times you need to execute this particular part program.
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.

Changing Date

To change the date 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 enter the new date using one of these formats:

   - Month/Day/Year is the English format
   - Year/Month/Day is the Japanese format
   - Day/Month/Year is the Italian and German format

Changing Time

To change the time setting:

1. Press the {SET TIME} softkey.

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

2. Change the current setting by pressing the [DEL] key to erase characters, and the left or right arrow keys to position the cursor. Then type in the new time. The control's clock rolls over every 24 hours.

   You must enter the new time using the format hours:minutes:seconds.

   For example:

   2:16 pm should be entered as 14:16:00.

Clearing/Resetting a Field

To clear/reset a field to zero:

1. Press the {ED PRT INFO} softkey, provided that you have supervisor-level access.

2. Press the up and down cursor keys to move to the field you want to clear/reset.

3. Enter a Y or a numeric value at the prompt for this field.

4. Press [TRANSMIT] to accept the change.

Press the exit softkey {↑} to return to the "Active Program" screen.

2.15 Calculator Function

You can perform simple mathematical calculations when you are operating or programming the control. These mathematical calculations are processed by the calculator function of the control.

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

How to Use the Calculator Function

To use the calculator function:

1. Press the [CALC] key on the operator panel. The CALC: prompt appears on the input line of the screen (line 2).

2. At the CALC: prompt, enter a mathematical equation.

3. Press the [TRANSMIT] key to evaluate the expression. The answer to the expression appears on the input line.

How to Enter Values and Do Mathematical Operations

Mathematical expressions entered on the input line cannot exceed 25 characters. These expressions must contain the numeric or special mathematical operation characters as described below. If you enter any characters that are not numeric or operation characters, the error message "INVALID CHARACTER" appears.

For the calculator function, 2147483647 (10 characters long) is the largest number that you can enter on the input line.

If you see the error message:	The number entered or calculated is:
“NUMBER IS OUT OF RANGE”	too large (longer than 10 characters)
“MATH OVERFLOW”	greater than 2147483647

Fractional numbers cannot exceed .999999 (6 decimal places). If the number of decimal places is exceeded, the control automatically rounds off based on the seventh digit.

If the seventh digit is:	The control:
less than 5	rounds down
5 or greater	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.E to indicate mathematic operations.

Table 2.E
Mathematical Operators

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

This is the order of execution for each of the above operations:

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 you perform the same level of evaluation, 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.F contains the function commands available with the `[CALC]` key.

Table 2.F
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 programming these functions, place the value that the function is to be performed on in brackets, e.g., SIN [10]. The arc tangent function (ATAN) is the exception. The format for ATAN requires the division of two values, e.g., ATAN [10]/[2] is used to calculate the arc tangent of 5.

The control executes functions in Table 2.F 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 2.2
Format for [CALC] Functions

This expression:	Evaluates the:
SIN[2]	sine of 2 degrees
SQRT[14+2]	square root of 16
SIN[SQRT[14+2]]	sine of the square root of 16

Example 2.3
Mathematical Function Examples

Expression Entered	Result	Expression Entered	Result
SIN[90]	1.0	FIX[12.7]	12.0
SQRT[16]	4.0	FUP[12.2]	13.0
ABS[-4]	4.0	FUP[12.0]	12.0
BIN[855]	357.0	LN[9]	2.197225
BCD[357]	855.0	EXP[2]	7.389056
ROUND[12.5]	13.0	ATAN[2]/[5]	21.801413
ROUND[12.49]	12.0		

Important: Take precautions when performing calculations within the brackets [ ] following a mathematical function. The control performs operations within the bracket first, and then the mathematical function is performed on this resultant.

For example:

If your mathematical function is:	Then the result is:
ROUND[2.8+2.6]	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

END OF CHAPTER

Chapter 3

Offset Tables and Setup

3.0 Chapter Overview

This chapter describes the offset tables and their setup. The major topics described in this chapter include:

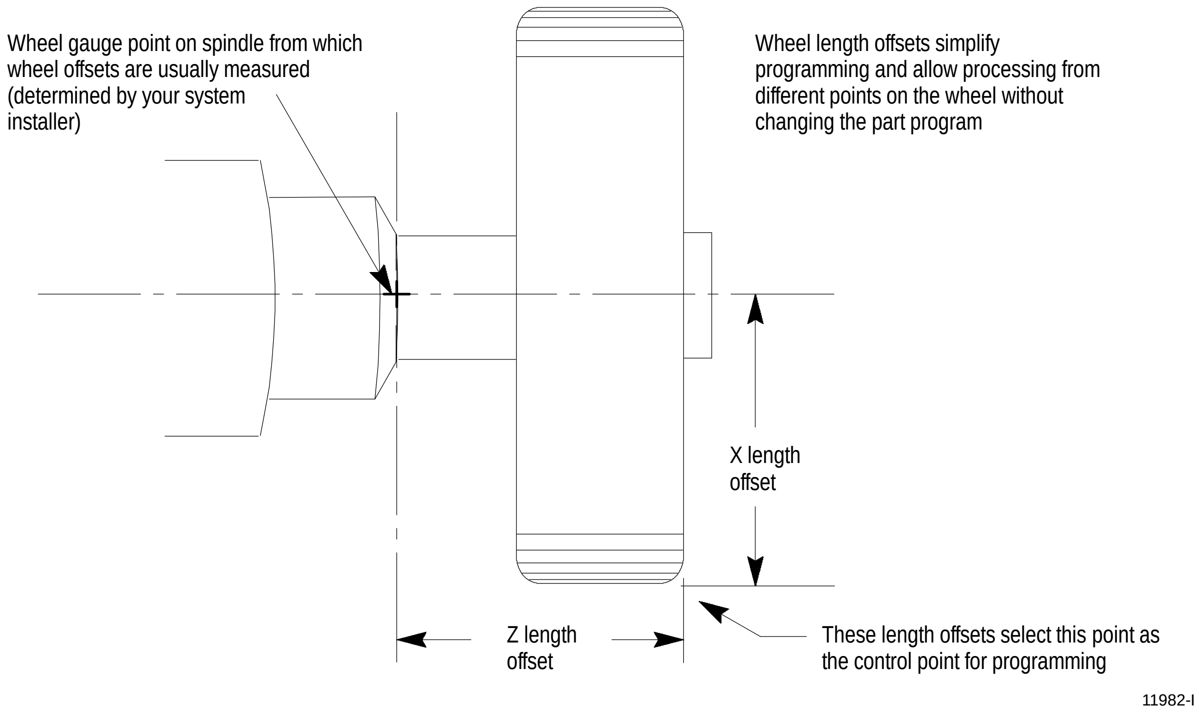
3.1 Wheel Length Offset Tables {WHEEL GEOMET}

Wheel length offsets let the programmer write a part program with respect to the same fixed point on the grinder regardless of the wheel being used. This fixed point (called the gauge point) is determined by your system installer. For the purpose of documentation, this manual assumes that the gauge point is a fixed point on the centerline of the grinding wheel. See your system installer's documentation for details on the location of the gauge point on your system.

This section describes how to enter the length values that are called later when the corresponding length offset is activated. Typically wheel length offsets are activated with a T word as described in chapter 13.

The control stores wheel length offsets in a table called the wheel geometry table. This table stores wheel length offset values under offset numbers. When you select a certain wheel offset number with a T word in the part program, the control references this table and looks up the associated length offset values stored for that offset number. For details on how and when the length offset is activated, see chapter 13.

Figure 3.1
Wheel length Offsets



Resolution of wheel radius and length offset data is determined by the programming resolution selected for the axis in AMP. For example, an X axis that can be programmed with 4 decimal place accuracy also allows 4 decimal place accuracy in the offset tables.

Figure 3.1 illustrates typical cylindrical grinder length offsets. Length offsets for a surface grinder application can be more complex as additional axes can be offset.

Wheel Length Offset Numbers

Call out wheel offset numbers in a program through use of the T word. The last two digits of the T word specify an offset number for wheel length. This number is independent of the dresser/wheel radius and orientation offset number which is the first two digits of T word. The control then accesses the values assigned to that offset number in the wheel length offset table. Length offsets are activated as described in chapter 13.

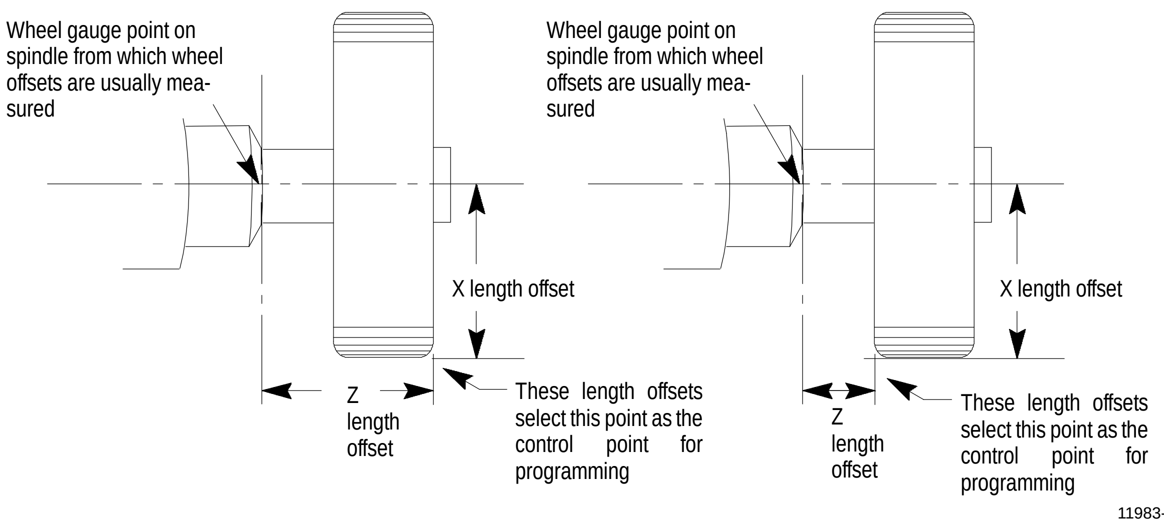
You can store valid wheel lengths for each axis by using offset numbers ranging from 01 to 32. You cannot use offset number “00” to store wheel offset data but you can program “00” to cancel wheel length offsets. You can call different offset numbers from the wheel geometry and wheel radius table using the same T word.

Important: The first 4 wheel offset numbers (01-04) are reserved for use in conjunction with an in-process dresser. When the in-process dresser is disabled, the control automatically updates these first 4 offset numbers with the current grinding wheel size. These offset values should not be manually entered. See chapter 21 for details on using the in-process dressing feature.

Assigning Wheel Length Offset Values

Use the wheel length offset to compensate for the wheel position as mounted on the machine. By using the wheel length offset along with wheel radius and orientation information, a programmer can write a part program without concern for wheel dimensions and shape. Measure wheel length values for each axis to define the control point for programming relative to the wheel gauge point (see Figure 3.2). The same wheel can have several offset numbers to let the programmer select which part of the wheel is being controlled.

Figure 3.2
Wheel Length Offsets



You can configure each axis on your system to have wheel length offset values. This manual assumes a two axis Z and X configuration.

Figure 3.2 shows typical length offsets for a cylindrical grinder. Generally grinders are configured such that axes move in the negative direction as they move the wheel towards the workpiece (along -X axis) and towards the chuck (along -Z axis). Assuming this applies to your system, enter a positive Z offset value to offset the gauge point away from the part spindle. Enter a positive X offset value to offset the gauge point away from the part spindle center line. Different wheel control points are established by the values entered in the length offset table.

Cylindrical grinders can have axes assigned in AMP as diameter axes. This allows values to be entered into the length offset table for that axis as either diameter or radius values. This manual assumes that you use radius values. If you enter values as diameters, you must enter the wheel diameter instead of the wheel radius for the axis configured as the diameter axis. G07 and G08 (radius and diameter modes) have no affect on values entered into the offset tables.

Enter your length offset data in the wheel geometry table as described on page 3-11.

If you are entering offset values for an angled-wheel grinder you should also read the angled-wheel tool offset section on page 14-16.

3.2 Dresser/Wheel Radius Offsets {RADIUS TABLE}

Radius offset values let the programmer write a part program without having to calculate wheel positions relative to the radius of the grinding wheel, grinding wheel corner, or edge of the diamond point dresser. This section describes entering these radius values that are used for dresser/wheel radius compensation as described in chapter 15.

The control stores wheel radius offsets in a table called the `radius table`. This table stores the radius values under radius offset numbers. Call out a radius offset number with a T word in the part program. The control references this table and looks up the associated radius values. For details on how and when the radius offset is activated, see chapter 15.

The first two digits of the T word specify an offset number for radius and orientation. This number is independent of the wheel length offset number which is the last two digits of T word. Length offsets are activated as described in chapter 13.

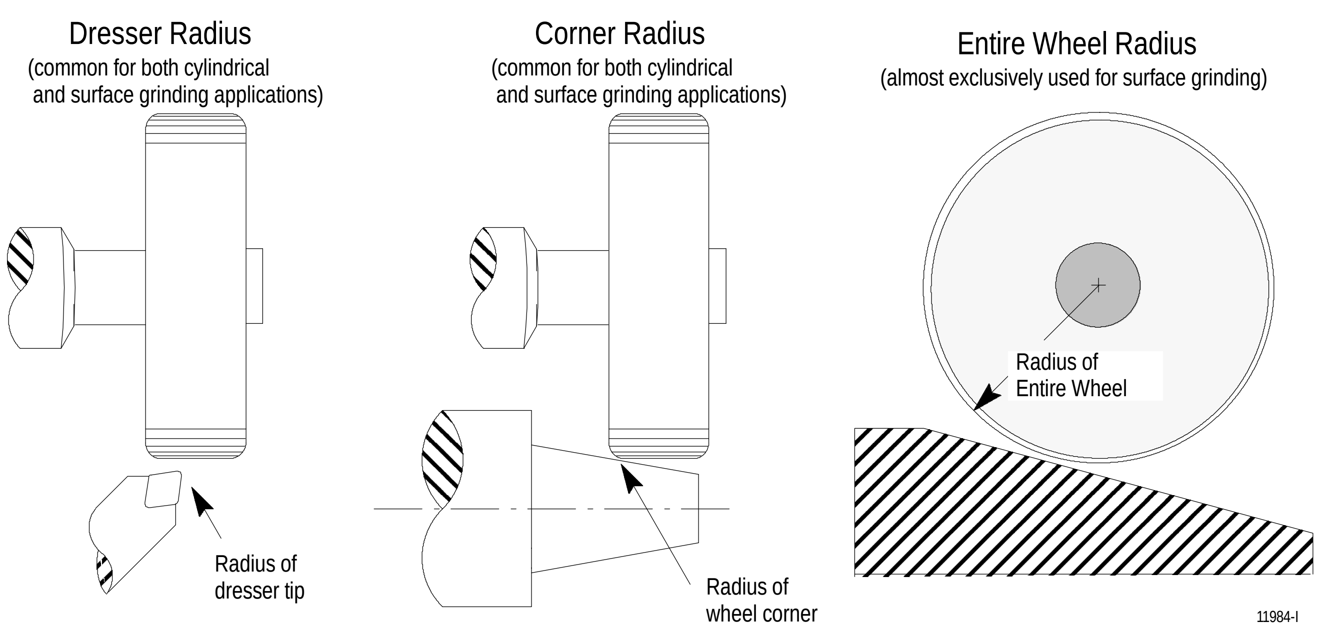
The control supports 3 dresser/wheel radius compensation schemes:

- Dresser radius
- Corner radius
- Entire wheel radius

The dresser radius and corner radius compensation schemes use the same radius table to store a radius value. The entire wheel radius scheme stores the entire wheel radius in paramacro variable #5508. Which dresser/wheel radius compensation scheme to use on your system depends on the current application of your grinder. See chapter 15 for details on how to properly implement these schemes.

Dresser/Wheel Radius Compensation Scheme	Length Offsets	Coordinate System Offset (G54-G59.3)	Dresser/Wheel Radius Compensation
Dresser Radius	Shifted on Z and X axis to wheel control point	Shifted to point on dresser tip	On radius of diamond dresser
Corner Radius	Shifted on Z and X axis to wheel control point	Shifted to point on part being machined	On radius of wheel corner where Z and X length offset is located
Entire Wheel Radius	Shifted on Z to control point of wheel, Y offset is taken into consideration with dresser/wheel radius compensation	Shifted to point on part being machined	On entire radius of wheel

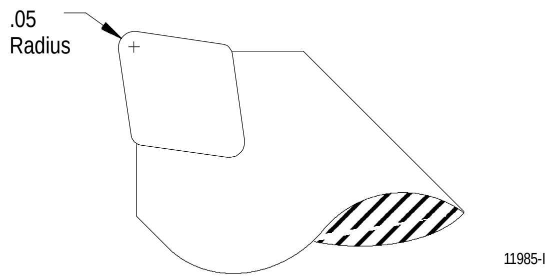
Figure 3.3
Dresser/Wheel Radius Compensation Schemes



Dresser Radius

The control can compensate for errors resulting from slight or even large rounding of the dresser tip. To do so, the radius of the dresser must be entered in the radius table. For more information on activating an offset for dresser/wheel radius compensation, see chapter 15.

Figure 3.4
Dresser Radius for a Typical Diamond Point Dresser

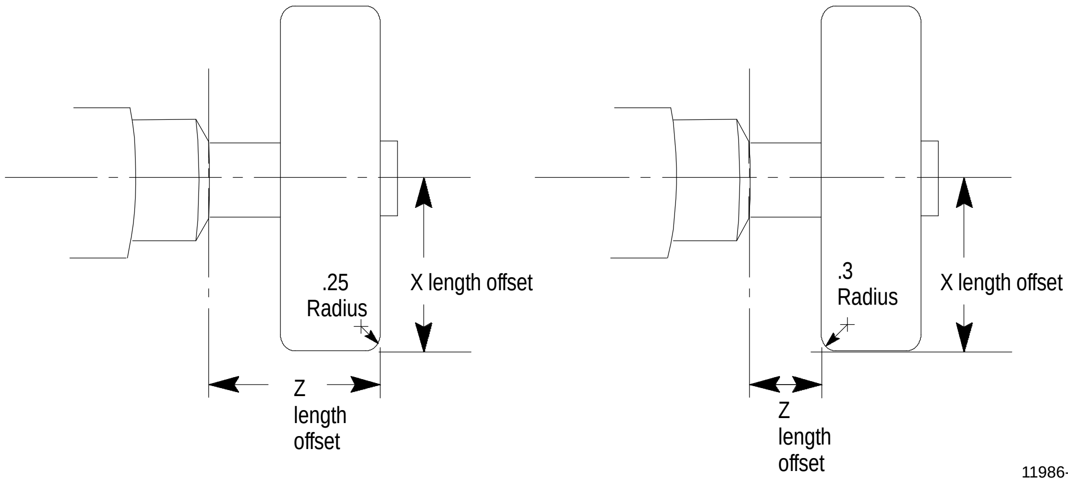


Corner Radius

The control can compensate for error resulting from the rounded corner commonly dressed into the grinding wheel. This corner radius should be the corner of the wheel selected as the control point using wheel length offsets.

In order for dresser/wheel radius compensation to properly compensate for wheel corner roundness, you must enter the corner radius of the control point wheel corner into the offset table. The same wheel can have 1, 2, or more corners entered for its corner radius. It is the programmer's responsibility to use the correct radius (called by first two digits in T word) for the correct control point of the grinding wheel (called by second two digits in T word).

Figure 3.5
Corner Radius for a Typical Grinding Wheel



Entire Wheel Radius

The control can compensate for the radius of the entire grinding wheel. To do so, the radius of the wheel must be entered in the radius table for dresser/wheel radius compensation. This method of compensation does not require any X-length offset to be activated. Only a Z-length offset needs to be activated with the X offset being compensated for by the wheel radius. For more information on activating an offset for dresser/wheel radius compensation, see chapter 15.

Important: An additional radius offset number (33) is available only to the in-process dresser feature. This radius value is the entire wheel radius and changes as the wheel radius diminishes each time it is dressed. See chapter 21 for details on the in-process dresser feature.

To assign a value to radius offset number 33, use either:

- the dresser data screen “Current Wheel Diameter” found on page 21-9.
- directly assign a value to paramacro parameter #5508 (see page 20-39)

Figure 3.6
Entire Wheel Radius for a Typical Grinding Wheel

3.3 Dresser/Wheel Orientation {RADIUS TABLE}

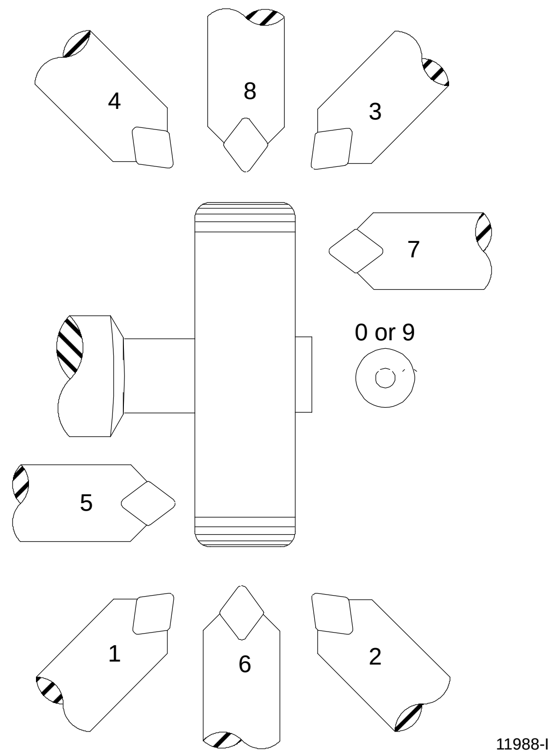
Orientation of the grinding wheel or diamond point dresser is essential information for dresser/wheel radius compensation to function properly. Orientation data tells the control which side of the dresser/wheel is available for grinding relative to the wheel/part surface.

The control uses orientation data during dresser/wheel radius compensation as described in chapter 15. This section describes entering this orientation data. Since orientation data is essential to radius compensation, the orientation data in entered in the radius table. This table also contains information on the dresser/wheel radius as described on page 3-4. Each radius value entered in the radius table requires corresponding orientation data.

3.3.1 Dresser Orientations

Figure 3.7 shows the possible dresser orientations relative to the grinding wheel.

Figure 3.7
Dresser Orientations

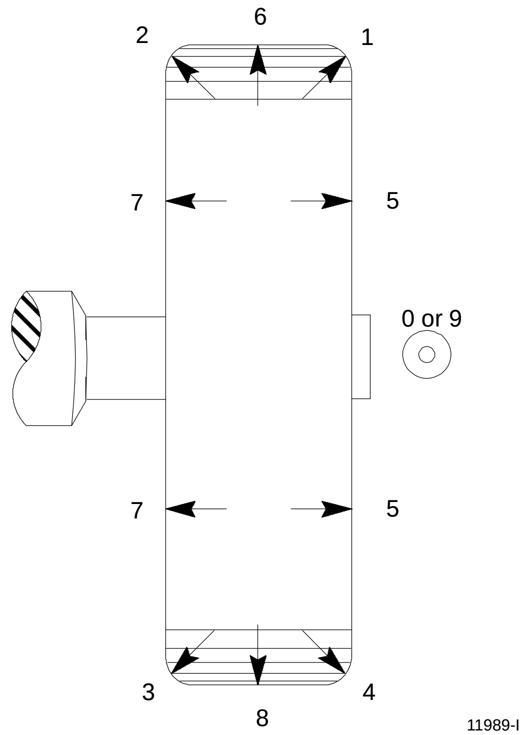


The control uses the value selected for orientation to determine the orientation of the dresser when dresser/wheel radius compensation is active. Enter the proper dresser orientation number (0-9) in the radius offset table for the ORNT parameters. From that information, the control can keep track of the orientation of the dresser you are currently using and prevent some programming errors. Details on entering this information in the actual offset table are covered on page 3-11.

3.3.2 Grinding Wheel Orientations

Figure 3.8 shows the possible grinding wheel orientations relative to the part surface. The orientation numbers point to the surface of the grinding wheel being used to grind the part.

Figure 3.8
Wheel Orientations



The control uses the value selected for orientation to determine the orientation of the grinding wheel when dresser/wheel radius compensation is active. Enter the proper wheel orientation number (0-9) into the radius offset table for the ORNT parameters. From that information, the control can keep track of the orientation of the wheel, and prevent some programming errors. Details on entering this information into the actual offset table are described on page 3-11.

3.4 Entering Offset Data {WHEEL GEOMET } or {RADIUS TABLE}

Enter data in the offset tables by using one of six methods:

Method:	On Page:
Editing wheel offset tables {WHEEL GEOMET} or {RADIUS TABLE}	3-11
Using {MEASURE}	3-16
Programming G10	13-5
Skip functions	19-3
Setting paramacro system parameters	20-16
Editing through the PAL program	see the PAL Reference manual or documentation provided by your system installer

When you use PAL to modify the dresser/wheel radius offset table, dresser/wheel radius compensation (G41 or G42) should not be active. If dresser/wheel radius compensation is active, be aware that the new offset is not placed in part program set-up buffers that have already been read into control memory. This results in the offset not being activated until several program blocks after the current block. The number of setup buffers depends on the number of block retrace steps configured in AMP and what software features are currently being used.

The method described here requires that the offset data be manually measured and then directly keyed into the table.

To manually display or alter the offset tables, 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	SWITCH LANG		→

2. Decide if you want to display:

- wheel length offset table

  or

- radius/orientation offset table

(softkey level 2)

↑	WORK CO-ORD	WHEEL GEOMET	RADIUS TABLE	DRESSR TABLE	SCALNG	→
↑	COORD ROTATE	BACKUP OFFSET				→

To display:	Press this softkey:
wheel length offsets	{WHEEL GEOMET} An example of a wheel length offset table screen is shown in Figure 3.9.
radius/orientation offsets	{RADIUS TABLE} An example of a radius/orientation offset table screen is shown in Figure 3.10.

3. Move the cursor to the offset data that you want to modify. The offset data located at the cursor appears in reverse video.

Important: The softkeys below appear after pressing {WHEEL GEOMET} on a cylindrical grinder. If you press {RADIUS TABLE} or if your control is configured as a surface grinder, several of these softkeys are not available.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	ACTIVE OFFSET	MORE OFFSET	→
↑	MEAS-URE	INCH/ METRIC	RADI/ DIAM		COPY OFFSET	→

If you want to:	Press:
move the block cursor to the offset data on the current page	up,down, left, or right cursor keys
change pages	{MORE OFFSET} softkey
search all pages for a specific offset number by keying in the desired offset number and by pressing the [TRANSMIT] key	{SEARCH NUMBER} softkey

4. Select data entry type:

Unit selection {INCH/METRIC}

To select units of “mm” or “inch” for the offset data, press the {INCH/METRIC} softkey. The unit selection changes each time you press the softkey. When you alter the units, the control converts all existing data to the new unit selection for that offset number.

Diameter or Radius {RADI/DIAM}

If the offset value being changed is for an axis selected in AMP as a diameter axis, you can enter data into the offset table as either a radius or diameter value (typically the axis perpendicular to the headstock center line). Next to the axes offset, the control displays the current mode for that axis.

If you see:	Then the current mode is:
R	radius
D	diameter

Press the {RADI/ DIAM} softkey to toggle the offset between these two modes.

This feature applies only to the {WHEEL GEOMET} screen for wheel lengths on <u>cylindrical grinders</u>. If your control is configured as a surface grinder this softkey is not available (also the R and D do not appear on the wheel geometry screen).

This feature is not available for surface grinder applications.

Important: This softkey does not change the current mode of control operation (as selected with G07 or G08), it only alters how data is entered in the table and displayed. For details on radius or diameter mode (G07/G08), see page 11-46.

(softkey level 3)

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

↑	MEAS- URE	INCH/ METRIC	RADI/ DIAM		COPY OFFSET	→

5. Offset data can be replaced or added to:

If you want to:	Key in the:	Press this softkey:
replace stored offset data with new data	new data	{REPLCE VALUE}
add to previously stored offset data	amount to be added	{ADD TO VALUE}

You can copy length offset data from one axis to another axis for all offset numbers (rather than having to change each axis individually).

a. Press the {COPY OFFSET} softkey.
"COPY (SOURCE, DESTINATION) :" appears.

b. Enter the axis letter <u>from</u> which the data is coming, then <u>a comma</u>, and then enter the axis letter <u>to</u> which the data is going. For example:

COPY (SOURCE, DESTINATION): X1,X2

copies the offset data from the X1-axis to the X2-axis for all offset numbers.

```
(softkey level 3)
```

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

↑	MEAS-URE	INCH/ METRIC	RADI/ DIAM		COPY OFFSET	→

Figure 3.9
Offset Table Screen for Wheel Length

```
TOOL OFFSET NUMBER:

 WHEEL GEOMETRY TABLE                    PAGE  1 OF 7

 TOOL #   1  [INCH]     2  [INCH]        3   [INCH]
     R X    -12345.678     -12345.678        -12345.678
       Z    -12345.678     -12345.678        -12345.678



 TOOL #   4  [INCH]     5  [INCH]        6   [INCH]
     R X    -12345.678     -12345.678        -12345.678
       Z    -12345.678     -12345.678        -12345.678
```

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

Figure 3.10
Radius Offset Screen

```
TOOL OFFSET NUMBER:

 RADIUS TABLE                            PAGE  1 OF 3

 TOOL #          RADIUS         ORIENT         UNITS
   1             0.25             3            [INCH]
   2             0.33             2            [INCH]
   3             0.37             3            [INCH]
   4             0.55             4            [INCH]
   5             0.59             3            [INCH]
   6             0.61             6            [INCH]
   7             0.77             3            [INCH]
   8             0.82             8            [INCH]
   9             0.85             9            [INCH]
  10             0.90             3            [INCH]
  11             0.93             3            [INCH]
  12             0.96             3            [INCH]
  13             0.99             4            [INCH]
```

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

3.5 Set Offset Data Using {MEASURE}

The measure feature offers an easy method of establishing wheel length offsets. This feature is not available for generating any radius offset data. The control, not the operator, computes the wheel length offsets, and enters these values in the wheel geometry offset table. You can only perform the measure operation on one axis at a time.

To enter wheel length offsets using measure, follow these steps:

1. Establish a fixed machine position. This position can be any fixed, non-movable location on the machine that the wheel can be jogged against consistently. Determine the value of this location at the gauge point with no wheel offsets active.

   Important: No wheel offsets should be active. The value of this position at the gauge point, located in the work coordinate system, must be recorded.

   You key in this value in steps 6 and 7.

2. Access the wheel geometry offset table {WHEEL GEOMET}.

3. Cursor down to the offset that you want to change. You can select to display the offset in either inch or metric measurements.

4. Using incremental, continuous or handwheel mode, jog the wheel surface to be measured against the fixed location determined in step 1 on the axis being measured.

5. Press the {MEASURE} softkey.

6. Key in the coordinate value of the fixed location determined in step 1. This coordinate value must be in the same units as the control is currently operating in. For example, you cannot key in an inch value when the control is in metric mode (G71).

   If the same length offset must be transferred to other axes, they must be keyed in manually. If the offset must be copied from offset number to offset number, you can use the {COPY OFFSET} feature.

7. Press the [TRANSMIT] key.

   The control now subtracts the keyed in location from the current wheel position and enters this difference as the length offset value in the table.

3.6 Changing the Active Dresser/Wheel Offset {ACTIVE OFFSET}

This feature allows the manual activation of wheel length and radius/orientation offsets without the need to program the correct T word to call the corresponding offset number.

Typically you change the wheel length and radius/orientation offsets by programming a T word as described in chapter 13. Use this feature only when you need to activate one of these offset numbers manually.

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

1. Press the {OFFSET} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the {WHEEL GEOMET} softkey to activate a wheel length offset number, or press the {RADIUS TABLE} softkey to activate a radius/orientation offset number.

(softkey level 2)

↑	WORK CO-ORD	WHEEL GEOMET	RADIUS TABLE	DRESSR TABLE	SCALNG	→

↑	COORD ROTATE	BACKUP OFFSET				→

The control displays the offset table. Currently active offset values (if any) are indicated with a * to the right of the offset number if you have selected the radius table. If you selected the wheel geometry table, the * is to the left of the offset number.

3. Move the cursor (using cursor keys) on the offset table until the desired offset appears in reverse video. Only one length and one radius/orientation offset number can be active at any time.

4. Press the {ACTIVE OFFSET} softkey when the offset you want is selected. Length offsets are made active as described in chapter 13. Radius/orientation offsets are made active as described in chapter 15.

```
(softkey level 3)

 ↑  | SEARCH | REPLCE | ADD TO | ACTIVE | MORE   | →
    | NUMBER | VALUE  | VALUE  | OFFSET | OFFSET |
```

3.7 Work Coordinate System Offset Table {WORK CO-ORD}

You enter two types of data in the work coordinate system table:

Zero Point Parameters	the initial work coordinate system zero point locations that are called when programming G54 - G59.3.
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

Enter positions for the zero points as machine coordinate values. The control then uses the specified machine coordinate position as the work coordinate system zero point. See page 11-10 for details on work coordinate systems.

Enter a machine coordinate system position for each axis of the work coordinate system below the corresponding G code (G54-G59.3) as described on page 3-19.

External offset

Use the external offset to modify all of the work coordinate system zero points. The value entered here offsets 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 lets a programmer use the same set of work coordinate system values in a variety of applications. Adjusting this value, for example, allows for use of the same work coordinate systems and programs after a different part or dresser/wheel mounting fixture has been installed on the machine. You can also use this to offset all work coordinate systems when part programs are transferred from different machines with different mechanical features. See page 11-10 for details on the external offset.

3.7.1 Entering Work Coordinate System Data

Enter data in the coordinate system table by using one of four methods:

Method:	On page:
Entering work coordinate data manually	3-19
Programming G10	11-8 and 11-11
Setting paramacro system variables	20-16
Entering through the PAL program	see the PAL Reference manual or documentation provided by your system installer

When you use PAL to modify the work coordinate system tables, dresser/wheel radius compensation (G41 or G42) should not be active. If dresser/wheel radius compensation is active, be aware that the new offset is not placed in part program set-up buffers that have already been read into control memory. This results in the offset not being activated until several program blocks after the current block. The number of setup buffers depends on the number of block retrace steps configured in AMP and what software features are currently being used.

The method described here requires that you key in the data directly to the table.

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	PRGRAM 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 3.11.

```
(softkey level 2)
```

↑	WORK CO-ORD	WHEEL GEOMET	RADIUS TABLE	DRESSR TABLE	SCALNG	→
↑	COORD ROTATE	BACKUP OFFSET				→

Figure 3.11
Work Coordinate System Data Entry

```
WORK COORDINATES TABLE     PAGE  1  OF  4


DIAMOND 1          DIAMOND 2          CHUCK 1

 G54  [INCH]       G55  [ MM ]       G56  [ MM ]
R X  -999.9999      X  -999.9999      X  -999.9999
  Z  -999.9999      Z  -999.9999      Z  -999.9999
```

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

3. Move the cursor to the offset data you want to modify. The data located at the cursor appears in reverse video.

If you want to:	Press:
move the block cursor to the offset data on the current page	up, down, left, or right cursor key
change pages	{MORE OFFSET} softkey

4. Select data entry type:

Unit selection {INCH/METRIC}

To select units of “mm” or “inch” for the offset data, press the {INCH/METRIC} softkey. The unit selection changes each time you press the softkey. When you alter the units, the control converts all existing data to the new unit selection for that offset number.

Diameter or Radius {RADI/DIAM}

If the offset value being changed is for an axis selected in AMP as a diameter axis, you can enter data into the offset table as either a radius or diameter value (typically the axis perpendicular to the spindle center line). Next to the axes offset, the control displays the current mode for that axis.

If you see:	Then the current mode is:
R	radius
D	diameter

Press the {RADI/ DIAM} softkey to toggle the offset between these two modes.

Important: This softkey does not change the current mode of control operation (as selected with G07 or G08), it alters only how data is entered in the table and displayed. For details on radius or diameter mode (G07/G08), see page 11-46.

```
(softkey level 3)
```

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

5. Data can be replaced or added to:

```
(softkey level 3)
```

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

If you want to:	Key in the:	Press this softkey:
replace stored work coordinate data with new data	new data	{REPLCE VALUE}
add to previously stored work coordinate data	amount to be added	{ADD TO VALUE}

Important: You can alter the values for the work coordinate systems by using the G10 command in MDI or within a part program. For details on G10 commands, see page 11-8 and 11-11.

Entering a Coordinate System Label

The work coordinate system table provides an area to enter a label for each of the offsets (G54 -G59.3 and the external offset). This label is only for display purposes; it lets you assign different text strings to different offsets for easy identification.

These labels are not backed up when a coordinate system offset table backup is made. They are not part of the G10 format. They do, however, remain when you execute a G10 program.

To change or enter a label for a coordinate system offset:

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

2. Press the {WORK CO-ORD} softkey to display the offset values for the work coordinate systems and the external offset. See Figure 3.11.

3. By using the left and right cursor keys and the {MORE OFFSET} softkey, select the coordinate offset that you want to label.

4. Press the {MODIFY LABEL} softkey to change or enter a new label. The control displays the prompt "ENTER LABEL."

(softkey level 3)

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

↑	MODIFY LABEL					→

5. Type in the new label name by using the keyboard.

6. Press the [TRANSMIT] key. The control overwrites any existing label with the new label name.

3.8 Backing Up Offset Tables

The control can back up all the information entered in the offset tables and the work coordinate system tables. The control can generate a program consisting of G10 blocks to save these tables. These G10 blocks can contain offset and work coordinate values. Any time this program is run, the set of values contained in these G10 blocks replace the current values in the offset tables.

You can save the G10 program in either control memory or to a peripheral device. We recommend using both to insure this data is not lost.

This feature is very useful if you use the same wheel or coordinate system offsets on different machines. The same offset tables can be easily set up by running this G10 program on other machines. Also, if written to a peripheral device, you can use this program as a safety backup in case of memory failure.

Important: Once the control begins executing a G10 program that has been previously generated, it clears 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. Wheel geometry, radius offset, and work coordinate tables are separate. Loading data in to one does not clear the others.

To back up the offset tables, 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	SWITCH LANG		→

2. Press the {BACKUP OFFSET} softkey.

(softkey level 2)

↑	WORK CO-ORD	WHEEL GEOMET	RADIUS TABLE	DRESSR TABLE	SCALNG	→

↑	COORD ROTATE	BACKUP OFFSET				→

The backup offset screen appears:

```
                BACKUP TOOL OFFSETS

                RADIUS TABLE
                WHEEL GEOMETRY TABLE
                WORK COORDINATE OFFSETS
                ALL

      SELECT OPTION USING THE UP/DOWN ARROW

   ↑    | TO     | TO     | TO     |        |        |
        | PORT A | PORT B | FILE   |        |        |
```

3. Select the offsets you want to back up by using the up and down cursor keys. The selected offset appears in reverse video. Select one of four options:

If you want your G10 program to contain:	Select:
all data from the radius offset table	RADIUS TABLE
all data from the wheel geometry tables	WHEEL GEOMETRY
the work coordinate offset information for G54 - G59.3 and the external offset value	WORK COORDINATE
all data from the wheel geometry, radius offset, and work coordinate offset tables	ALL

4. Once you have indicated the data you want to save, determine the destination for the G10 program from these 3 options:

If you want to send the G10 program to:	Press this softkey:
a peripheral attached to port A	{TO PORT A}
a peripheral attached to port B	{TO PORT B}
control memory	{TO FILE}

5. When you press the {TO FILE} softkey, the control prompts you for a program name. Enter a program name by using the alphanumeric keys on the operator panel and press the [TRANSMIT] key (see page 10-8 on program names).

   If you press {TO PORT A} or {TO PORT B} instead of {TO FILE} the control sends the backup program to the device connected to the peripheral port.

3.9 Programmable Zone Table

The programmable zone table feature provides a means to prevent wheel motion from entering or exiting a designated area. For details on programmable zones, see page 11-34.

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 can also be entered in AMP by your system installer. You can also modify programmable zone 3 table values by programming a G22 command. See page 11-40 for details.

To display or alter the values in the programmable zone table, follow these steps:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		

2. Press the {PRGRAM PARAM} softkey.

(softkey level 2)

	PRGRAM PARAM	AMP	DEVICE SETUP	MONI-TOR	TIME PARTS	

	PTOM SI/OEM		SYSTEM TIMING			

3. Press the {ZONE LIMITS} softkey to display the programmable zone table.

```
(softkey level 3)
```

↑	ZONE LIMITS	F1-F9				

The programmable zone table appears:

```
ENTER VALUE:

PROGRAMMABLE ZONE

          LOWER LIMIT        UPPER LIMIT

LIMIT 2

R X  AXIS     -10.0000          0.0000  [INCH]
  Z  AXIS       0.0000          0.0000  [INCH]


LIMIT 3

R X  AXIS       5.0000          9.0000  [INCH]
  Z  AXIS       3.0000          4.0000  [INCH]
```

↑	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	

Important: The control displays the programmable zone coordinates in inch or metric units for a liner axis, depending on which is the currently active program mode, or in degree units for a rotary axis.

4. Use the up and down cursor keys to move the block cursor to the data you want to change. The data located at the cursor appears in reverse video.

5. Data can be replaced or added to:

(softkey level 4)

	REPLCE VALUE	ADD TO VALUE	MORE LIMITS	UPDATE & EXIT	QUIT

If you want to:	Key in the:	Press this softkey:
replace stored zone data with new data	new data	{REPLCE VALUE}
add to previously stored zone data	amount to be added	{ADD TO VALUE}

6. When you have finished editing the inhibit zone data:

(softkey level 4)

	REPLCE VALUE	ADD TO VALUE	MORE LIMITS	UPDATE & EXIT	QUIT

If you want to:	Press this softkey:
save all changes you just made	{UPDATE & EXIT}
delete all changes you just made	{QUIT}

Unlike the other tables described in this chapter, the zone table cannot be backed up with a G10 program.

3.10 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 F word then becomes the active feedrate. For details on single-digit F words, see page 12-61.

To display or alter the values in the single-digit feedrate table, follow these steps:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the {PRGRAM PARAM} softkey.

```
(softkey level 2)
```

↑	PRGRAM PARAM	AMP	DEVICE SETUP	MONI- TOR	TIME PARTS	→

↑	PTOM SI/OEM		SYSTEM TIMING			→

3. Press the {F1 - F9} softkey to display the single-digit feedrate table.

```
(softkey level 3)
```

↑	ZONE LIMITS	F1-F9			

The single-digit feedrate table appears:

```
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: The control displays feedrates in inch or metric depending on which is the active mode. The control displays IPM (inches per minute) or MMPM (millimeters per minute) at the top of the screen, indicating the currently active mode.

4. Use the up and down cursor keys to move the cursor to the feedrate you want to change. The selected feedrate appears in reverse video.

5. Change feedrate values by using one of two choices:

(softkey level 4)

↑	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	

If you want to:	Key in the:	Press this softkey:
replace stored feedrate feedrate data	new data	{REPLCE VALUE}
add to previously stored feedrate data	amount to be added	{ADD TO VALUE}

6. Leave the feedrate parameter screen by using one of two choices:

(softkey level 4)

↑	REPLCE VALUE	ADD TO VALUE		UPDATE & EXIT	QUIT	

If you want to:	Press this softkey:
save the changes you just made	{UPDATE & EXIT}
delete all changes you just made	{QUIT}

3.11 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:

1. Press the {SYSTEM SUPORT} softkey.

(softkey level 1)

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

2. Press the {AMP} softkey.

(softkey level 2)

↑	PRGRAM PARAM	AMP	DEVICE SETUP	MONI-TOR	TIME PARTS	→
↑	PTOM SI/OEM		SYSTEM TIMING			→

3. Press the {AXIS PARAM} softkey.

(softkey level 3)

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

4. Press the {RANGE PARAM} softkey.

(softkey level 4)

↑	SPNDL PARAM	SERVO PARAM	AXIS CALIB	HOME CALIB	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
	X AXIS	[inch]
Per axis values	MAXIMUM WEAR OFFSET CHANGE	.01000
	MAXIMUM GEOM OFFSET CHANGE	1.00000
	MAXIMUM +/- WEAR OFFSET	.10000
	MAXIMUM +/- GEOM OFFSET	10.00000

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

About the Offset Range Verification Screen

- on a grinder, range checking units for this screen are <u>always</u> RADIUS, regardless of the program/control mode
- 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 <u>entry</u> 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.”

END OF CHAPTER

Chapter 4

Manual/MDI Operation Modes

4.0 Chapter Overview

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

This manual assumes that you use the push button or rotary MTB and you have installed standard PAL to run that MTB panel. For applications that do not use an MTB panel or standard PAL to run the MTB panel, see documentation prepared by your system installer for details about operations described in this chapter.

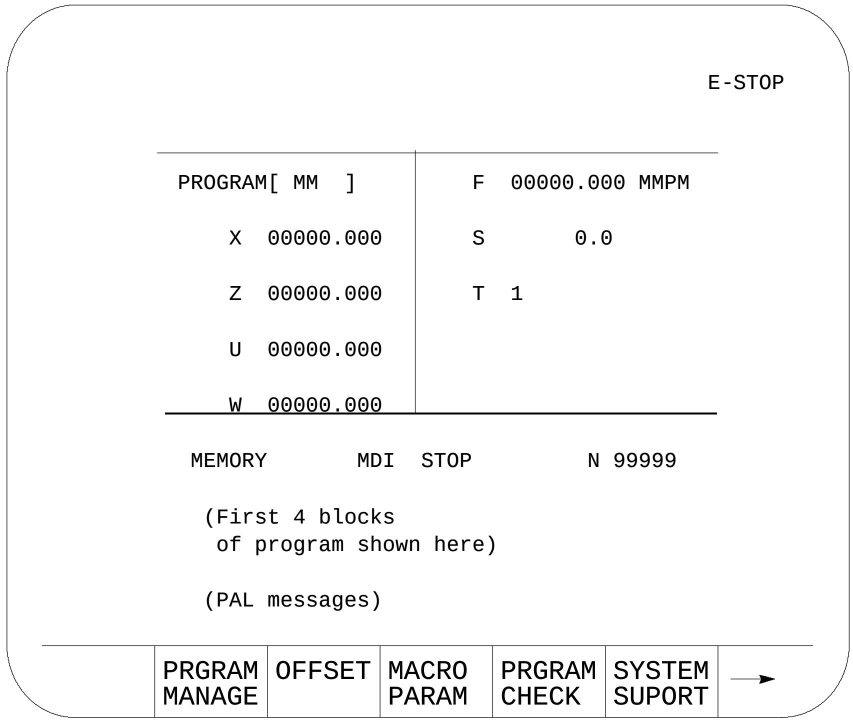
4.1 Manual Operating Mode

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 (see page 2-32 on changing operating mode).

In manual mode, the control displays both the axis position data and any currently active part program in the data display area if the normal display is used for softkey level 1.

Normally you can press `<CYCLE STOP>` to abort manual operations. Your system installer has the option, however, to assign some other switch to abort manual operations in the PAL program. See 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 mode, pushbuttons, switches, or hand pulse generators (HPGs) control the motion of the grinding wheel. Typically these are mounted on or near the MTB panel.

You can jog the grinding wheel by using 3 different methods:

If you choose:	While a pushbutton on the MTB is pressed, the axes move:
continuous jog	continuously
incremental jog	a predetermined amount
HPG jog	when an HPG is turned by hand and an axis has been selected

Normally you can only jog the axes in manual mode. However, your system installer can write PAL to allow jogging in the automatic and MDI mode. See page 4-6 for manual gap elimination or chapter 11 for details on jogging an offset.

You can equip your control with an optional jogging an offset feature, activated by a switch installed by your 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. For information about jogging an offset, see chapter 11.

During a jog retract operation (see chapter 7), you are permitted to use only normal single axis jogs (one axis at a time in the continuous, incremental, or HPG modes).

4.1.2 Continuous Jog

To jog an axis continuously:

1. Select CONTINUOUS under `<JOG SELECT>`.

2. Select the feedrate for continuous jog under `<SPEED/MULTIPLY>`.

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

For details on using the feedrate override switches, see chapter 11.

3. Press the `<AXIS/DIRECTION>` button for the axis and direction to jog. The axis moves while the button is held down.

   Important: You can jog more than one axis at a time. To jog multiple axes, press and hold more than one axis direction button. The selected axes drive at the feedrate chosen with the `<SPEED/MULTIPLY>` switch. 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 you press the `<AXIS DIRECTION>` button. 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 it recognizes the `<AXIS/DIRECTION>` button. 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 normally jogs the axes, the selected distance and direction, at the feedrate set in AMP for the MED feedrate. Your system installer can select a different feedrate with a specific PAL Program. See documentation prepared by your 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 using a hand pulse generator (HPG). We assume that your system is configured for only one HPG. If your system installer has installed more than one HPG, they may not function as described here. For details, see documentation provided by your system installer.

To jog an axis using 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>`.

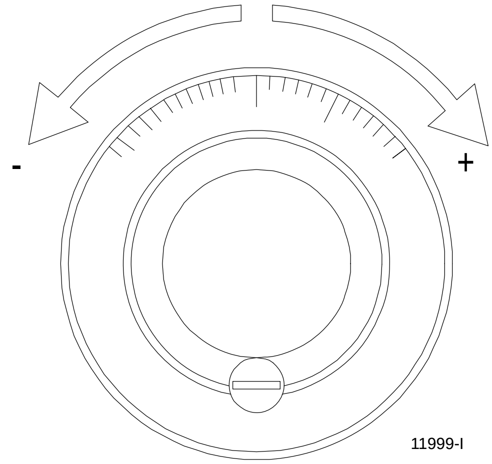
   Important: The x1000 and x10000 selections under `<SPEED/MULTIPLY>` are not usable in HPG jog mode.

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

   Important: You can jog more than one axis at a time using the HPG if your system installer has written PAL to allow it. See documentation prepared by your system installer for details.

4. Typical HPG configuration results in:

If you select:	The direction for the axis is:
clockwise	plus (+)
counterclockwise	minus (-)



4.2 Arbitrary Angle Jog

Your system installer can enable a feature that allows control of the angle of a multiple axis jog. Since this feature is PAL dependent, see your system installer's documentation for its actual operation.

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

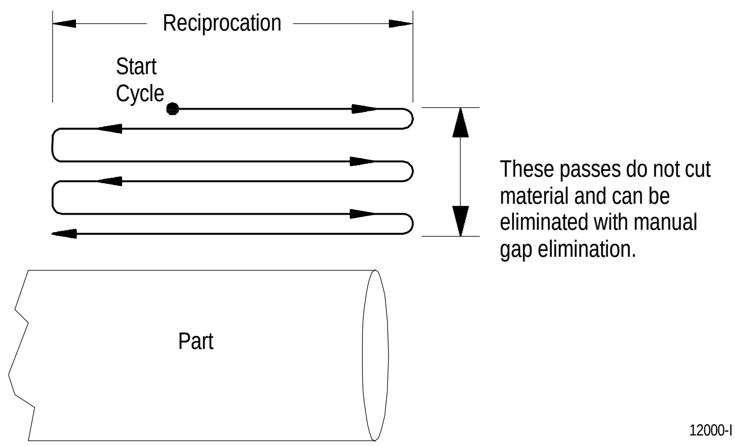
Once this vector has been defined, a jog move can be made along the defined vector by selecting either HPG, incremental, or continuous jogging modes in the normal fashion (see page 4-2). The direction that the wheel travels along the defined vector is determined by your system installer in PAL.

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

4.3 Manual Gap Elimination

The manual gap elimination feature allows the operator to manually jog the grinding wheel without interrupting reciprocation. Using this feature, the operator can speed up the grinding process by skipping over reciprocation strokes that are not making wheel contact with the part.

Figure 4.2
Large Gap between Start Point of Cycle and Actual Part Location



The manual gap elimination feature is available only when reciprocation is active on the control. You can activate reciprocation by using G81 reciprocation startup, or through one of the fixed cycles described in chapter 16 and 17.

You can perform manual gap elimination in either Auto or MDI modes. Your system installer must write PAL to allow execution of manual gap elimination in automatic or MDI mode. See ‘jog on the fly’ in your PAL manual or see documentation provided by your system installer. In AUTO or MDI modes, you can perform manual gap elimination during block execution.

You cannot perform manual gap elimination in these cases:

- Any axis that is performing a circular motion cannot be jogged until that circular motion is complete

  If during the course of the gap elimination, a circular move is performed, the gap elimination jog is aborted until the circular move is completed.

- While the control is performing a block retrace or returning from block retrace (see page 7-30)

- During dresser/wheel radius compensation

If you attempt to perform a manual gap elimination while dresser/wheel radius compensation is active, a change in resulting contour can occur as dresser/wheel radius compensation must be re-initialized at the end of the manual gap elimination jog. Make sure no damage to the part or grinding wheel occurs from this possible re-entry into dresser/wheel radius compensation.

ATTENTION: The manual gap elimination feature should not be used when programming in incremental mode (G91). Be aware that incremental moves continue incrementally from the endpoint of the manual gap elimination jog, not the original start point of the move. Failure to consider these effects change final part contour.

Performing Manual Gap Elimination

To perform manual gap elimination:

1. Select any of the available jogging types as described on page 4-1. You cannot perform arbitrary angle jogs and manual homing operations as manual gap elimination jogs.

2. Select the proper <SPEED/MULTIPLY> value. If you are performing manual gap elimination in automatic or MDI mode, activate the Jog on the Fly feature as described by your system installer's documentation.

ATTENTION: Do not jog the wheel beyond a point where contact with the part is made. Doing so can cause damage to the grinding wheel, the part, or personal injury.

---

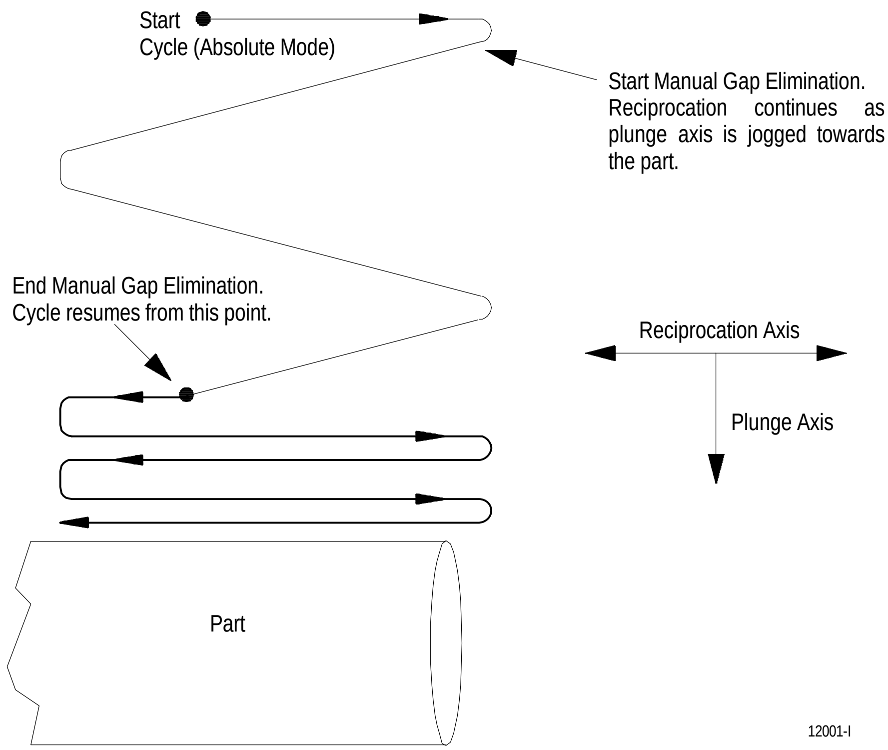
3. Select and jog the axis with the <AXIS/DIRECTION> button or HPG. Jog the axis to a point just before wheel contact with the part is made.

Important: You can jog only non-reciprocating axes using the manual gap elimination feature. Any attempt to jog an axis that is reciprocating results in an error.

Results of Gap Elimination

When you perform manual gap elimination during block execution (as can be the case in AUTO or MDI modes), it bypasses any motion generated by an executing cycle block that occurs above the newly jogged to position. Cycle execution continues from the new grinding wheel location.

Figure 4.3
Cycle Execution Continues With Gap Eliminated



Important: The manual gap elimination feature does not offset the coordinate system. It is an actual jog of the grinding wheel closer to the part location in the current work coordinate system. If you want to jog an offset, see chapter 11 for details.

4.4 Resetting Overtravels

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

Overtravel Condition	Cause
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 grinding wheel 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 your 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.

We describe these 3 causes of overtravel in detail on pages 11-34 through 11-36.

When an overtravel condition occurs, all axis motion is forced to a stop. The control is placed in cycle stop and one of these applicable error messages appears.

If you get this error message:	It means that:
"HARDWARE OVERTRAVEL (-) BY AXIS (X)"	the specified axis has tripped either the + or - hardware limit switch mounted on the machine.
"SOFTWARE OVERTRAVEL (+) BY AXIS (X)"	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)"	an attempt was made by the specified axis to enter the overtravel area defined by either programmable zone 2 or 3.

When an overtravel of any type has taken place, you are not allowed to feed the axis in the same direction as the feed causing the overtravel. Only axis feed in the reverse direction is possible.

How a hardware overtravel condition is reset depends 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. 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 then 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. This does not include homing or jogging an offset. You are not allowed to jog the axis in the direction of the overtravel.

4.5 Mechanical Handle Feed (Servo Off)

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

You can use this feature in conjunction with the digitize feature described in chapter 5.

You can only enable this feature 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. See your system installer's documentation for details on how to activate and use the "Mechanical Handle Feed" feature.

4.6 Removing an Axis (Axis Detach)

Use this feature to remove a rotary table or other axis attachment from a grinder without reconfiguring the system. 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.

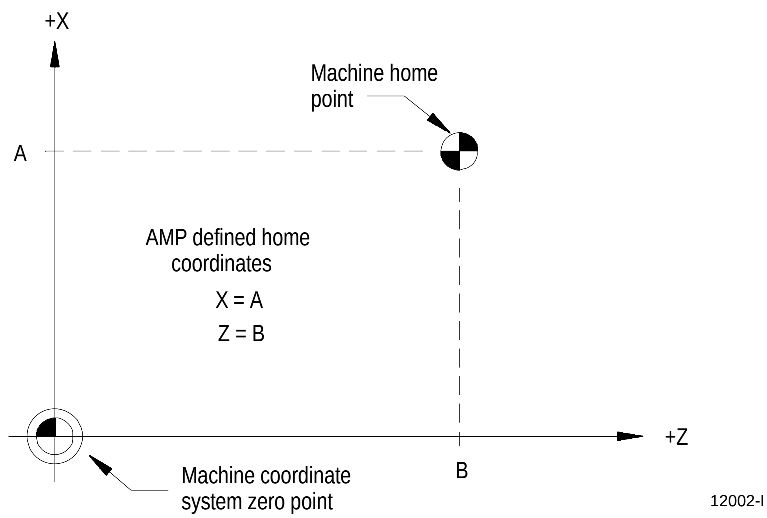
This feature can only be enabled in AMP. You must select the axis as "detached." See your system installer's documentation for the necessary steps involved in physically removing axis hardware from a specific machine.

4.7 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 marker on your 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 you have conducted the machine homing operation. Homing the axis should be the first operation done on the control after power has been turned on.

Figure 4.4
Machine Home



The following procedure describes how you home the control manually by using the pushbuttons on the standard MTB panel. Manual homing can 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: An automatic homing feature is available and is executed by programming a G28 code. For details, see chapter 12.

Figure 4.5
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 (see page 4-1).

3. Determine the direction that each axis must travel to reach the home limit switch. See documentation prepared by your system installer on the location of the home limit switch on your 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 a wrong direction for an axis, it continues to travel in the selected direction until it contacts a hard limit and an overtravel occurs (see chapter 11). Your system installer has the option to enable some buttons or switches (typically cycle stop) through the PAL program to abort a jog operation or prevent the user from homing the axis in the wrong direction. See your system installer's documentation for details.

At this point, the control:

- moves the axis until it trips its home limit switch. Then the axis decelerates to a stop

- reverses the axis direction and moves off the home limit switch at a feedrate specified in AMP

- records the distance to the nearest encoder marker or null position

- 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. The control cancels all active coordinates offsets.

4.8 MDI Mode

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

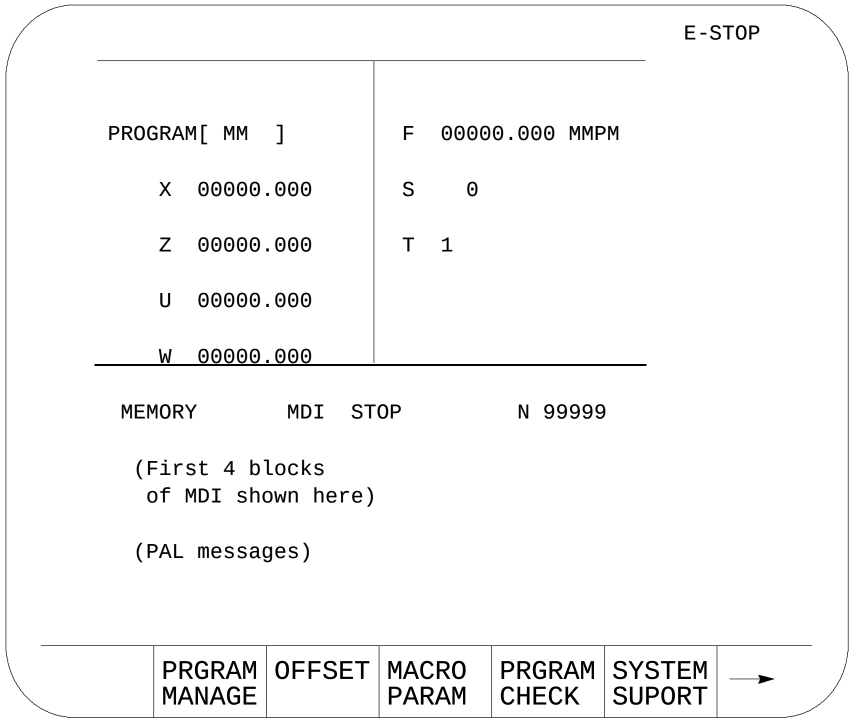
To begin MDI operations, select MDI under `<MODE SELECT>` or press the `{FRONT PANEL}` softkey followed by the left and right cursor keys to select the mode if not equipped with a mode select switch.

Your system installer has the option of disabling G- or M-code AMP-defined paramacro calls in MDI mode. For details on paramacros, see chapter 18.

To insert blocks in an active, executing program using MDI, the control must be in the end of block state to allow the selection of MDI mode. If you press cycle stop to interrupt a program while executing in automatic mode, the control does not allow the selection of MDI. This is because the control is in cycle suspended state, not end of block state and so, a mode change is not accepted.

ATTENTION: When program blocks are executed in MDI, dresser/wheel radius compensation is not allowed. If dresser/wheel radius compensation was previously active before the MDI blocks are executed, it is temporarily cancelled for the execution of the MDI blocks. See chapter 15 for details on the effect of MDI on dresser/wheel radius compensation. Any dresser/wheel radius compensation G codes (G41, G42, or G40) that are programmed in MDI mode affect the dresser/wheel radius compensation mode when compensation is re-activated.

Figure 4.6
Program Display Screen in MDI Mode



You can call subprograms or paramacros within an MDI program; however, there are limitations to the allowable commands. See chapter 10 on subprograms or chapter 20 on paramacros for details on illegal MDI commands for these features.

4.8.1 MDI Basic Operation

Operating procedures in the MDI mode are as follows:

1. With the control in MDI mode, the control is ready to accept standard programming blocks.

2. Key in programming blocks (see the programming section of this manual beginning with chapter 10). Each block, a maximum of 62 characters long, must be separated with an end of block character [EOB]. The blocks entered appear in the input area of the screen (lines 2 and 3). You should enter the complete MDI program on these lines since once you send the blocks to control memory, you cannot edit or add-to them.

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

If you make a mistake while keying in a character before it is sent from the input area, you can edit that character by using the input cursor described in chapter 2.

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

   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 inserting lines 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 (see chapter 2).

4. Press the <CYCLE START> button to continuously execute the MDI blocks in the AUTO or MDI modes. The single block, block retrace and jog retract features are also available for MDI programs (see chapter 7 for details on these features).

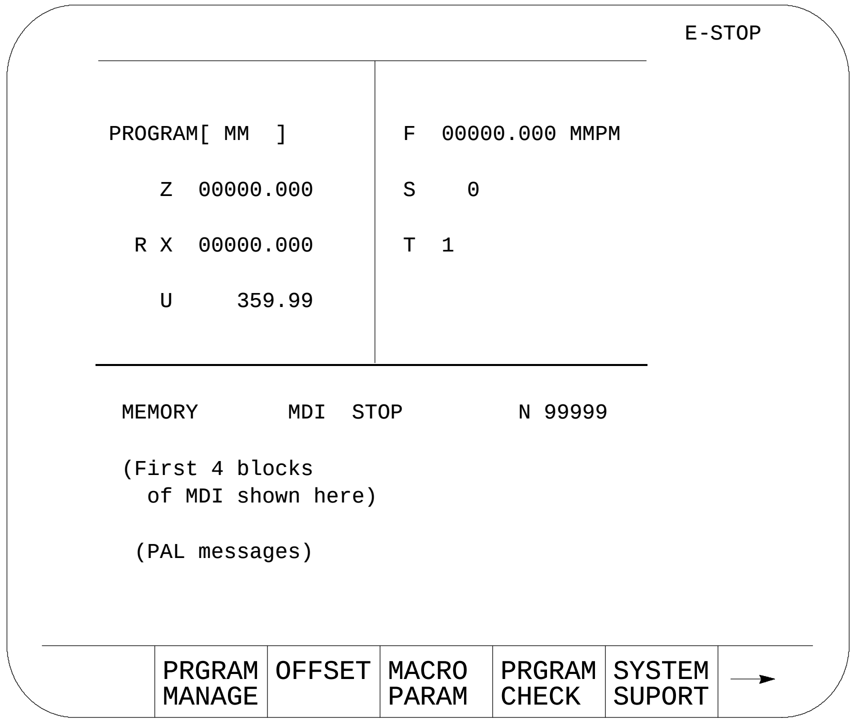
   The control displays an “@” symbol next to any executed MDI blocks.

   The error message:

   “NO MORE MDI BLOCKS”

   appears if you press cycle start in the MDI mode when there are no more executable MDI blocks remaining in memory. If the MDI blocks were entered into an executing part program, you must return the control to automatic mode to continue executing the part program. If you execute the MDI program in the MDI mode, execution is halted when the control encounters the first block of the part program.

Figure 4.7
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 (see chapter 2 for details). By performing a control reset operation, as described in the control, the control erases all MDI blocks that have not been executed in the MDI program (see chapter 3).

END OF CHAPTER

Chapter 5

Editing Programs On Line

5.0 Chapter Overview

This chapter covers the basics of editing programs on line (at the control's keyboard) including:

You can also edit, upload, copy, and perform file management operations on your programs off line at a personal computer. These offline operations are covered in chapter 6.

5.1 Selecting a Program to Edit

This section provides information on how to select a part program for editing. You can only edit on line part programs that you store in control memory. If a part program is on tape or some other storage device and you must edit it on line, copy the program to control memory as described in chapter 5.

You can edit both active and inactive programs. You must exit the edit operation of an active program before the control can execute that program in automatic mode.

If an:	displays to the left of the part program name, it means that the program is:
A	currently active
E	currently open for editing

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

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

The control displays the main part program directory screen:

```
SELECTED PROGRAM:

MAIN       DIRECTORY        PAGE    1  OF    1

  NAME     SIZE          COMMENT

MAIN         2.3
O12345      14.3
RRR          9.3          THIS IS A TEST PROG
TEST         3.9

      4 FILES   120.2  METERS FREE
```

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

2. Select the directory that contains the program to edit. You can select the main directory or the protectable directory. The control displays the main directory as the default directory at power-up. You must select the protectable directory by using the {CHANGE DIR} softkey.

   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. See page 5-42 for additional information on the protectable directory and the {CHANGE DIR} softkey.

3. Select the part program you want to edit by using one of these two methods:

   - Key in the program name of the part program to edit or create

   or

   - Move the cursor to the program name on the program directory screen by using the up and down cursor keys

   Important: If you are creating a new program and using it as a subprogram, see page 10-8 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.

4. Press the {EDIT PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→
↑	REFORM MEMORY	CHANGE DIR				→

5.2 Editing Programs at the Control (on line)

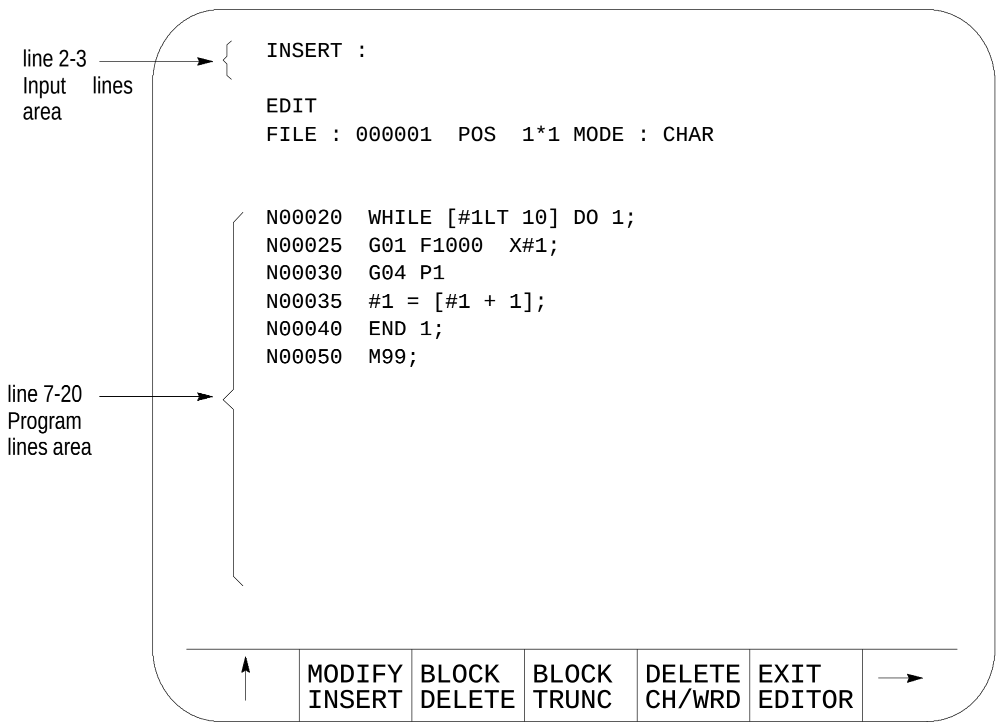
This section provides information on how to edit part programs after a program has already been selected to edit as described on page 5-1.

This section contains the following subsections:

ATTENTION: Any edit operation on a part program is permanent. You cannot discard any changes that you made to a program. The control saves programs in memory at the same time they are edited.

Two major areas of the edit screen are dedicated to displaying specific information:

Line	Area	Content
2 - 3	input lines	when the data is entered, the control displays the character here.
7 - 20	program display lines	the control displays the part program blocks as they are currently been edited.



The following subsections describe in detail, the use of the features available when editing a part program.

5.2.1 Moving the Cursor

{STRING SEARCH}

This section covers moving the cursor in the program display area (lines 7-20 of the CRT). It assumes that you have selected a program to edit as covered on page 5-1.

The input cursor is the cursor shown on the input lines (lines 2 and 3 on the screen). Details on the input cursor are given on page 2-41.

To move the block cursor:	Press these cursor keys on the operator panel:
right and left	right and left
up and down	up and down
up or down one full screen display	up or down cursor keys while holding down the [SHIFT] key

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	→

↑	DIGITZ E					→

2. Move the cursor to the beginning or end of the part program:

(softkey level 4)

↑	FORWRD	REVRSE		TOP OF PRGRAM	BOT OF PRGRAM	

To move the cursor to the:	Press this softkey:
beginning of the part program	{TOP OF PROG}
end of the part program	{BOT OF PROG}

3. Key in the character or character string to search for.

4. Select in which direction to search the part program.

(softkey level 4)

↑	FORWRD	REVRSE		TOP OF PRGRAM	BOT OF PRGRAM

To search the part program in the:	Press this softkey:
forward direction	{FORWRD}
reverse direction	{REVRSE}

If the control cannot find the character or character string, it places the cursor at the end or beginning of the program being searched, depending on the direction of the search.

5. To end the search operation, press the exit [↑] softkey.

Changing Cursor Size

You may want to change the cursor size for editing operations, such as changing, inserting, or erasing. The control has two cursor sizes available:

- single character cursor size -- automatically assigned to the cursor when you access the edit screen
- word cursor size -- used to encompass a word and its value for using erasing, inserting, or changing operations

To modify the cursor size:

1. Press the continue {→} softkey to change the softkey functions.
2. Press the {CHAR/WORD} softkey to select the block length for the cursor (character or word). The block size changes each time you press the {CHAR/WORD} 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	→

↑	DIGITZ E					→

5.2.2 Entering Characters and Blocks

After you have selected a part program to edit, 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. Use the up, down, left and right cursor keys to move the block cursor to the location where you want to add program blocks or characters.

2. Type the program characters you want to enter in the input area. Press the `[EOB]` key (end of block) at the end of each block.

   If you make a mistake while keying in a character before it is sent from the input area, you can edit the input lines as described on page 2-41.

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

   After data has been sent to the program display area of the screen (lines 7-22), you cannot edit it by using the input cursor.

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

5.2.3 Changing and Inserting

`{MODIFY INSERT}`

Changing

You can 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 control displays the MODIFY: prompt on the input line. The control toggles between change and insert each time you press the `{MODIFY INSERT}` 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	→

↑	DIGITZ E					→

2. Use the up, down, left, and right cursor keys to move the block cursor to the location where you need to change characters. The characters to changed appear in reverse video.

3. Key in a new character or word to replace data located within the cursor, then press the [TRANSMIT] key.

   You can only change the data that is within the cursor. If you type in more data than is contained in the cursor, the control inserts the extra data to the right of the cursor. If you type in less data than is contained in the cursor, the control deletes the remaining data in the cursor.

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 page 5-5):

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

You can 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 control displays the INSERT: prompt on the input line. The control toggles between change and insert each time you press the {MODIFY INSERT} 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	

	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 you want to insert 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.
G01X97; Z93;		Result

Example 5.5
Inserting Characters

`To change "X123.0" to "X123.034"`

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 you want to erase 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 PRGRAM	MERGE PRGRAM	QUICK VIEW	CHAR/ WORD	→
↑	DIGITZ E					→

Erasing Commands to the EOB

1. From the edit menu, move the cursor until the first character or word you want to erase is in reverse video.

2. Press the {BLOCK TRUNC} softkey. The control erases all information located from the cursor to the End of Block. You cannot erase the End of Block character using the {BLOCK TRUNC} 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	→
↑	DIGITZ E					→

Example 5.7
Erasing To The End Of 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 until it is located on any character that is in the block you want to delete.

2. Press the {BLOCK DELETE} softkey. The control erases the selected block including the end of block character.

(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					→

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 is completely deleted

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

Erasing A Character Still In The Input Area

The control can also erase characters 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. The control inserts a new character to the left of the cursor automatically when you press any character key. You can move the cursor on the input line by holding down the [SHIFT] key and pressing the left and right cursor keys.

5.2.5 Sequence Numbers {RENUM PRGRAM}

You can assign each block in a part program a five-digit numeric value following an N address. These numbers are referred to as sequence numbers and distinguish one block from another.

You can assign sequence numbers at random to specific blocks or to all blocks. Blocks assigned sequence numbers can be called later by referencing their sequence number.

You can assign or renumber sequence numbers 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 PRGRAM} 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	→

↑	DIGITZ 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: You must enter both the initial sequence number and the incremental value as integer values. If you use a decimal point, the control issues the error message “ILLEGAL VALUE”.

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

4. Select the blocks to renumber by using one of the two choices:

(softkey level 4)

↑	ALL	ONLY N				

If you want to:	Then press:
assign sequence numbers or to reassign sequence numbers to all blocks from the beginning of the part program	{ALL} softkey
assign new sequence numbers to only the blocks that already have sequence numbers	{ONLY N} softkey

ATTENTION: When the control performs a renumber operation, features in the part program that reference a specific block number may no longer be referencing the correct block. For example, a paramacro GOTO 10 command may not call the correct block after the renumber operation. When a renumber operations is performed, you must manually go through the program to make sure that these block references are correct.

5.2.6 Merging Part Programs {MERGE PRGRAM}

You can 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. Use the up, down, left and right cursor keys to move the block cursor to the location in the program display area just past the location in the program where you want to insert another program. The control inserts the merged program to the left of the cursor. Each end of the block statement creates a new line.

2. Press the continue softkey {→} to change the softkey functions.

3. Press the {MERGE PRGRAM} softkey. The prompt "MERGE PRGRAM:" appears on the input line of the screen.

```
(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					→

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

```
(softkey level 4)
```

↑	EXEC					

5.2.7 Exiting Edit Mode

When you are editing a program, all changes and additions that you make are saved immediately in the control's memory. There is no formal "save" command to be executed.

You cannot "quit," "abandon," or "abort" an edit session and restore the original version of a program you have been editing. For that reason, it is good practice to copy a program (see page 5-41) prior to editing if there is a chance the original version is needed.

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 PRGRAM	MERGE PRGRAM	QUICK VIEW	CHAR/ WORD	→

↑	DIGITZ E					→

Important: Do not press the Exit { ↑ } softkey to leave a program being edited. You must use the {EXIT EDITOR} softkey to close the program editor.

Exiting using the Exit { ↑ } softkey (shown on the CRT as an up arrow) causes the following problems:

- the program editor remains open for that program
- the open editor prevents execution of that program (the error PROGRAM BEING EDITED appears)
- no other programs can be edited

  Even though it appears that you can select other programs for editing, when you press the {EDIT PRGRAM} softkey, the previous program remains in the editor and appears on the edit screen. No error or warning message is generated.

5.3 Programming Aids QuickView

This section contains the following subsections:

Topic:	On Page:
Using {QPATH+ PROMPT} Sample Patterns	5-19
G Code Format Prompting {G CODE PROMPT}	5-23
Grinder Cycle Format Prompting	5-25
Selecting a QuickView™ Plane	5-27

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 automatically generates the required block(s) to insert into an existing part program.

The QuickView feature aids 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 page 12-11 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
- Grinder Prompting – a selection of the more commonly used grinder cycles with prompts for the necessary words to program it
- Plane Select – a selection of the planes in which the different QuickView features are programmed

Important: Before accessing a QuickView screen any active graphics or Quick Check graphics features must be disabled. If you attempt to access a QuickView screen without first deactivating both graphics features, the control issues the message MUST DISABLE RUN-TIME GRAPHICS. The {DISABL GRAPHS} softkey becomes available, letting the operator disable graphics from the current location in the softkey tree.

Use these steps to select the QuickView features:

1. Select a program for editing as covered on page 5-1.
2. From the edit menu, press the {QuickView} 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	→

↑	DIGITZ E					→

The softkey functions change to those indicated below:

(softkey level 4)

↑	QPATH+ PROMPT	G CODE PROMPT	GRINDR PROMPT		PLANE SELECT

See the following subsections for information about using the 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. G codes are either planar, or non-planar.

- Planar G Codes – Planar G codes are used by 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 5-27. 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 then 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 then the next linear axis defined that is not already in the QuickView plane is used.
- Non-Planar G Codes - Non-planar G codes are used by 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 linear axes configured.

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 where the asterisks is present.

5.3.1 Using {QPATH+ PROMPT} Sample Patterns

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

Use this pattern:	When you are programming this geometry:	And you know the:
{CIR ANG PT}	an arc to a line	arc radius and the taper angle of a line
{CIR CIR}	an arc to another arc	coordinates of the arc centers
{ANG CIR PT}	a line to an arc	taper angle of a line and the arc radius
{1ANG PT}	a line	taper angle and either of the end point coordinate values
{2ANG 1PT}	two successive lines	angles of two tapered lines
{2ANG 1PT R}	two lines joined by an arc	angles of two tapered lines
{2PT R}	two lines joined by an arc	intersection point of two lines
{2ANG 1PT C}	two lines joined by a chamfer	angles of two tapered lines
{2PT C}	two lines joined by a chamfer	intersection point of two lines
{2ANG 2PT R}	three lines joined by and tangent to two arcs	taper angles of two lines and two arc radii
{3PT R}	three lines joined by an arc between two adjacent lines	two intersection points
{2ANG 2PT C}	three lines joined by chamfers	taper angles of two lines and one intersection point
{3PT C}	three lines joined by chamfers	two intersection points
{2ANG 2PT RC}	three lines joined by an arc and a chamfer	taper angles of two tapered lines and one intersection point
{3PT RC}	three lines joined by an arc and a chamfer	two intersection points
{2ANG 2PT CR}	three lines joined by an arc and a chamfer	taper angles of two lines and one intersection point
{3PT CR}	three lines joined by an arc and a chamfer	two intersection points

Angle of a line, corner radius, and chamfer size are often necessary for a sample pattern in QuickPath Plus prompting. These prompts in QuickPath Plus prompting refer to these drawing dimensions:

If you see a:	It means:
A	Angle
,R	Corner radius
,C	Chamfer size
L	Length of line

For more information regarding these designations, see page 12-11 on programming QuickPath Plus and page 12-22 on corner radius and chamfer.

Your system installer can select a different address for angle A in AMP. See your system installer's 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 appear in reverse video indicate that this parameter is optional and you do not need to enter it in this block. Prompts that appear 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.

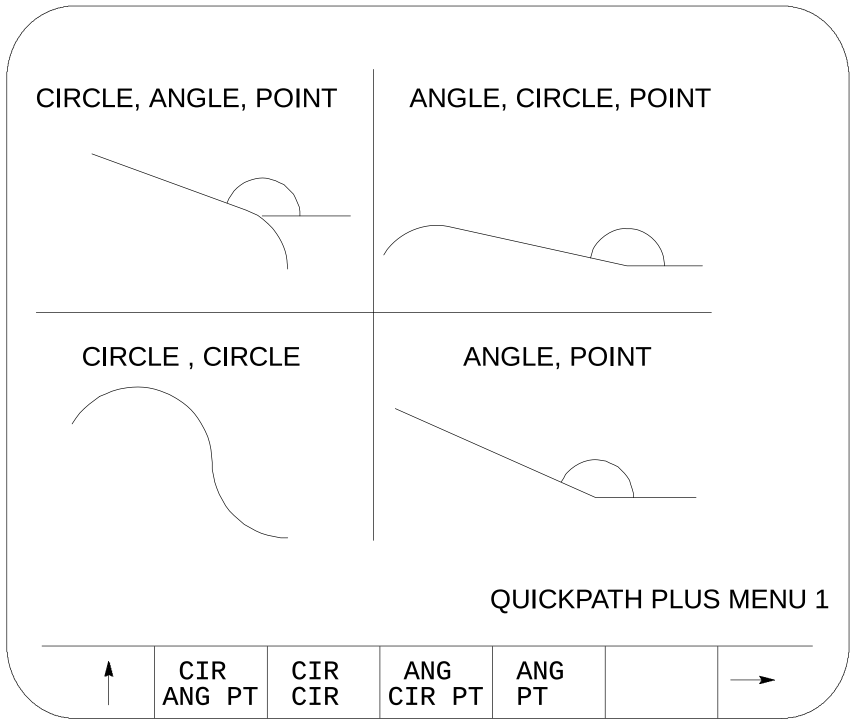
Follow these steps to use the QuickPath Plus QuickView feature.

1. From the QuickView menu, press the {QPATH+ PROMPT} softkey.

   (softkey level 4)

↑	QPATH+ PROMPT	G CODE PROMPT	GRINDR PROMPT		PLANE SELECT

The control displays the first QuickPath Plus sample pattern screen:



2. Select a sample pattern matching the part geometry you want to program and press the corresponding softkey.

   To select other sample patterns, press the continue softkey {→}.

3. After selecting the sample pattern, enter values for the parameters in this way:

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

   - Type in the data on the input line of the CRT and press the `[TRANSMIT]` key. The data typed in is entered as the data for the selected parameter.

4. After you enter all data for the pattern, press the {STORE} softkey to store the data.

(softkey level 6)

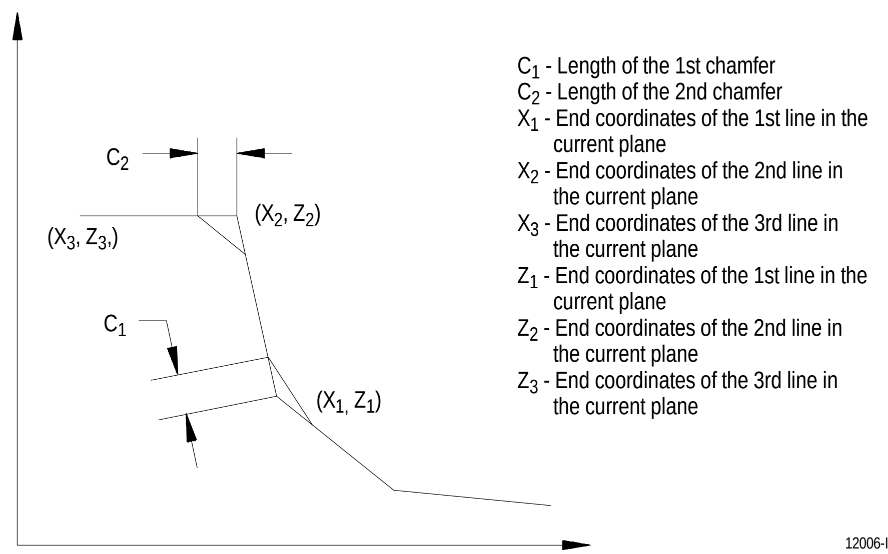
↑	STORE					

The control generates the necessary block(s) to create the axis moves. The control displays these blocks in the input area next to the EDIT: prompt. You can edit these blocks in the input area as covered on page 2-41.

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 you want to insert the new blocks. Then press the [TRANSMIT] key. The generated blocks are 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 you press the {3PT C} softkey, the control displays the prompt screen for that sample pattern. The following figure shows what data must be entered for that prompted screen to generate the necessary paths correctly.



5.3.2 G Code Prompting {G CODE 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 to show the programmer a sample of what the G-code parameters are used for.

Grinder surface or cylindrical G codes are listed under the {GRINDER PROMPT} softkey.

To use the G-code prompting menus:

1. From the QuickView menu, press the {G CODE PROMPT} softkey.

   (softkey level 4)

↑	QPATH+ PROMPT	G CODE PROMPT	GRINDR PROMPT		PLANE SELECT

   The control displays the following G-code prompt select screen:

```
G CODE PROMPTING MENU DISPLAY          PAGE  1 OF  3

G00/01        RAPID/LINEAR INTERPOLATION
G02/03        CIRCULAR/HELICAL INTERPOLATION, CW/CCW
G04           DWELL
G07/G08       RADIUS/DIAMETER PROGRAMMING
G09/61/62/    CUTTING MODE SELECTION
 64
G10L2         WORK COORDINATE SYSTEM TABLE ENTRY
G10L10&L11    TOOL OFFSET TABLE ENTRY
G14.1/14      SCALING ENABLED/DISABLED
G20           SINGLE PASS O.D. AND I.D. ROUGHING
G24           SINGLE PASS ROUGH FACING CYCLE
```

↑	SELECT				

2. Position the cursor at the G code to prompt by using the up and down cursor keys. The selected G code appears in reverse video.

3. Once you select the correct G code, press the {SELECT} softkey. The control displays a screen with prompts for that G code.

4. Use the up and down cursor keys to select the parameters you want to change or enter. The selected item appears 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 appears on the drawing accompanying the prompt screen.

   Prompts that appear in reverse video indicate that this parameter is optional and does not need to be entered in this block. Prompts that appear 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. Enter the data for that parameter on the input line of the CRT and press the `[TRANSMIT]` key.

6. After you enter all data for the G code, press the `{STORE}` softkey to store the data.

   (softkey level 6)

↑	STORE					

   The control generates the necessary G-code block. The control displays the generated block in the input area next to the EDIT: prompt. You can edit this block in the input area by using the techniques covered on page 2-41.

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 you want to insert the new blocks. Press the `[TRANSMIT]` key to 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.

5.3.3 Grinding Cycle Prompting {GRINDR PROMPT}

Grinder cycle prompting aids the operator in programming surface or cylindrical grinding cycle blocks by prompting the programmer for the necessary parameters and giving a graphical representation of the cycle operation.

For G-code prompts other than these cycles, see page 5-23.

To use the grinder cycle prompting function, follow these steps:

1. From the QuickView menu, press the {GRINDR PROMPT} softkey.

   (softkey level 4)

↑	QPATH+ PROMPT	G CODE PROMPT	GRINDR PROMPT		PLANE SELECT	

   One of the following two screens appear depending on your control type configuration in AMP.

   If you have configured a cylindrical grinder, this screen appears:

```
                                                    E-STOP

      GRINDER PROMPT MENU  DISPLAY

    G80     CANCEL OR END FIXED CYCLE
    G81     RECIPROCATION
    G81.1   RECIPROCATION PREDRESS
    G82     INCREMENTAL FACE
    G82.1   INCREMENTAL FACE PREDRESS
    G83     INCREMENTAL PLUNGE
    G83.1   INCREMENTAL PLUNGE PREDRESS
    G84     MULTI FACE
    G84.1   MULTI FACE PREDRESS
    G85     MULTI PLUNGE
    G85.1   MULTI PLUNGE PREDRESS
    G86     SHOULDER
    G86.1   SHOULDER PREDRESS
    G87     FACE PLUNGE SHOULDER
    G87.1   FACE PLUNGE SHOULDER PREDRESS
    G88     DIAMETER PLUNGE SHOULDER
    G88.1   DIAMETER PLUNGE SHOULDER PREDRESS
```

↑	SELECT					

If you have configured a surface grinder, this screen appears:

```
                                                        E-STOP

        GRINDER PROMPT MENU  DISPLAY

      G80     CANCEL OR END FIXED CYCLE
      G81     RECIPROCATION
      G81.1   RECIPROCATION PREDRESS
      G82     PLUNGE
      G82.1   PLUNGE PREDRESS
      G83     INCREMENTAL PLANE 1
      G83.1   INCREMENTAL PLANE 1 PREDRESS
      G84     INCREMENTAL PLANE 2
      G84.1   INCREMENTAL PLANE 2 PREDRESS
      G85     CONTINUOUS PLANE 1
      G85.1   CONTINUOUS PLANE 1 PREDRESS
      G86     CONTINUOUS PLANE 2
      G86.1   CONTINUOUS PLANE 2 PREDRESS

   ↑   | SELECT |        |        |        |        |
```

2. Position the cursor at the desired cycle to prompt by using the up and down cursor keys. The selected G code cycle appears in reverse video.

3. Once you select the correct cycle code, press the {SELECT} softkey. The control displays a screen with prompts for that cycle along with a graphical representation of the cycle.

4. Use the up and down cursor keys to select the parameters you want to change or enter. The selected parameter appears 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 appears on the drawing accompanying the prompt screen.

   Prompts that appear in reverse video indicate that this parameter is optional and need not be entered in this block. Prompts that appear 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 data for that parameter on the input line of the CRT and press the [TRANSMIT] key.

6. After you enter all data for the G code, press the {STORE} softkey to store the data.

(softkey level 6)

↑	STORE					

The control generates the necessary G code block. The control displays the generated block in the input area next to the EDIT: prompt. You can edit this block in the input area by using the techniques covered on page 2-41.

7. To enter the blocks in the program you want to edit, move the block cursor in the program display area just past the location in the program where you want to insert the new blocks. Press the [TRANSMIT] key to 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.

For details on the parameters used in a specific cycle, see chapter 18 for details on turning cycle operation and parameters.

5.3.4 Selecting a QuickView Plane

You can select the plane used to program the different QuickView features. This feature determines which plane is displayed during QuickView prompting and the axis names displayed for the prompts. You can only use primary planes with the QuickView feature. You cannot select any parallel planes with the QuickView feature.

Your system installer determines the primary planes established by G17, G18, and G19 in AMP. See page 11-33 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	G CODE PROMPT	GRINDR PROMPT		PLANE SELECT	

The control displays the currently selected plane. The right side of the screen lists 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. Press the softkey that corresponds to the plane you want to program in (G17, G18, or G19). See documentation prepared by your system installer for details on the planes selected by these G-codes.

   The display changes to show the selected plane.

   (softkey level 5)

↑	SET	ANGLED PLANE	G17	G18	G19	

3. If the plane displayed is the plane in which you want to program the QuickView feature, 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. When the control executes this program, the correct planes are established.

Important: Special consideration for planes should be made when QuickView is used to program an angled-wheel grinder. Refer to page 14-6 for details.

5.4 Digitizing a Program (Teach)

This section contains the following subsections:

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

You can use the digitize feature in any operating mode (AUTO, MANUAL, or MDI). It is accessed through the regular edit menu which allows any of the other edit features to be used on blocks that are generated using the digitize feature.

To use the digitize feature:

Important: The following description covers the use of softkeys to perform digitizing. Your system installer may have written PAL to allow some other method of digitizing. If this is the case, see documentation provided by your 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 page 5-1 on selecting a program.

2. From the edit menu, press the {DIGITIZE} 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	→

↑	DIGITZ E					→

3. Position the grinding wheel using one of the following methods. The grinding wheel 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
   - Manually move the axes using any means as long as the encoder is still actively recording the wheel position (see the documentation prepared by your system installer)

4. Press the {MODE SELECT} softkey to change any of the following programming modes while digitizing a program:

(softkey level 4)

↑	LINEAR	CIRCLE 3 PNT	CIRCLE TANGNT	MODE SELECT		

- Inch/metric
- Absolute programming/incremental programming
- Change planes G17, G18, or G19
- Radius programming/Diameter programming

5. Press the softkey that corresponds to the mode you want to change.

(softkey level 5)

↑		INCH/ METRIC	ABS/ INCR	PLANE SELECT	DIA/ RADIUS	→

The control displays the mode that the next block is programmed in in the upper right hand corner of the screen. The modes and their abbreviations are listed in Table 5.A.

Table 5.A
Changing Programming Modes 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}
DIAMETER MODE	DIA	G08	{DIA/ RADIUS}
RADIUS MODE	RAD	G07	{DIA/ RADIUS}
INCH MODE	INCH	G20	{INCH/ METRIC}
METRIC MODE	METRIC	G21	{INCH/ METRIC}

Each time you press the softkey corresponding to the mode, the displayed G-code changes.

6. When the mode you want appears, press the exit { ↑ } softkey to exit the mode select screen.

This returns the control to softkey level 4. The next block generated has the necessary G codes 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.

7. Determine if the next move is linear or circular.

```
(softkey level 4)
```

↑	LINEAR	CIRCLE 3 PNT	CIRCLE TANGNT	MODE SELECT		

If the next move is:	Then press this softkey:
linear	{LINEAR} (see page 5-31)
circular	{CIRCLE 3 PNT} if 3 points on the arc are known (see page 5-33) or {CIRCLE TANGNT} if the endpoint of the arc and the line that is tangent to the start point of the arc is known (see page 5-35).

5.4.1 Linear Digitizing

To digitize a linear move:

1. Press the {LINEAR} softkey.

   When you press the {LINEAR} softkey, the control sets the current wheel position as the start point of a linear move. The screen changes to display the current wheel location in large display characters:

```
DIGITIZE:
                                          E-STOP

 TARGET[ MM  ]

R X    -     0.000

  Z    -     0.000



  F       0.000 MMPM S     00
```

↑			STORE END PT		EDIT & STORE	

2. Reposition the wheel at the desired end point of the linear move by 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 wheel position (see the documentation prepared by your system installer)

3. Press either the {STORE END PT} or the {EDIT & STORE} softkeys after the axes are positioned at the end point of the linear move. This records the current wheel 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 can be edited as covered on page 2-41. The control then inserts this block at the location of the block cursor when you press the [TRANSMIT] key.

Important: To abort the linear digitize operation, press the exit { ↑ } softkey at any time before you press the {STORE END PT} or {EDIT & STORE} softkeys. The control returns to softkey level 4. Then select another digitize feature (by pressing one of the softkeys) or press the exit { ↑ } softkey to return the control to the regular edit session.

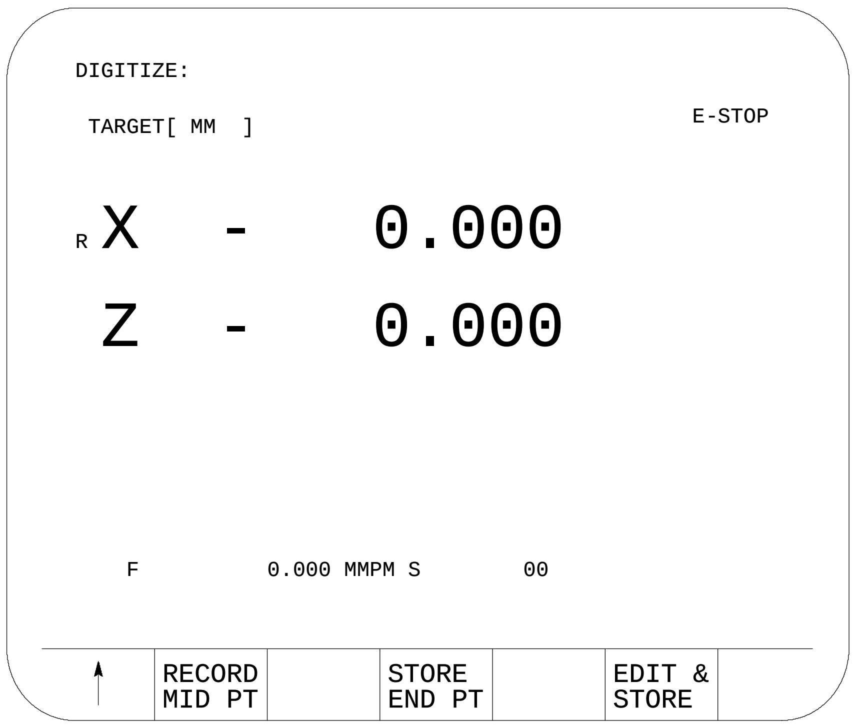
5.4.2 Digitizing an Arc (3 Points)

To digitize a 3 point arc:

1. Press the {CIRCLE 3 PNT} softkey.

   When you press the {CIRCLE 3 PNT} softkey, the control sets the current wheel position as the start point (first point of 3 that is necessary to describe an arc) of a circular move.

   The screen changes to display the current wheel location in large display characters:



2. Reposition the wheel at any point on the arc <u>between</u> the start and the end point by 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 wheel position (see documentation prepared by your system installer)

3. After storing the second point on the arc, reposition the axes at the end point of the arc.

4. Press either the {STORE END PT} or the {EDIT & STORE} softkeys to store this block as a circular block. This records the current wheel 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 can be edited as covered on page 2-41. The control then inserts this block at the location of the block cursor when you press the [TRANSMIT] key.

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 returns to softkey level 4. Then select either another digitize feature (by pressing one of the softkeys) or press the exit { ↑ } softkey to return the control to the regular edit session.

5.4.3 Digitizing An Arc Tangent at End Points

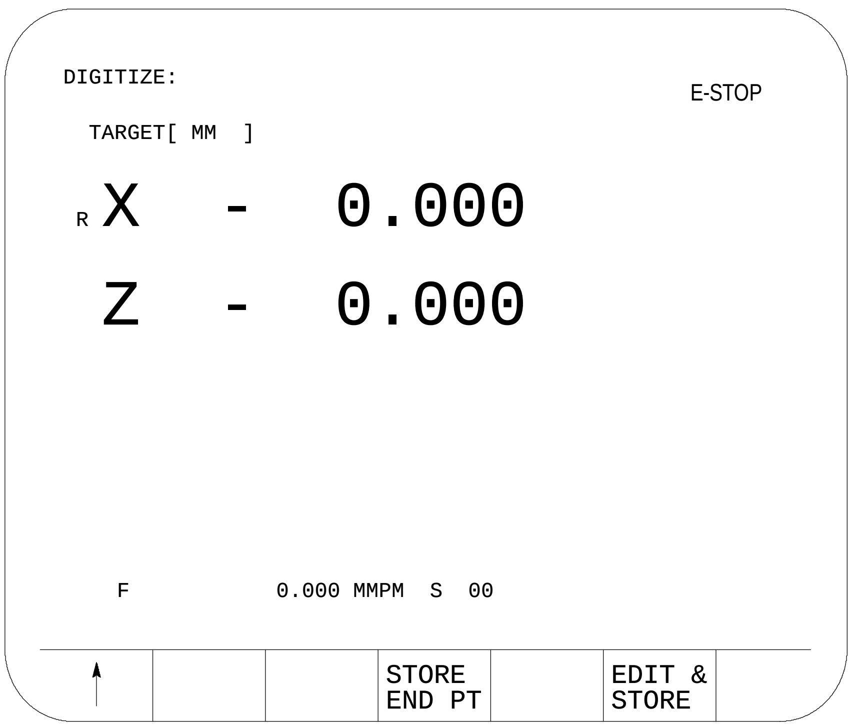
To digitize an arc that is tangent at the endpoint of the previous path:

1. Press the `{CIRCLE TANGNT}` softkey.

   When you press the `{CIRCLE TANGNT}` softkey, the control sets the current wheel position as the start point of a circular move. 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: You cannot select the `{CIRCLE TANGNT}` feature as the first motion block in a part program because there is no path for the start of the arc to be tangent to.

   The screen changes to display the current wheel location in large display characters:



2. Reposition the wheel at the end point of the arc by 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 wheel position (see documentation prepared by your system installer)

3. Press either the {STORE END PT} or the {EDIT & STORE} softkeys after the axes have been positioned at the end point of the arc. The control stores the current wheel 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 can be edited as covered on page 2-41. This block is then inserted at the location of the block cursor when you press the [TRANSMIT] key.

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 returns to softkey level 4. Then select either another digitize feature (by pressing one of the softkeys) or press the exit { ↑ } softkey to return the control to the regular edit session.

5.5 Deleting A Program {DELETE PRGRAM}

To delete a part program stored in memory:

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the {DELETE PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

ATTENTION: Once you delete a program from memory, it cannot be recovered. To abort the delete program operation, press the {DELETE NO} softkey.

3. Select one of these two choices:

- Key in 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				

You can delete all programs at once by formatting the RAM disk as described on page 2-42. Formatting the RAM disk, however, deletes all programs from both the main directory and the protected directory. If you do not have access to the protected directory, do not format the RAM disk.

5.6 Renaming Programs {RENAME PRGRAM}

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

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the {RENAME PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. Key in the current program name or cursor down until the desired program is in reverse video. Then type in a comma, the new program name, and press the {RENAME YES} softkey. To abort the operation, press the {RENAME NO} softkey.

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

(softkey level 3)

↑	RENAME YES	RENAME NO				

5.7 Displaying a Program {DISPLY PRGRAM}

The 9/Series control has a part program display feature that lets you view, but not edit, any part program.

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

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Select the input device by using the {INPUT DEVICE} softkey (as covered in chapter 7). This is only necessary if the currently active input device is not the device that the part program to display is currently resident on. 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)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

5. To scroll the part program blocks, hold down the [SHIFT] key, then press the up and down cursor keys. If you display the program from a peripheral device, you can only use the forward direction to display the program.

6. To end the displaying operation, press the exit { ↑ } softkey. The display returns to the program directory screen.

5.8 Comment Display {PRGRAM COMENT}

You can assign each individual program a short comment that is displayed on the program directory screens. Use these comments to help identify a program when selecting it for automatic operation or for editing.

Important: These comments are not normally the same as a comment block made within a part program. Comment blocks are covered on page 10-10. If a comment block is assigned as the first block of the part program, it is also 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 these steps:

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

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Using the up and down cursor keys, select the program to assign a comment. The selected program name appears in reverse video.

3. Press the {PRGRAM COMENT} softkey. The comment softkey is displayed in reverse video and the control displays the prompt "COMMENT:" on line 2 of the screen.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

If a comment has previously been entered, it is displayed to the right of the "COMMENT" prompt. This comment can be edited using the input cursor as covered on page 2-41, or the old comment can 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 using the keyboard keys.

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

5.9 Copying Programs {COPY PRGRAM}

This section contains information on making a duplicate of an existing part program in the control's memory. To input or output a part program from/to a peripheral device, see the sections on inputting or outputting programs covered in chapter 9.

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

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the {COPY PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. Key in or cursor down to the program name of the program you want to copy.

4. Key in a comma followed by a new program name for the duplicate program.

```
COPY: FROM_NAME,TO_NAME
```

5. Press the {MEM TO MEM} softkey.

(softkey level 3)

↑	MEM TO PORT A	PORT A TO MEM	MEM TO PORT B	PORT B TO MEM	MEM TO MEM	

The following message appears:

"FROM: (source program name)
"TO: (new programs name)

Important: The control displays 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 page 9-1.

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

(softkey level 4)

↑	COPY YES	COPY NO				

7. If you want to verify that the copied program identically matches the original, use the {VERIFY PRGRAM} feature covered 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|OFFSET|MACRO |PRGRAM|SYSTEM| -->
      |MANAGE|      |PARAM |CHECK |SUPORT|

      |FRONT |ERROR |PASS- |SWITCH|      | -->
      |PANEL |MESAGE|WORD  |LANG  |      |
```

The control displays the main program directory screen:

```
SELECTED PROGRAM:

MAIN       DIRECTORY        PAGE    1  OF    1

  NAME     SIZE          COMMENT

MAIN         2.3
O12345      14.3
RRR          9.3          THIS IS A TEST PROG
TEST         3.9

      4 FILES   120.2  METERS FREE

  ^   |ACTIVE| EDIT |RESTRT|DISPLY| COPY | -->
  |   |PRGRAM|PRGRAM|PRGRAM|PRGRAM|PRGRAM|
```

2. Press the {CHANGE DIR} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	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
```

↑	REFORM MEMORY	CHANGE DIR	NCRYPT MODE	SET-UP NCRYPT		→

4 FILES   120.2 METERS FREE

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 PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	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 = O    ; = 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 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)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR	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 FILL			

END OF CHAPTER

Chapter 6

Editing Part Programs Off Line (ODS)

6.0 Chapter Overview

This chapter describes the Offline Development System (ODS). The major topics in this chapter include:

Use ODS to write or edit part programs. Once completed these part programs can be downloaded from the workstation to the control. Programs that already exist on the control can be uploaded to the workstation for editing or backup. Programs on ODS can be edited using the screen or text editor that is configured in ODS. Enhancements to this feature can be purchased in a Mini-DNC package from Allen-Bradley. If the Mini-DNC package has been purchased as an option, see its accompanying documentation.

We make these assumptions:

- ODS has been installed on an IBM PC XT/AT or compatible computer that is 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
- your programmer understands the basics of the ODS system and how it works

For additional information, see the ODS Software User Manual, publication MCD-5.1.

Important: Be aware that some features described here may not be available with your ODS. Some may require the purchase of the Mini-DNC package to be functional.

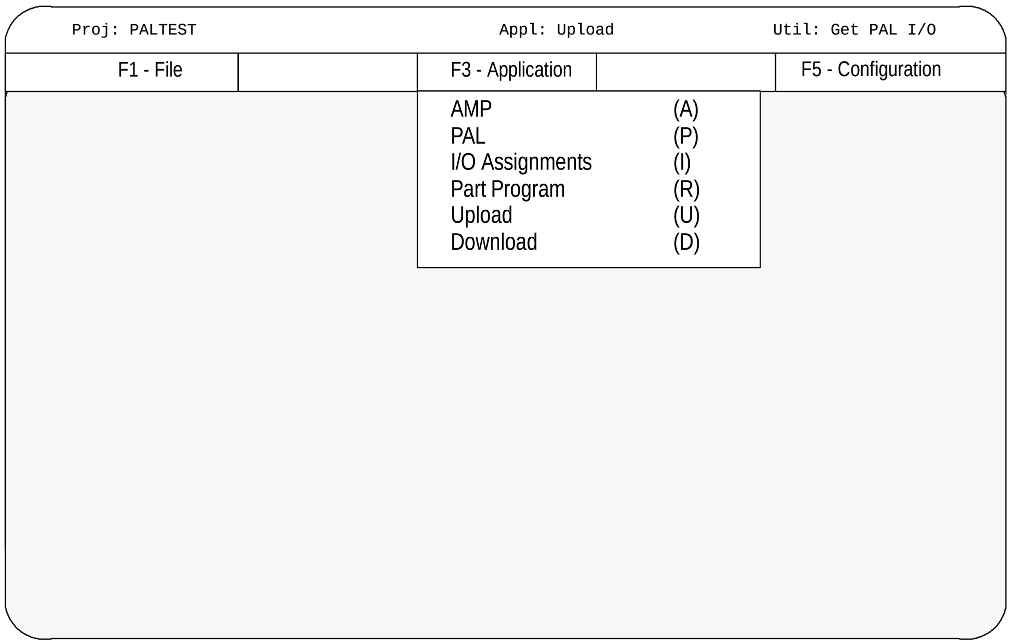
6.1 Selecting the Part Program Application

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:



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.

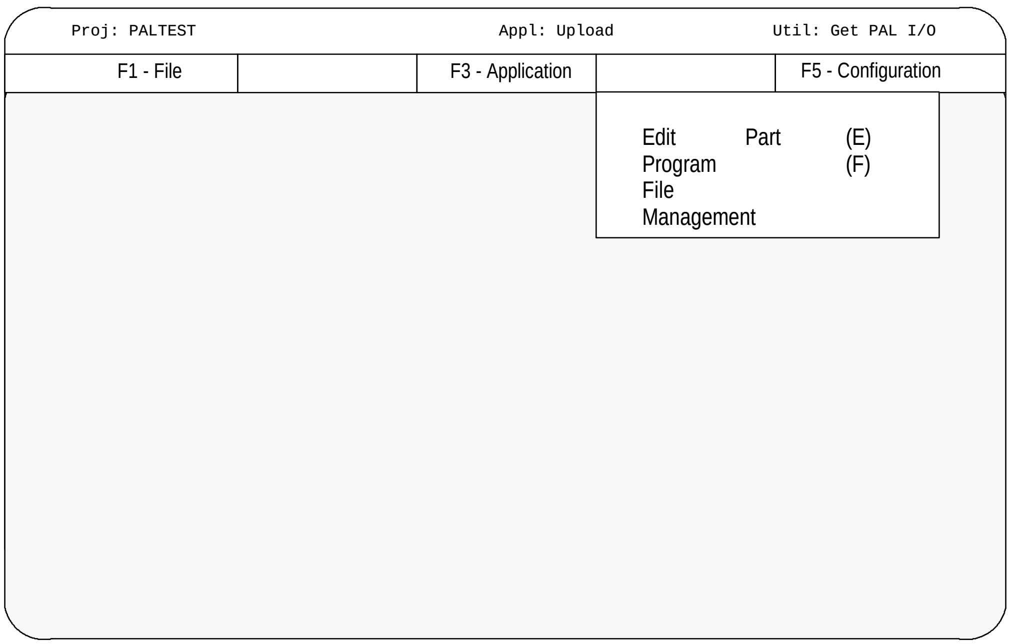
6.2 Editing Part Programs Off Line

Use the Edit Part Program utility of ODS to edit part programs on a workstation. Programs that already exist on the control can be uploaded to the workstation for editing. These programs or programs created using ODS can be edited using the screen or text editor that is configured in ODS. To edit part programs through ODS:

1. Select the Part Program Application (see above).

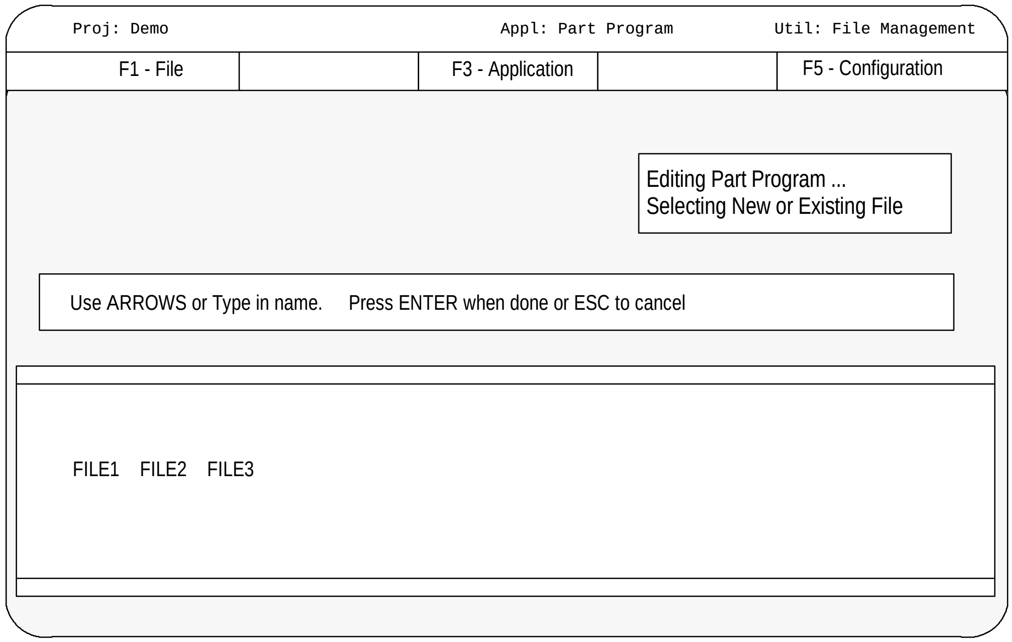
2. Press `[F4]` to pull down the Utility menu:

   The workstation displays this screen:



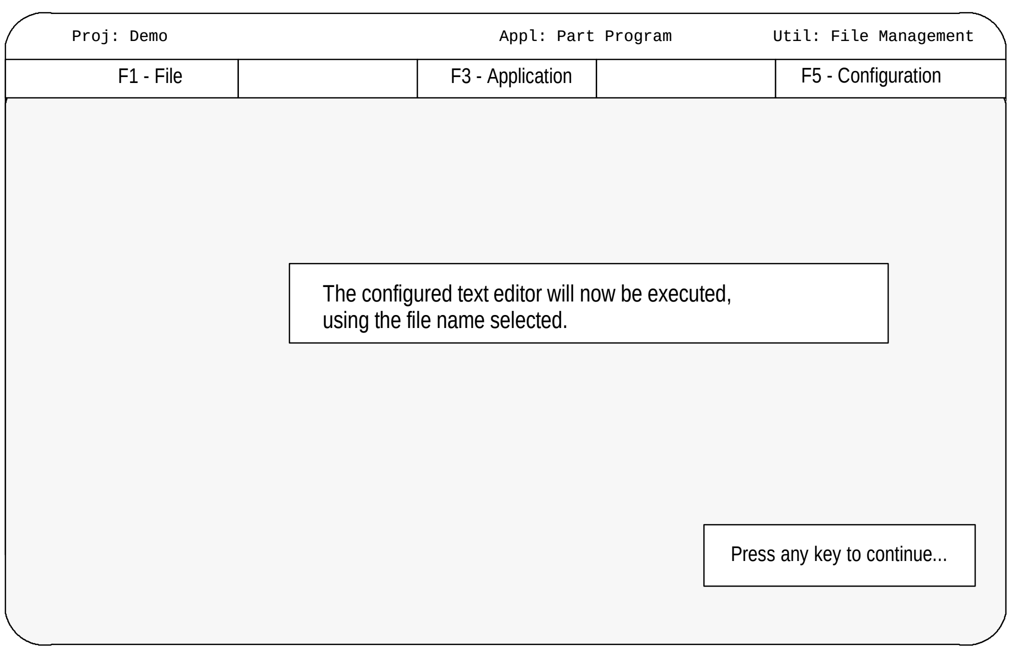
3. Press `[E]` to select the Part Program option.

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 explaining the text editor:



Use the configured screen or text editor 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, such as ending an edit session, displaying a program, etc., see the documentation provided with the screen or text editor.

Details on programming blocks of the 9/Series control are found in the programming manual.

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 automatically inserts the end of block statements ";" at the end of each line when the program is downloaded to the control.

6.3 Connecting the Workstation to the Control

The following sections require that the workstation be connected to the control or storage device. Connect the workstation to the control or storage device with the RS-232 serial interface cable. See your integration manual for specific cable information.

Use the interface cable to connect the RS-232 interface port on the rear of the workstation to port B on the control or to the RS-232 port on the storage device. For information about connecting the workstation to a storage device, see your integration manual.

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.

You must configure port B 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 SUPORT} 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, see 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.

If the serial communication parameters of port B do not correspond to the serial communication parameters of the workstation, see your programming manual.

6.4 Downloading Part Programs from ODS

After using the part program edit utility to create or edit a part program file off line, the programmer can download this part program to the control or to a storage device by using the Download application of ODS.

Important: When a program is downloaded from ODS to the control, it is automatically inserted into the Main program directory on the control (unless the program is encrypted). An encrypted program is automatically loaded into the protected directory. See page 5-45. The control automatically inserts the end of block statements ";" at the end of each line when the program is downloaded to the control.

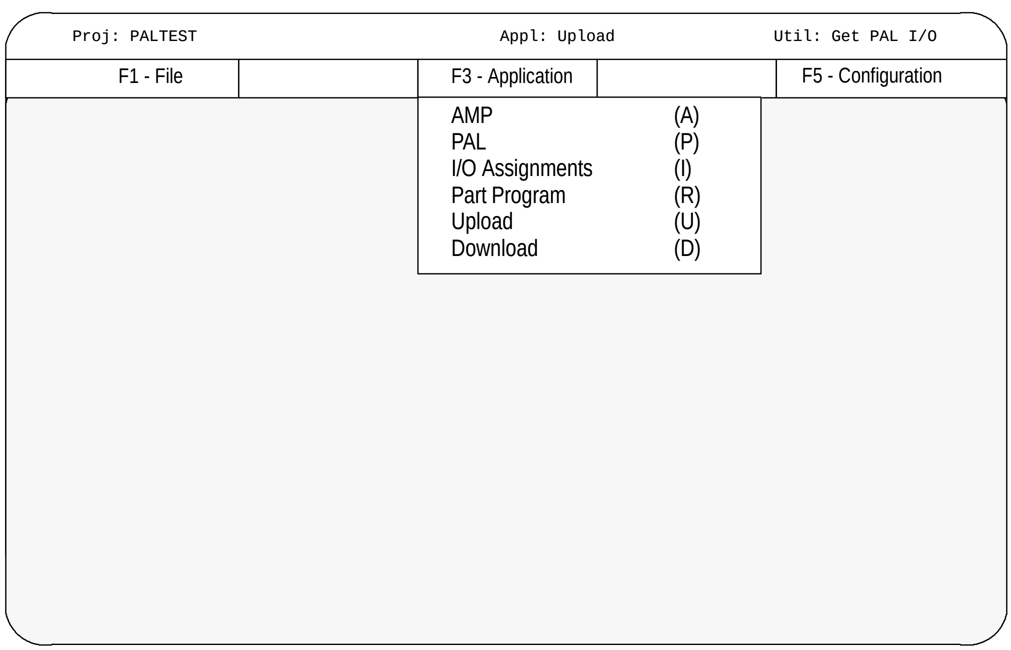
To download a part program from ODS to the control's memory, follow these steps:

1. Interface the workstation with the control. See page 6-5.

2. Return to the main menu line of ODS.

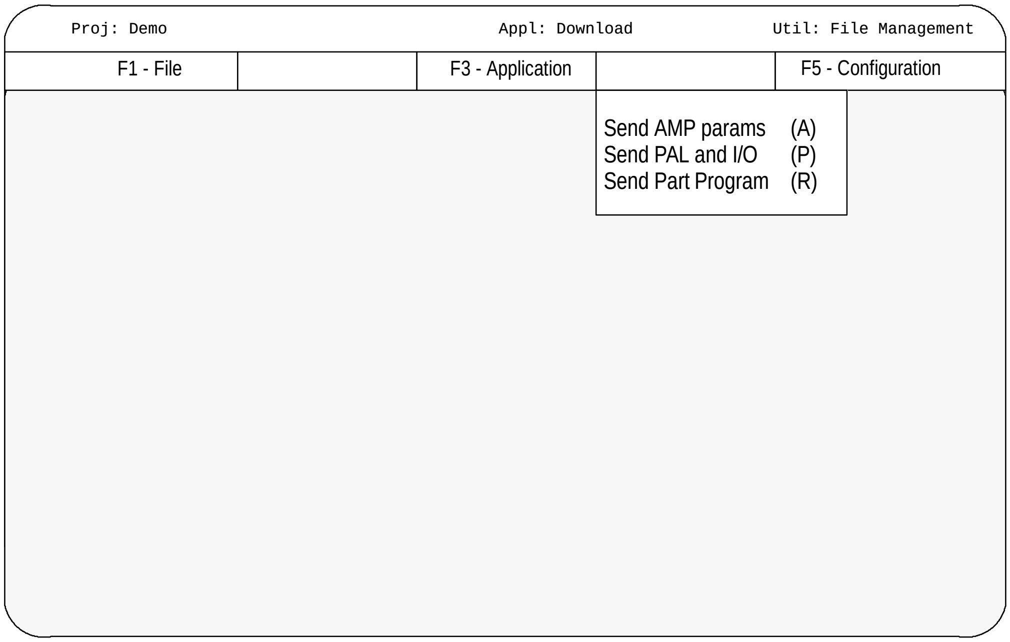
3. Press `[F3]` to pull down the Application menu.

   The workstation displays this screen:



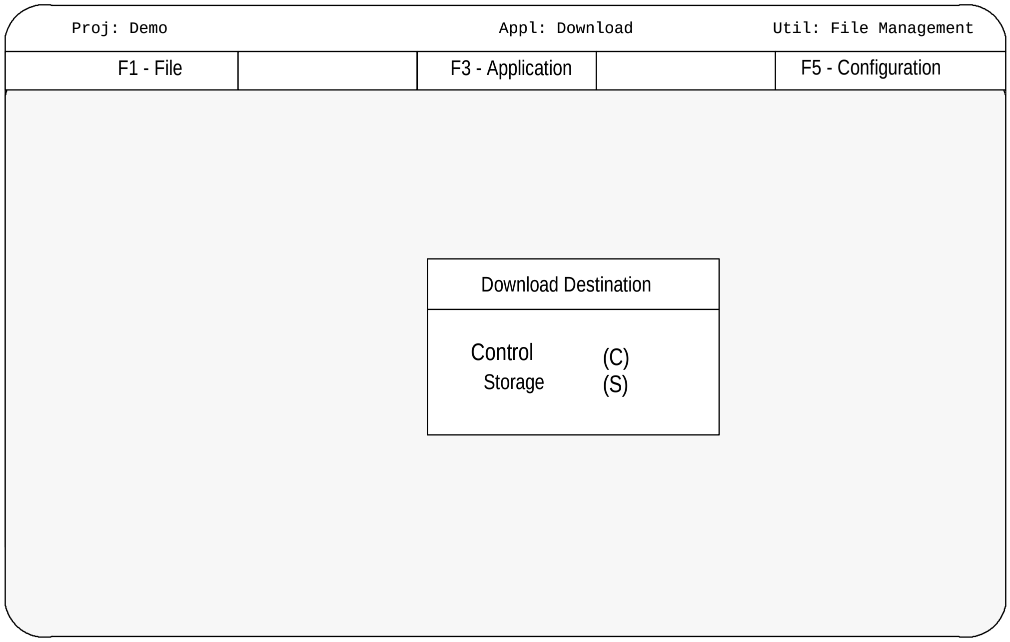
4. Use the arrow keys to highlight the Download application then press `[ENTER]`, or press `[D]`.

5. Press `[F4]` to pull down the Utility menu.



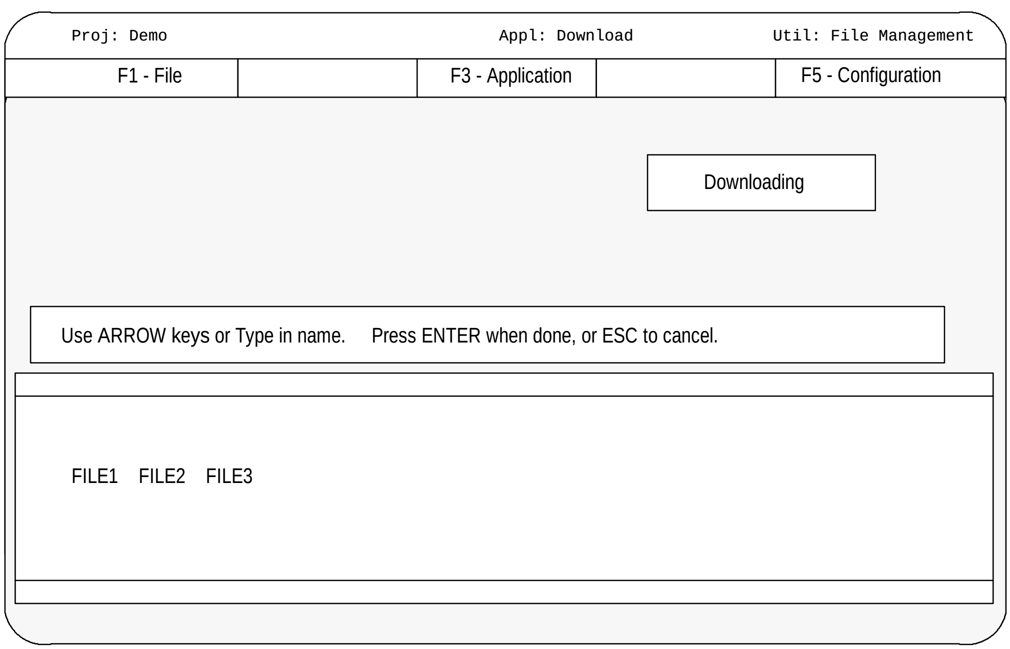
6. Use the arrow keys to highlight the Send Part Program option then press`[ENTER]`, or press `[R]`.

   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]`.

   Important: It is possible to upload more than one part program by using wildcards (“*” or “?”) in place of all or part of a file name. See the workstation’s DOS manual for additional information about using wildcards.

   If the selected part program file name already exists on the control, the workstation displays this screen:

```
Proj: Demo                         Appl: Download         Util: Get Part Program
```

F1 - File		F3 - Application		F5 - Configuration

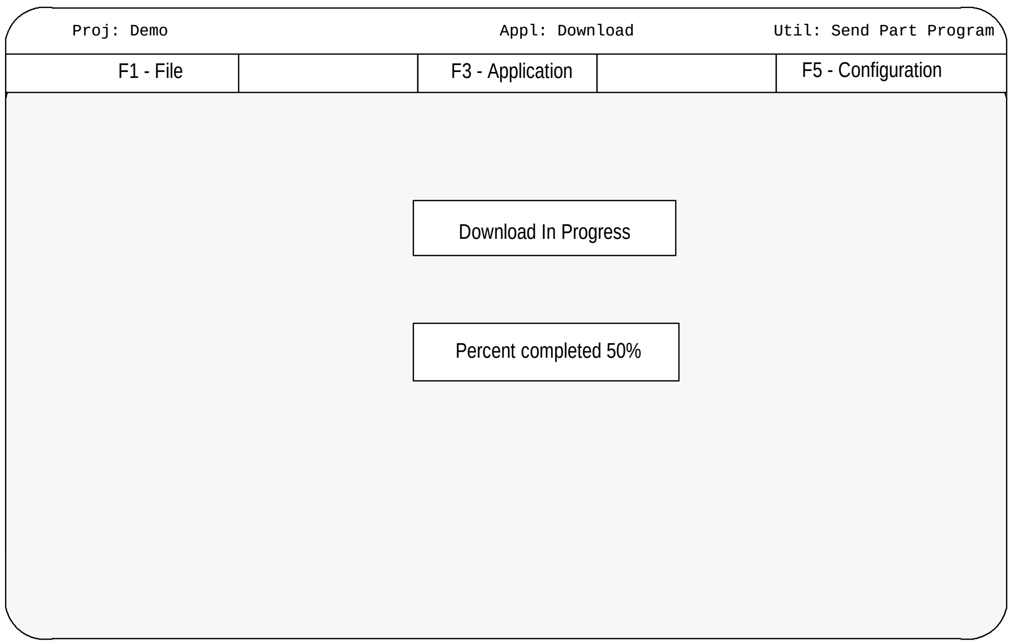
File Already Exits Enter Option	
Rename existing file	(R)
Overwrite existing file	(O)
Abort current file	(A)

Important: The currently active or open part program on the control cannot be renamed or overwritten during a download procedure.

If you select this option:	then:
rename existing file	the workstation renames the existing file, which has the same name as the file being uploaded from the workstation. The workstation displays the part program files that have been stored. Type in the new name for the existing part program on the control.
overwrite existing file	the part program file being downloaded overwrites the file having the same name on the control.
abort current file	the download process is discontinued and the workstation prompts the programmer for additional files to download.

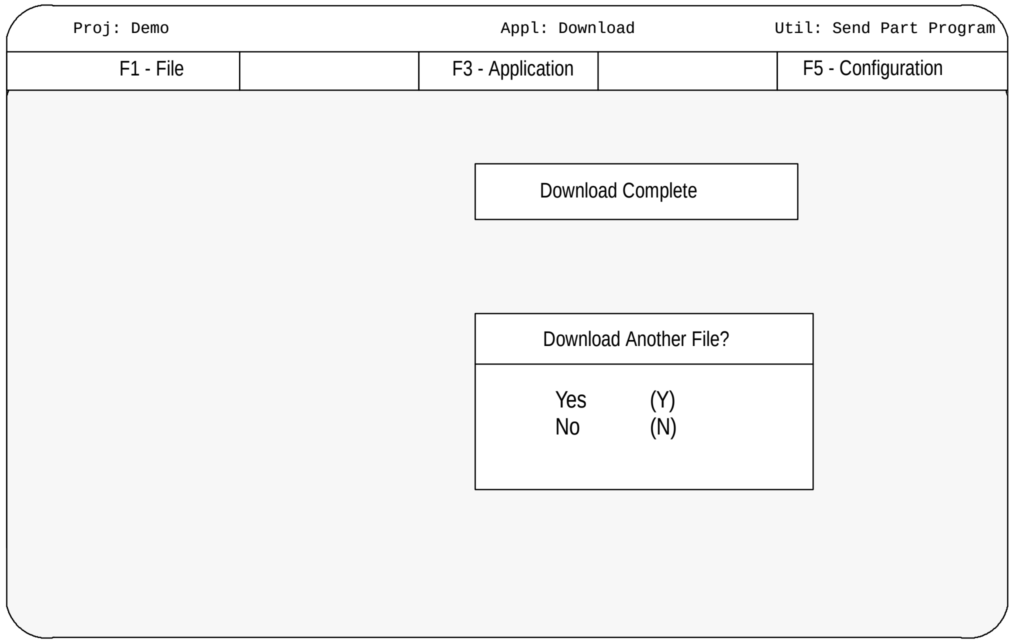
Important: If you enter a wildcard in place of a file name, the Abort option is repeated for each file that matches the wildcard. Pressing the [ESC] key quits the abort wildcard process.

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:



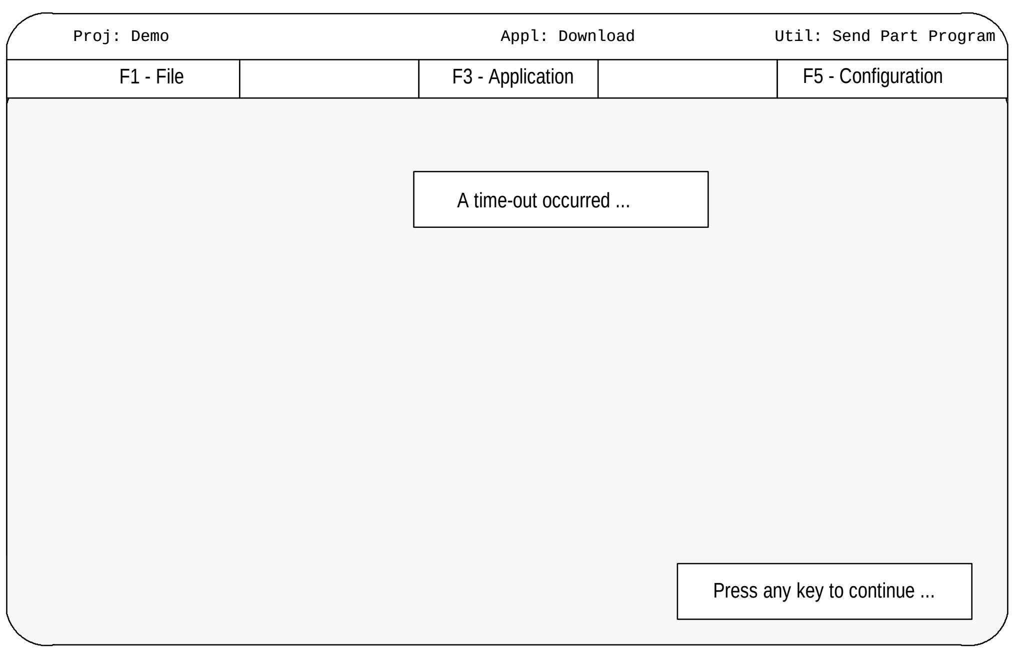
The percentage of the download process that has currently been completed is displayed on the screen. This value is updated continually throughout the download process.

When the download process is complete, the workstation displays this screen:



9. Select "Yes" or "No." Selecting "Yes" starts the download process again. Selecting "No" returns you to the main menu line of ODS.

If the workstation is unable to complete the download procedure in the allotted time frame, this screen is displayed:



Press any key to return to the ODS main menu.

6.5 Uploading 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.

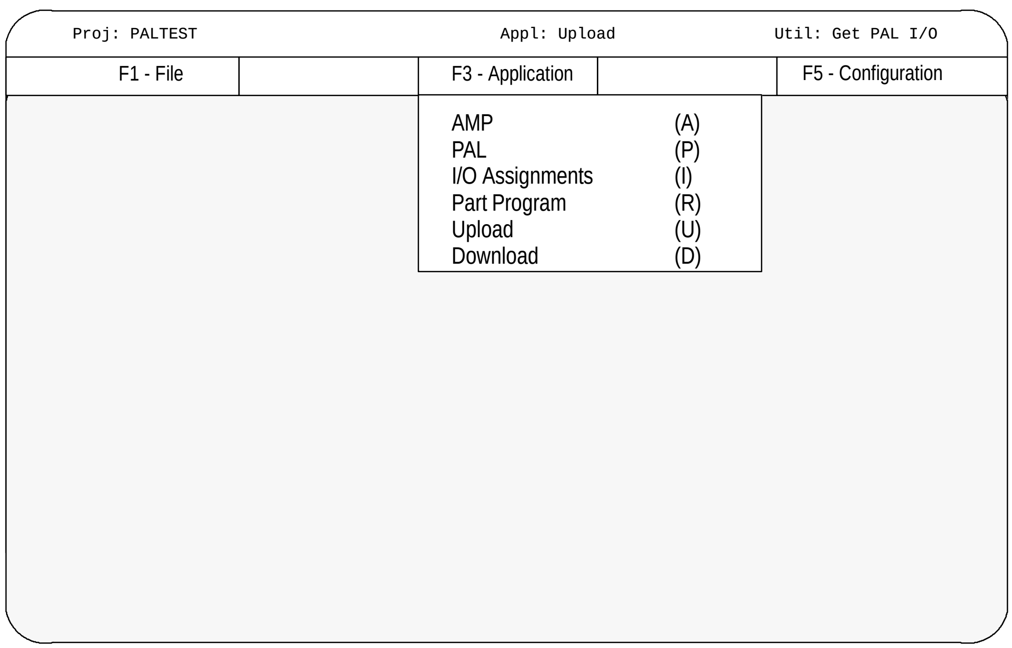
Important: Part programs in the protectable program directory may be encrypted when they are uploaded from the control to ODS. If a protected part program is encrypted, it can be opened through the Part Program application of ODS, but it is unreadable because it was encrypted during the upload. See page 5-42 for additional information on the protectable part program directory and program encryption.

1. Interface the workstation with the control. See page 6-5.

2. Return to the main menu line of ODS.

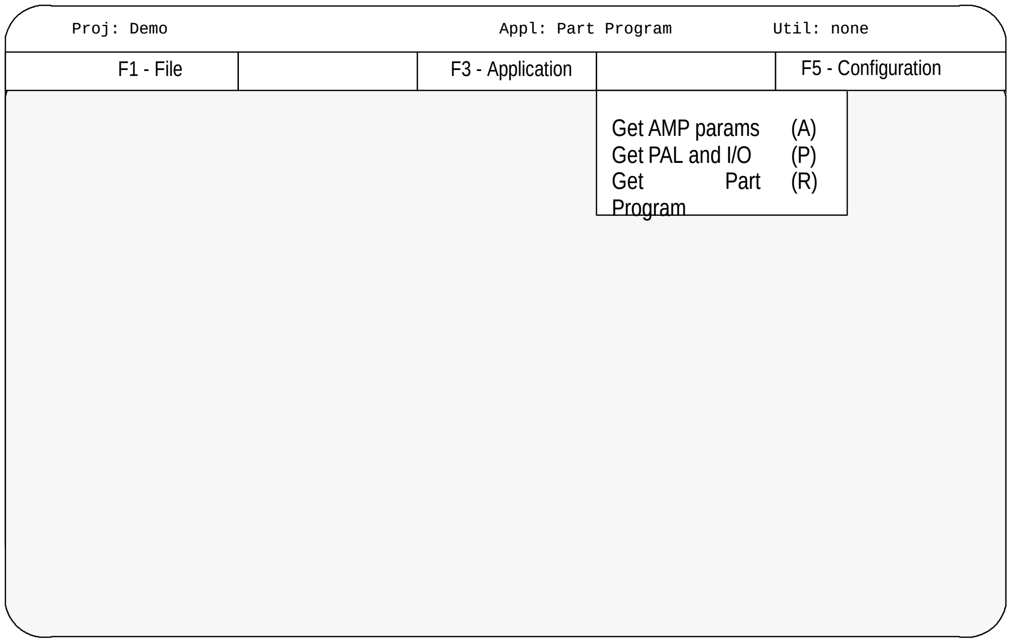
3. Press `[F3]` to pull down the Application menu.

   The workstation displays this screen:



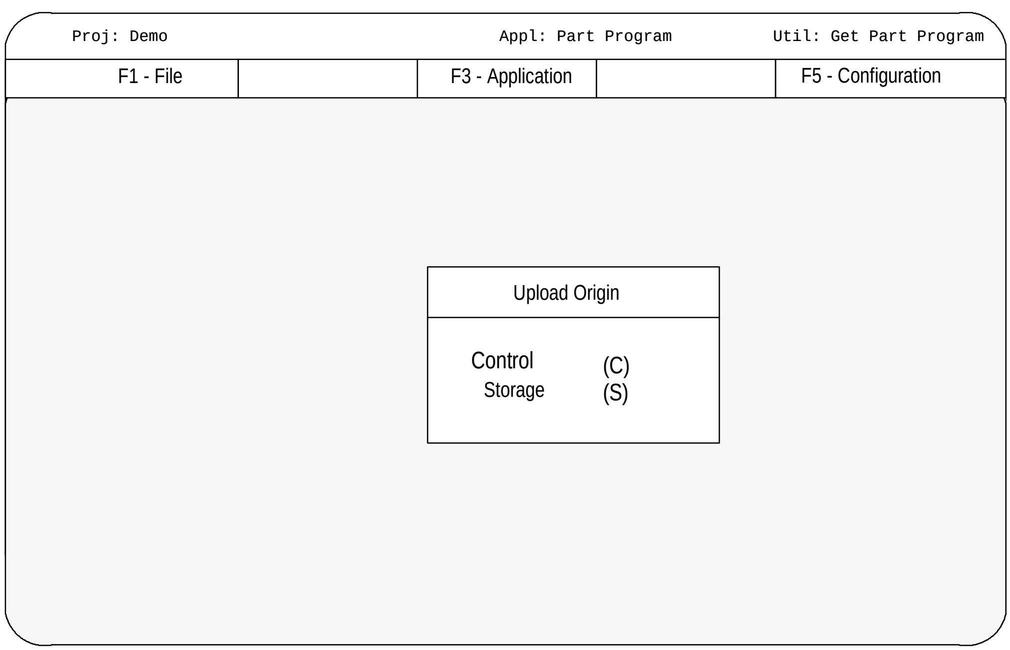
4. Use the arrow keys to highlight the Upload application then press `[ENTER]`, or press `[U]`.

5. Press [F4] to pull down the Utility menu:



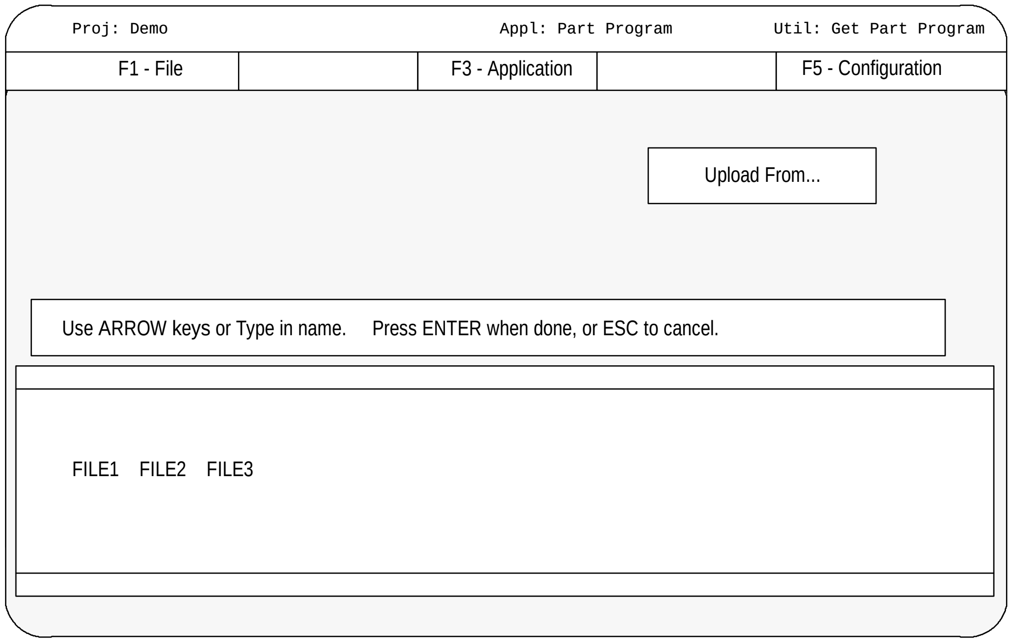
6. Use the arrow keys to highlight the Get Part Program option then press [ENTER], or press [R].

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.

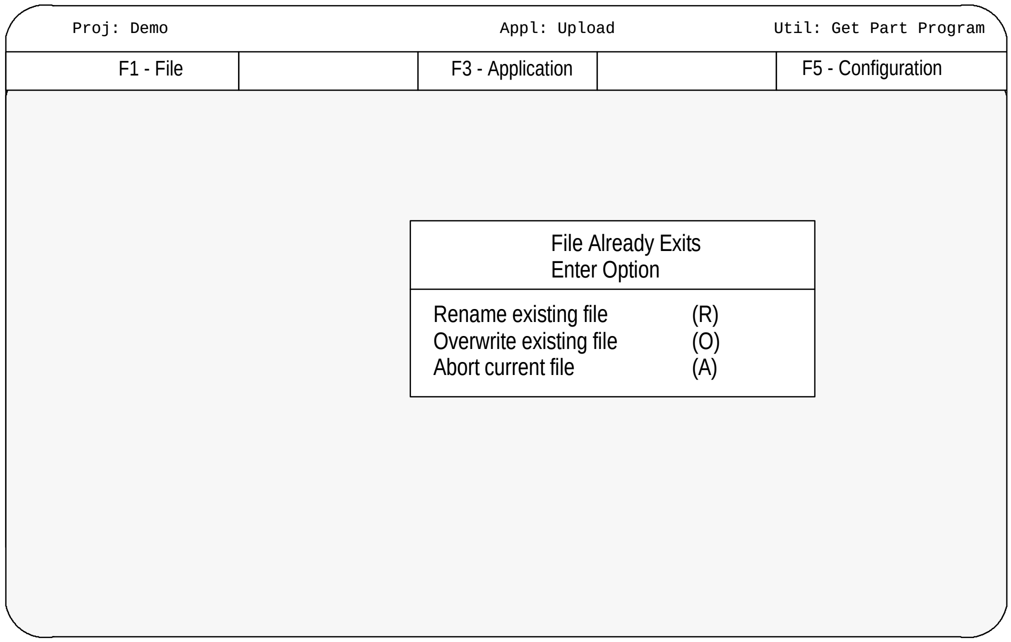
   The workstation displays the part program files that are stored on the control or storage device:



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]`.

   Important: You can upload more than one part program using wildcards (“*” or “?”) in place of all or part of a file name.
   See 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:



If you select the Rename option, the workstation renames the existing file, which has the same name as the file being uploaded, on the workstation.

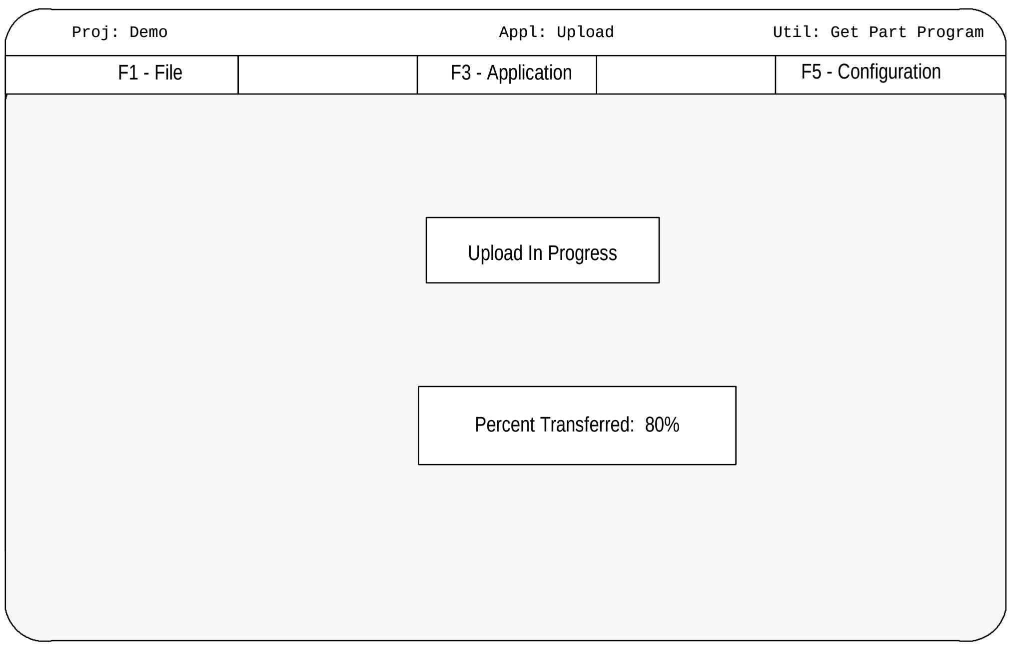
The workstation displays the part program files stored on the workstation:

9. Type in the new name for the existing part program file on the workstation.

If you select this option:	then:
overwrite	the part program file being uploaded overwrites the file having the same name on the workstation.
abort	the upload process is discontinued and the workstation prompts the programmer for additional files to upload.

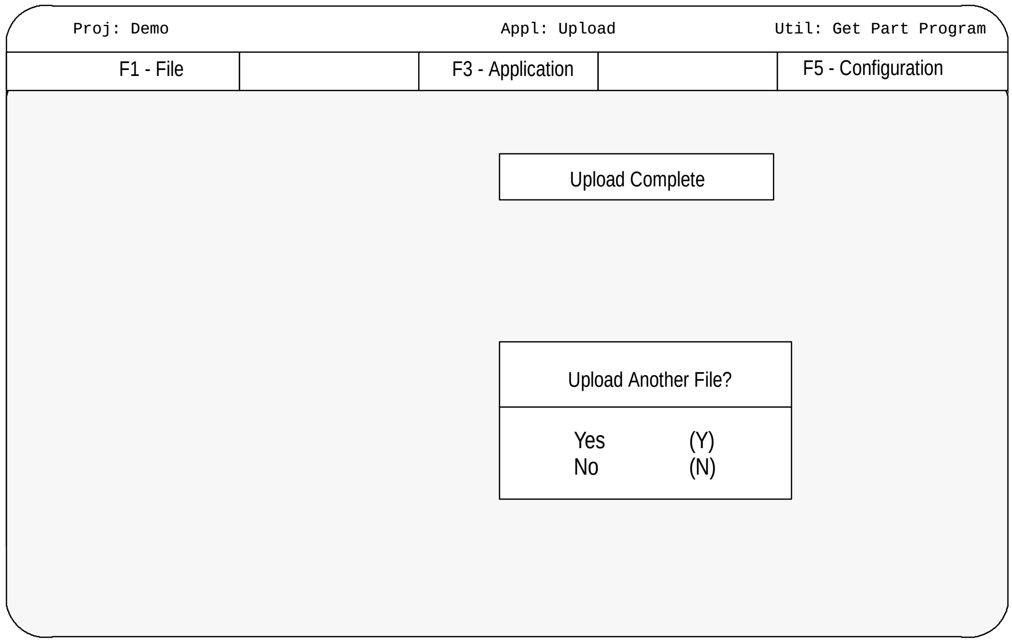
Important: If you enter a wildcard in place of a file name, the Abort option is repeated for each file that matches the wildcard. Pressing the [ESC] key quits the abort wildcard process.

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.

After the part program has been uploaded to the workstation, the workstation displays this screen:



Select “Yes” or “No.” If you select “Yes,” the workstation prompts you through the upload procedure again. If you select “No,” the workstation returns to the main menu line.

END OF CHAPTER

Chapter 7

Running a Program

7.0 Chapter Overview

This chapter describes how to test a part program and execute it in automatic mode. Major topics covered here include:

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 skips (not execute) all remaining motion commands 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 control skips the entire block regardless of the position of the block delete character. For details on the block delete feature, see page 10-11.

The block delete feature is activated through your system installer’s PAL program (see documentation prepared by your system installer), or by using 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 ignores M, second auxiliary functions (B codes), S, and T codes in the part program, except for M00, M01, M02, M30, M98, and M99.

The miscellaneous function lock feature is activated through your system installer’s PAL program (see documentation prepared by your system installer), or by using the {FRONT PANEL} softkey (chapter 2, page 2-11).

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 can 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 begins execution of that block. If it is not entered before the block begins, it is ignored and execution continues as normal.

Automatic execution stops 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 you enter a sequence stop number 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 results in the sequence stop function being cancelled.

If you call a subprogram or macro that also contains a sequence number that corresponds to the sequence stop number, program execution stops in the subprogram or macro at the corresponding sequence number.

To enter a sequence number to stop execution:

1. Press the {PRGRAM MANAGE} softkey. A program must have already been selected for automatic execution as described on page 7-6.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the {ACTIVE PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. Press the {SEQ STOP} softkey.

(softkey level 3)

↑	DE-ACT PRGRAM	SEARCH	MID ST PRGRAM	T PATH GRAPH	SEQ STOP	→

↑					TIME PARTS	→

4. Key in the sequence number where automatic operation in the part program is to stop, then press the [TRANSMIT] key.

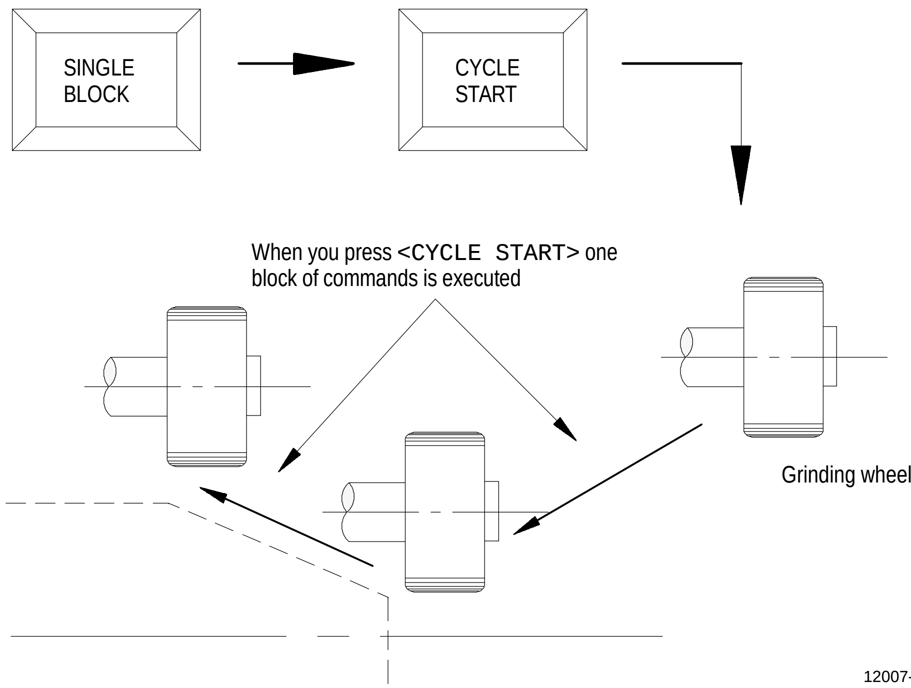
   The control stops automatic operation after it completes the commands in that block.

   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. The control executes one block of commands in the part program when in single block mode each time you press the <CYCLE START> button.

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.

ATTENTION: Single block execution is not possible during reciprocation.

If you press the <SINGLE BLOCK> button 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 currently executing block.

The <SINGLE BLOCK> button is a toggle switch. If you press it again while the single block function is active, the control cancels the function and the light inside the button turns off. The remaining program blocks can be executed normally by pressing the <CYCLE START> button.

7.2 Selecting a Part Program Input Device

Before selecting a part program, you must tell the control where this part program is currently residing. You have 3 options:

- the program can reside in the control's memory
- the program can reside on a peripheral device attached to port A such as a tape reader (see your system installer's documentation)
- the program can reside on a peripheral device attached to port B such as a tape reader (see your system installer's 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. See 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 `{PRGRAM MANAGE}` softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the `{INPUT DEVICE}` softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

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 described on page 7-6 for selecting a program.

7.3 Selecting a Program

To select a program for automatic execution, follow these steps:

Important: Consider the following when selecting a program:

- The control does not allow the selection of a program for execution if that program file is still open for editing. See page 5-15 on how to exit the edit mode.
- Your system installer can write PAL to allow some other method of part program selection. See the documentation prepared by your system installer for additional information.
- Before selecting a part program to activate, the input device must have been previously selected as described on page 7-5. The default condition is to select the part program out of control memory.
- If a program was previously activated and not deactivated, the control does not allow you to select a different part program. If you want to select a different part program, you must first deactivate the active program as described on page 7-9.

To select a program for automatic execution:

1. Press the {PRGRAM MANAGE} softkey. Figure 7.2 appears.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

Figure 7.2
Part Program Directory

```
SELECTED PROGRAM:

MAIN        DIRECTORY         PAGE    1  OF    1

  NAME      SIZE          COMMENT

TEST     AE   3.9
O12345        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 | EDIT   | RESTRT | DISPLY | COPY   |  →
    | PRGRAM | PRGRAM | PRGRAM | PRGRAM | PRGRAM |
```

Important: Figure 7.2 shows program TEST as active and being edited. Make sure no part program is currently active. If a part program is currently active, the control does not allow the selection of a different part program until the currently active one is deactivated (see page 7-9 on deactivating a part program).

If you see an:	To the right of the program name, it means that the program is:
A	currently active
E	currently being edited

2. Select the directory that contains the program to activate. You can select either the main directory or the protectable directory. The control displays the main directory as the default directory at power-up. You must select the protectable directory by using the `{CHANGE DIR}` softkey.

   Important: The `{CHANGE DIR}` softkey controls access to the protectable part program directory. The `{CHANGE DIR}` softkey is password protected. You must have the proper password to access this softkey. See page 5-42 for additional information on the protectable directory and the `{CHANGE DIR}` softkey.

3. Key in the name of the part program to activate. If the program is being selected from the control's memory, you can use the ↑ or ↓ cursor keys to select the program to activate from the directory screen.

   If you select the part program from a peripheral device (attached to port A or port B), you must manually key-in the part program name. Also make sure that 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.

4. 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: 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 are not available.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

Important: Before the control can execute the program, you must place the control in automatic mode.

7.4 De-Selecting a Part Program

To select a different part program for automatic execution, you must deactivate the part program that is currently active. To do this, follow these steps:

1. Press the {PRGRAM MANAGE} softkey. The control displays the program directory screen as shown in Figure 7.2.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

If you see an:	To the right of the program name, it means that the program is:
A	currently active
E	currently being edited

Figure 7.2 shows program TEST as active and being edited.

2. Press the {ACTIVE PRGRAM} softkey. The control displays the first few blocks of the currently active program.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. If the program selected is not the desired active program, press the {DE-ACT PRGRAM} softkey. The control deactivates the part program and returns to the directory screen.

(softkey level 3)

↑	DE-ACT PRGRAM	SEARCH	MID ST PRGRAM	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 can start a program at a chosen block and establish 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 page 7-6 for automatic execution.

The program search feature is not effective for subprograms and paramacros. You can search only blocks that are in the main program.

To perform a program search operation:

1. Press the {PRGRAM MANAGE} softkey. The program to search must have previously been selected for automatic execution as described on page 7-6.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the {ACTIVE PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. Press the {SEARCH} softkey.

(softkey level 3)

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

4. There are 6 different search options:

(softkey level 4)

↑	N SEARCH	O SEARCH	EOB SEARCH	SLEW	STRING SEARCH	→
↑	NEXT PRGRAM					

If you are searching for:	Press this softkey:
a sequence number	{N SEARCH}
an O word	{O SEARCH}
the end of each block	{EOB SEARCH}
the program one line at a time	{SLEW}
a specific character string	{STRING SEARCH}
the beginning of your next program	{NEXT PRGRAM} This softkey is available only if your input device has been configured as a tape reader. See page 7-5 on input device selection.

5. When using the SLEW or the EOB search options,

If you want to:	Press this softkey:
move to the next or previous blocks in the program	{FORWRD} or {REVRSE}
return to the top of the program (the beginning of the first block)	{TOP OF PRGRAM}
indicate that the desired block is found	{EXIT}

The selected block is the next block for automatic execution.

When using the N search, O search, or STRING search features, first key in the N number, O number, or character string to search for. After it has been keyed in, press the [TRANSMIT] key to start the search.

If you want to:	Press this softkey:
search for the entered value in the forward or reverse direction	{FORWRD} or {REVRSE}
return to the top of the program (the beginning of the first block)	{TOP OF PRGRAM}
indicate that the desired block is found	{EXIT}

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, page 9-1). 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.

ATTENTION: It may be necessary to position the grinding wheel at a location that allows this block to execute without damaging the workpiece or grinding wheel. This can be done through a manual operation or through MDI.

7.6 Search With Recall {MID ST PRGRAM}

Use the mid-start program feature to begin program execution from some block other than the first block of the program. This is done without the operator knowing what G, M, T, work coordinate offsets, etc. should be active for that block's execution or to re-execute all of the prior blocks to establish these conditions.

When you perform a search with recall, the control finds a character string or sequence number in a specific block for execution to begin from. 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).

Important: If wheel radius compensation is activated it will require a move to re-initialize. This can result in a different path than the program path if the previous blocks had actually been executed. See page 15-17 or 15-27 for details on entry moves into compensation.

As the control proceeds through the program searching for the entered character, it is setting active the necessary G, M, S, F, and T words. The control also establishes the work coordinate system that should be active for the execution of the selected block including all offsets and rotations to the work coordinate systems.

You can also use the program search with recall feature to search into any subprograms or paramacros that are contained in the main program. This is provided of course, that the searched block is in the path of normal program execution.

To perform a program search with recall, follow these steps:

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the {ACTIVE PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

Make sure that the program to search is the currently active program. If it is not, select it for automatic execution as described on page 7-6.

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
- 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,

   (softkey level 5)

↑	CONT	TOP OF PRGRAM	QUIT	EXIT		→

If you want to:	Press this softkey:
continue to search for the entered character string or sequence number	{CONT}
return to the first block in the program	{TOP OF PRGRAM}
end either the sequence number search or the character string search operations	{QUIT}

6. Press the {EXIT} or the {EXIT & MOVE} softkey once the program is at the desired location.

   {EXIT} - Use this softkey if the wheel 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 wheel 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 wheel 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 wheel 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.

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 wheel 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 ground. 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.

- QuickCheck (see page 7-18) -- This mode is a basic syntax checker for a part program. It checks that proper format and syntax have been followed. If you purchase QuickCheck with the graphic option, you can use it to determine if the proper paths are being generated (see page 8-24 for QuickCheck graphics). QuickCheck does not produce actual axis motion; however, it does perform offsets and coordinate system shifts.

- Axis Inhibit (see page 7-20) -- 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, see page 8-24 for details.

- Dry Run (see page 7-21) -- Dry run simply replaces all F word feedrates in a program with a special feedrate determined by your system installer in AMP. Graphics are available on the active program graphics screen, see page 8-24 for details.

- Part Production/Automatic (see page 7-22) -- 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, see page 8-24 for details.

All of the above modes of execution begin program execution when you press the `<CYCLE START>` button.

If you see this to the left of the block:	It means that the control:
*	is executing a part program block
@	has completed the execution of a 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.

You can interrupt axis inhibit, dry run, and automatic operation by using any of the operations listed below. You can resume execution at the interrupted location by pressing the `<CYCLE START>` button.

(1) Pressing `<CYCLE STOP>`

When you press the `<CYCLE STOP>` button, motion of the grinding wheel decelerates and stops, and the control stops automatic operation. If you press the `<CYCLE STOP>` button during a dwell, the control interrupts the dwell and stores any remaining time/revolutions for the dwell for later execution.

ATTENTION: Once axis reciprocation begins, it continues through program block execution until stopped by a G80, an end of program (M02, M30, M99), a change to manual mode or an emergency stop. This means pressing `<CYCLE STOP>` does not necessarily stop the reciprocating axis.

(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 your system installer.

ATTENTION: Once axis reciprocation begins, it continues through program block execution until stopped by a G80, an end of program (M02, M30, M99), a change to manual mode or an emergency stop. This means executing an M00 or an M01 in a part program does not necessarily stop the reciprocating axis.

(3) Entering a Sequence Stop Number

To interrupt execution at a specific block in the part program, use the sequence stop feature described on page 7-2. The control stops automatic operation after it completes the commands in the designated block.

(4) Feedhold Status

Your system installer may have written PAL to allow the activation of a feedhold status through the use of a button or switch. When activated, the control decelerates all moving axes to a feedrate of zero until the feedhold status is deactivated. For details on using feedhold, see the documentation provided by your system installer.

7.7.1 {QUICK CHECK}

QuickCheck is a basic syntax checker for a part program. It checks that proper format and syntax have been followed during programming. No actual programmed motion is produced in QuickCheck mode. The QuickCheck feature is also available with an optional graphics feature.

ATTENTION: It is your system installers responsibility to insure that no axis motion is generated from the PAL axis mover feature in the QuickCheck mode. Also any jog moves that can be performed on your system in automatic mode (features such as Jog Retract, Manual Gap Elimination, or Jog on the Fly) can produce axis motion in QuickCheck mode.

Modal features that are already active (such as G81 reciprocation, spindle direction and speed, and modal M codes that operate coolants or other peripherals) will remain in their current state when the QuickCheck mode is entered. For example if you start the spindle at 1000 RPM in the clockwise direction it continues to run at that RPM in that direction regardless of any commands executed in QuickCheck mode.

If you want to use the graphics feature, see page for QuickCheck with graphics. To use the QuickCheck feature as described below without graphics, the graphics option must be disabled.

To use the QuickCheck feature:

1. Select a program to check as described on page 7-6 and return to softkey level 1.

2. Press the {QUICK CHECK} softkey.

(softkey level 1)

PRGRAM MANAGE	OFFSET	MACRO PARAM	QUICK CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

When you press the {QUICK CHECK} softkey, the QuickCheck appears in reverse video and prompts the graphics option.

3. Press the <CYCLE START> button. The control executes program blocks as fast as it can process.

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.

If the control finds no errors during QuickCheck, the program screen displays the message "COMPLETED WITH NO ERRORS." The control then automatically resets the program to the first block.

To disable QuickCheck:	Press:
without the graphics option	the {QUICK CHECK} softkey again
with the graphics option	the {QUICK CHECK} softkey followed by the {STOP CHECK} softkey

ATTENTION: 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 paths of any inhibited axis are still drawn on the graphics screen.

The control executes the program in approximately the same amount of time as it would be in automatic mode, even though some or all axes may not physically move. You can use the axis inhibit feature in conjunction with dry run.

ATTENTION: 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 page 7-2).

You can activate axis inhibit to inhibit motion of any or all of the axes, including jogging moves, depending on the configuration determined by your system installer. This includes jog 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.

ATTENTION: Axes not selected for axis inhibit move as they would if the program were executed in automatic mode.

You can activate the axis inhibit feature by using a switch installed by your system installer (see documentation provided by your system installer) or by using the {FRONT PANEL} softkey (see page 2-11). The control must be in cycle stop or E-Stop to activate or deactivate the axis inhibit feature. The control ignores any attempt to activate or deactivate the feature during program execution or when in cycle suspend or feedhold states. The control also ignores attempts to activate the axis inhibit feature during jogging.

You can also inhibit spindle motion 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.

Press the <CYCLE START> button to start program execution with the axis inhibit feature provided a program has been selected for execution (see page 7-6 for selecting a program).

You can stop program execution with axis inhibit at any time by using any of the methods described for normal program execution 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 grinder 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 grinding feedrate. Jogging moves and moves that are programmed using rapid traverse (G00) are not effected by dry run.

You can use the axis inhibit feature in conjunction with a dry run.

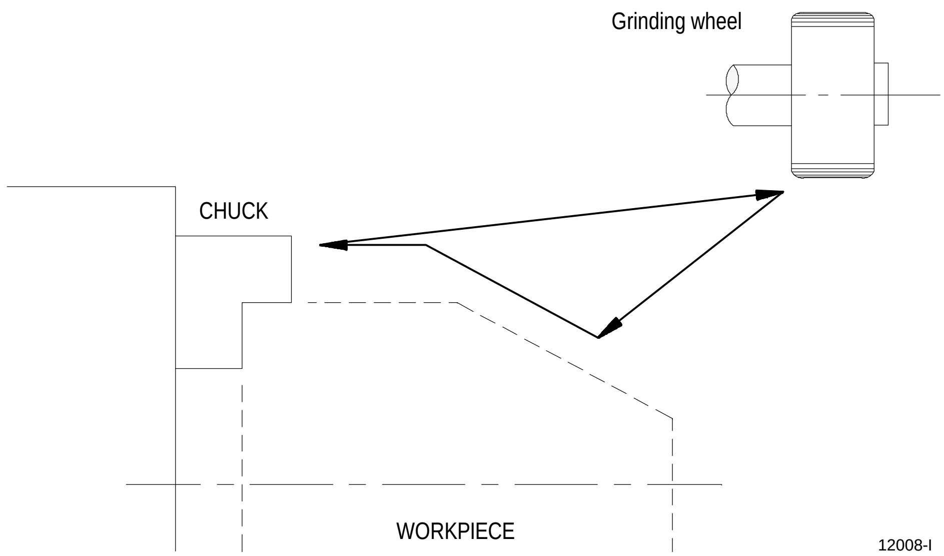
If you use the external decel feature simultaneously with the dry run feature, the control uses the feedrates that are assigned to external decel feature and ignores the dry run request.

You can use the <FEEDRATE OVERRIDE> switch to modify the grinding feedrate. Your system installer determines in AMP if rapid feedrates are overrides by the <RAPID FEEDRATE OVERRIDE> switch or the <FEEDRATE OVERRIDE> switch during dry run.

ATTENTION: 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 (see page 7-2).

ATTENTION: Your system installer can 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 can result in damage to the grinder, part, or injury to the operator.

Figure 7.3
Dry Run



You can activate the dry run feature by using a switch installed by your system installer (see documentation provided by your system installer), or by using the {FRONT PANEL} softkey (see page 2-11).

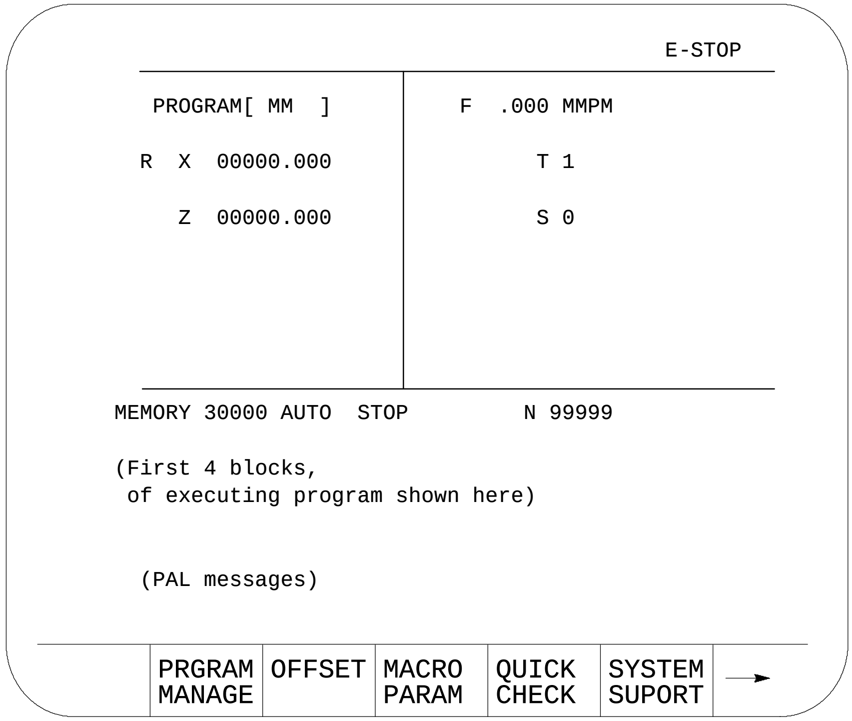
7.7.4 Part Production/Automatic Mode

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 described on page 8-24.

To select the automatic mode, place the <MODE SELECT> switch (on the MTB panel) in the AUTO position. If not equipped with a mode select switch, use the {FRONT PANEL} softkey.

AUTO appears on the main menu screen when the control is in automatic mode as shown in Figure 7.4.

Figure 7.4
Main Menu Screen in AUTO Mode

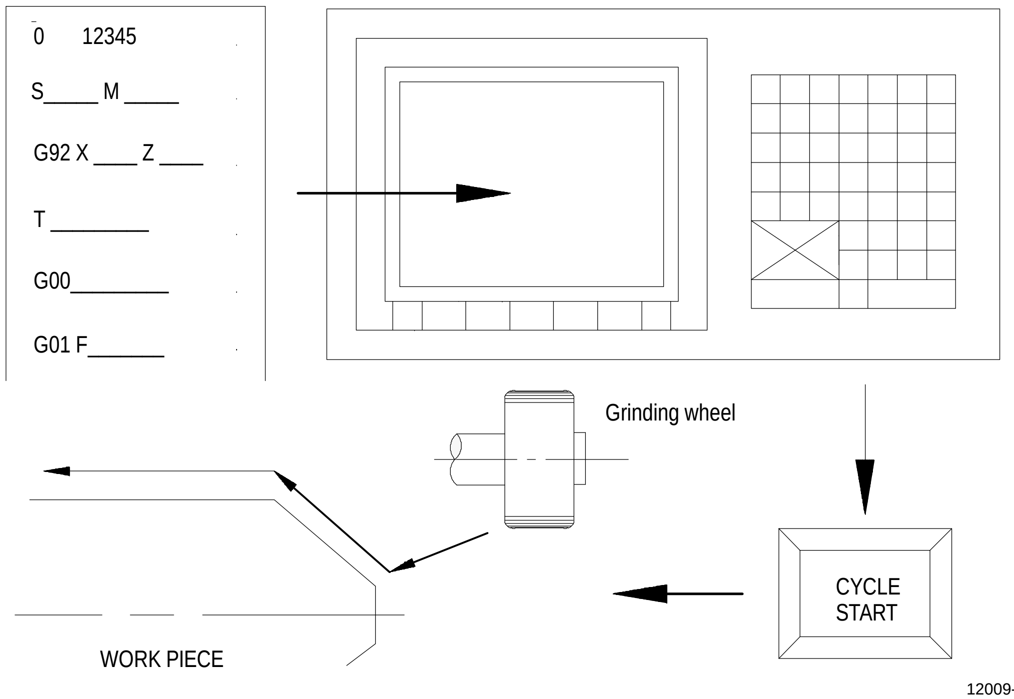


In automatic mode, the control manages grinder operations according to the commands in a part program.

- CYCLE START -- begins part program execution
- CYCLE STOP -- stops part program execution

ATTENTION: Always test a program prior to automatic operation. Always verify that the work space 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.

You can stop execution at any time by using any of the methods described on page 7-2 or by pressing the `<EMERGENCY STOP>` button.

7.8 Interrupted Program Recover {RESTRT PRGRAM}

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 enables the control to return to the block in the program that execution was interrupted at. This is done without the need for the operator to remember the block that was interrupted or the need to re-execute all of the prior blocks to reestablish modal operating conditions.

When you preform a program recover, the control automatically returns the program to the beginning of the block that was interrupted. In the case of power fail, the control even reselects the program as active.

ATTENTION: When you perform a program recover, 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 wheel is still somewhere along the path of the block that program execution was interrupted while grinding.
- In incremental or circular mode, however, if the grinding wheel is still located at the point that program execution was interrupted, a restart can damage the part.
- If you perform a program recover operation in incremental mode, it is important that the grinding wheel be at the location that the interrupted program block began, not the location that the program was interrupted at.
- If wheel radius compensation was active, it will require a move to re-initialize. This results in a path different than the originally programmed path. See page 15-17 or 15-27 for details on entry moves into compensation.

As the control proceeds through the program searching for the entered character, it is setting active the necessary G, M, S, T, and F words. The control also establishes the work coordinate system that should be active for the execution of the selected block including all offsets and rotations to the work coordinate systems.

You can also use this feature to search into any subprogram or paramacro that are contained in the main program.

To perform a program restore operation after automatic program execution has been interrupted, follow these steps:

1. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH 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)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

3. To automatically search for the block in the current program that was interrupted, press the {EXEC} softkey.

   The control locates 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 is aborted.

4. Press the {EXIT} softkey if the block selected is the block to begin program execution from. If it is not the block you want, you must disable the program or perform a search with memory operation to locate the desired block manually.

(softkey level 3)

↑	EXEC	QUIT	EXIT			→

When you press the <CYCLE START> button, the control resumes program execution from the block selected with the program restart feature.

7.9 Jog Retract

Use the jog retract feature to inspect, dress, or change the grinding wheel during automatic program execution. It allows the grinding wheel 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 can vary from 0 to 15 as determined by an AMP parameter set by your system installer. The control returns the wheel along the jog retract path at a feedrate specified in AMP.

Important: If you use the same axis in succession during a jog retract operation, the control assumes that only one jog retract move has been executed on that axis.

You can perform only simple single axes jog moves during the jog retract function. You are not allowed to perform multiple axis jogs, arbitrary angle jogs and jogging offset.

You can change wheel offsets at any time during jog retract (see page 3-4 for wheel offset). The control does not make these offsets active until the execution of the first block after the wheel has been returned from jog retract.

ATTENTION: If you deactivated the jog retract function during its execution (performing a control reset, E-Stop, etc.), attempting to return the wheel by pressing cycle start can cause an undesired return path. The wheel returns to the start point of jog retract along a linear path. This is most likely not the retracted path. To avoid possible part or wheel damage, we recommend the grinding wheel 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> button or the <SINGLE BLOCK> 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 grinding wheel from the workpiece using either continuous jog, incremental jog, or HPG jog operations (see page 4-1 for jogging information).

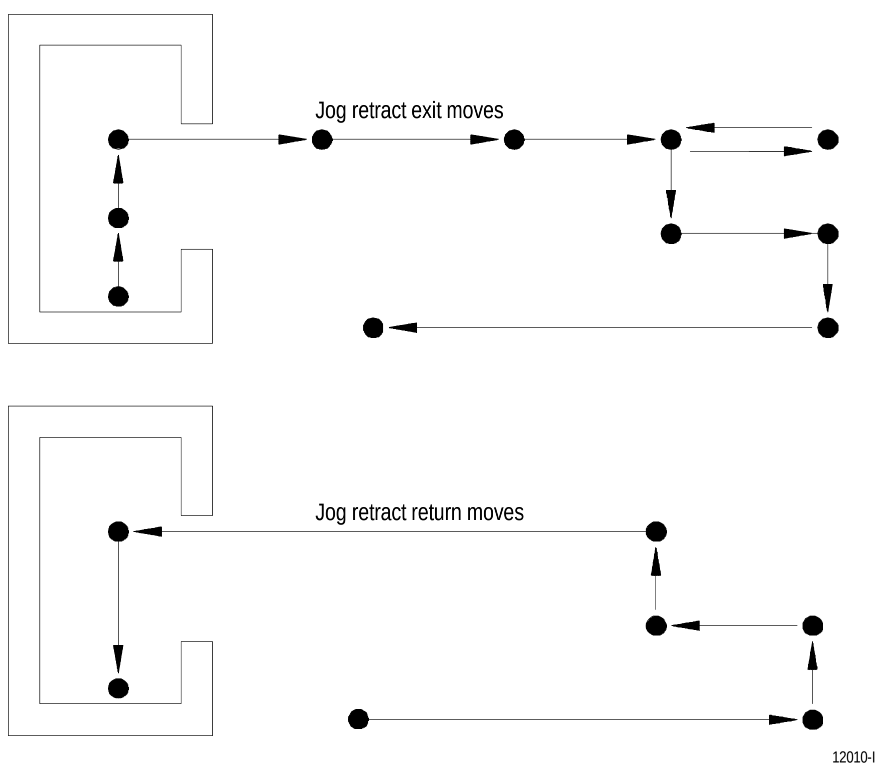
4. Inspect and change the wheel or wheel offset as desired. Details on how to do this are on page 3-4.

5. After completing the desired inspection, dressing, or wheel change, press the <CYCLE START> button. Any wheel offset changes you have made become active when the cycle start is requested. The wheel 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 offset values).

   You can press <CYCLE STOP> during the wheel automatic return to the jog retract start position. When this is done, the wheel can be retracted from this point using jog moves and the control adds these moves to any remaining jog retract steps that have not yet been returned.

6. Once the grinding wheel is fully returned from a jog retract operation, the control continues on in the part program unless in single block mode. If in single block mode, the control goes 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 wheel. This is because the jog retract feature records consecutive jog moves on the same axis as one move.

ATTENTION: If the number of jog retract moves performed exceeds the maximum allowed number set in AMP, the control moves the grinding wheel directly from the final point of jog retract to the last remembered jog retract point along a straight line when you press `<CYCLE START>`. Then the wheel is returned in the normal jog retract fashion.

Figure 7.7
Jog Retract Moves that Exceed the Maximum Allowed in AMP

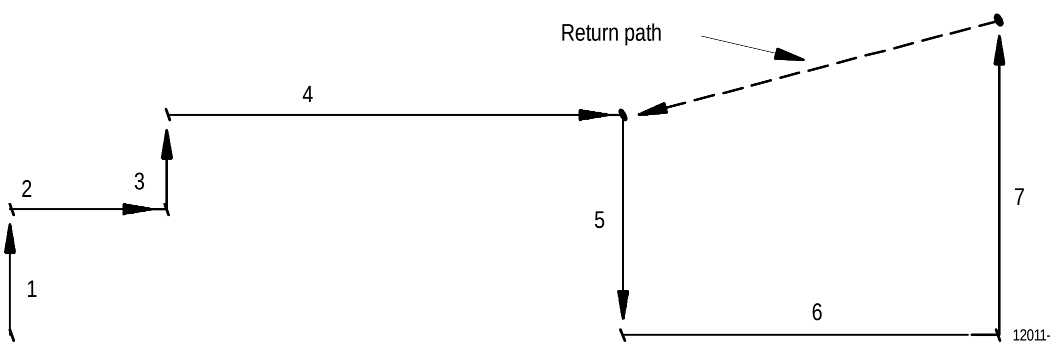


Figure 7.7 emphasizes the possible problems that can result from exceeding the maximum allowed jog retract moves. In this example, the number of allowed moves set in AMP is 4.

When you press the cycle start button 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 wheel.

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 your system installer in AMP, and can vary from 1 to 15.

Important: For maximum control efficiency when executing programs, we recommend that the maximum number of allowable block retraces is set as small as possible for the current grinder application. This is because the number of allowable block retraces directly affects the control's block look-ahead operation.

You can only enable this function when the control is in cycle stop or cycle suspend state and the control ignores this feature if it has already executed an M02 or M30 end of program.

To perform a block retrace operation:

1. Press the <CYCLE STOP> button or the <SINGLE BLOCK> button to stop program execution.

2. Press the <BLOCK RETRACE> button.

After you press the <BLOCK RETRACE> button, the control retraces the block that was being executed when the cycle stop occurred or retrace the block just completed if you press the single block button, provided that the block is a legal block for retrace.

While the block retrace function is active, the light in the <BLOCK RETRACE> button is on. The block that was shown as active when the block retrace was activated stills appear as the currently active block in the program display area during the entire use of the block retrace function.

Important: If you use the <CYCLE STOP> button 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 mm and the axis has moved 12 mm when you press the <CYCLE STOP> button, a block retrace reverses the axis direction 12 mm.

All retraced blocks are executed at the feedrate programmed for that block. You can, however, modify the feedrate by the use of the <FEEDRATE OVERRIDE> switch. See page 12-58.

Press the <CYCLE START> button at any time during a block retrace to return the grinding wheel to normal forward execution. Program execution returns to the normal forward direction from the currently retraced block. The control executes the retraced blocks in normal order until the wheel 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 halts execution when the return from block retract is complete.

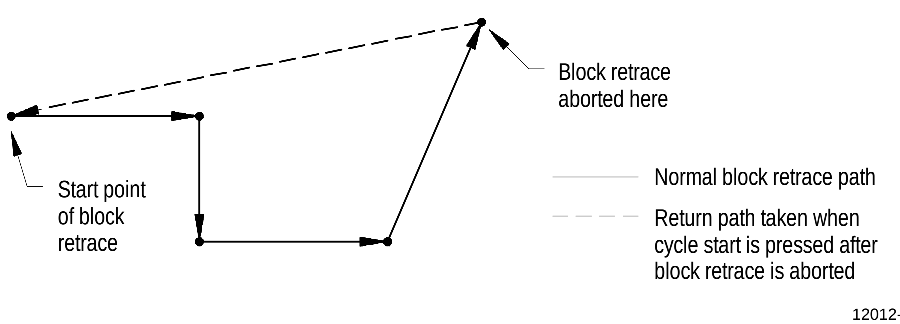
While block retrace is active, the control disables all jog features with the exception of <JOG RETRACT>. See page 7-27 for details on Jog Retract. MDI is not available to insert blocks during a block retrace operation.

The block retrace function is unable to retrace any of the following blocks and an attempt to do so results in an error message:

- Thread grinding
- Inch/Metric changes (unit conversion)
- 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 retraceable blocks as determined in AMP
- Certain Paramacro Parameter Assignments
- Interrupt programs or interrupt dressing operations

ATTENTION: If the block retrace function is deactivated during its execution (performing a control reset, E-Stop, etc.), attempting to return the axis by pressing cycle start can cause an undesired return path as shown in Figure 7.8. The axis returns to the start point of the block retrace along a linear path that is usually not the same as the retracted path. To avoid possible part or wheel damage, jog the grinding wheel 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

Chapter 8

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
`{PRGRAM}`	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 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, small, 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 then 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.

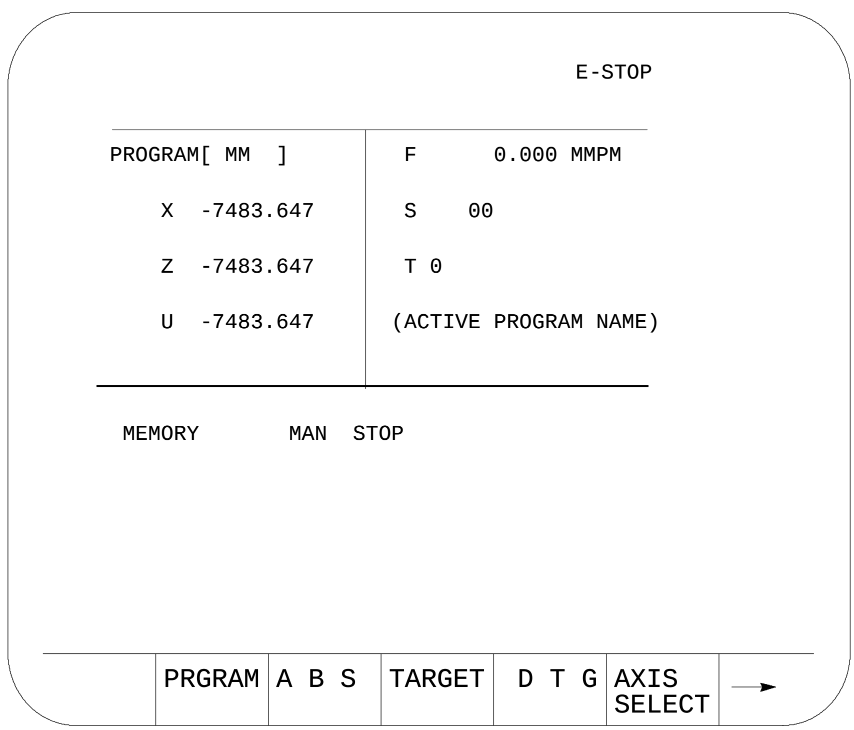
3. To return to softkey level 1, press the [DISP SELECT] key again. The most recently selected data position screen will remain in effect for softkey level 1 until either power is turned off or a different position display screen is selected. The default screen selected at power up is the regular size program display.

The following figures show the axis position data display that will result when the corresponding softkey is pressed.

{PRGRAM}

Axis position in the current work coordinate system displayed in normal size characters.

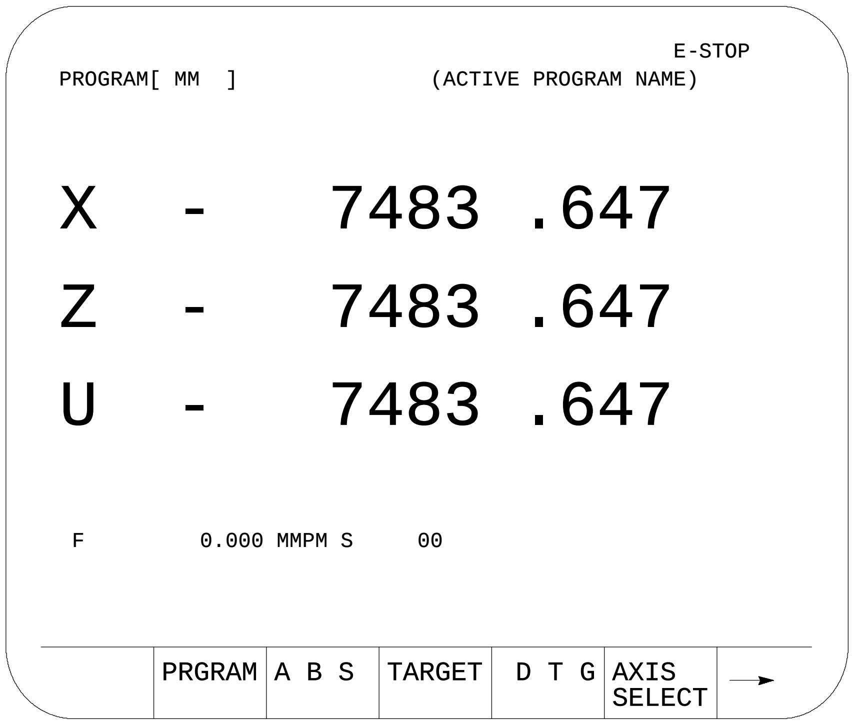
Figure 8.1
Result After Pressing {PRGRAM} Softkey



{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 axis are AMPed in the system, or more than 8 axis in the process for dual process systems).

Figure 8.3
Results After Pressing {PRGRAM} (Small Display) Softkey

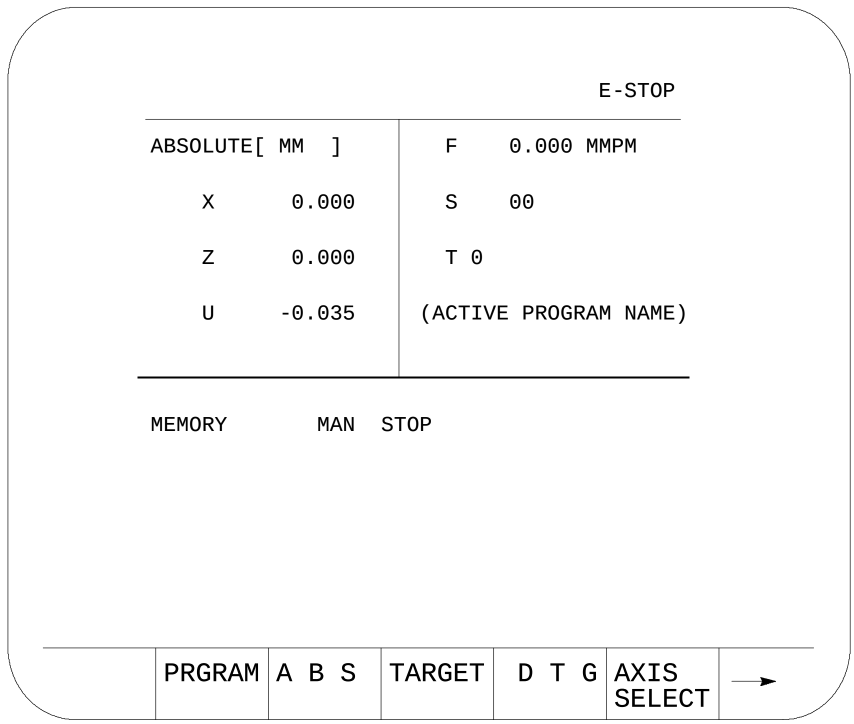
```
PROGRAM[ MM  ]

    X  -9999.647
    Y  -3333.647
    Z  -1111.647
    U  -2222.647
    V  -2222.647
    W  -2222.647
    A  -2222.647
    B  -2222.647
    C  -2222.647
    $X -2222.647
    $Y -2222.647
    $Z -2222.647


  F      0.000 MMPM            S    00
```

	PRGRAM	A B S	TARGET	D T G	AXIS SELECT	→

{ABS}

The axis position data in the machine coordinate system.

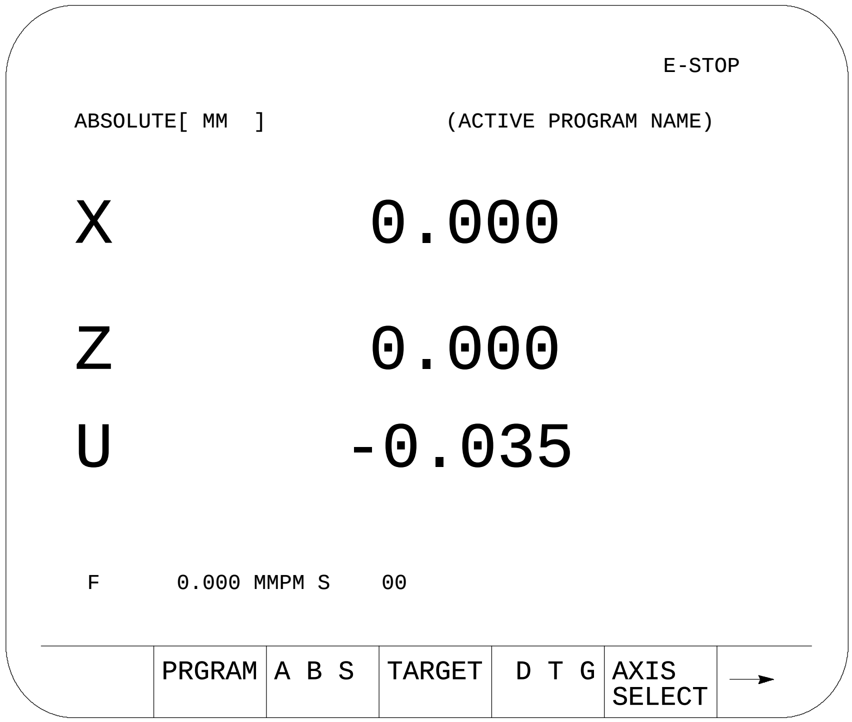
Figure 8.4
Results After Pressing {ABS} Softkey



{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 axis are AMPed in the system, or more than 8 axis in the process for dual process systems).

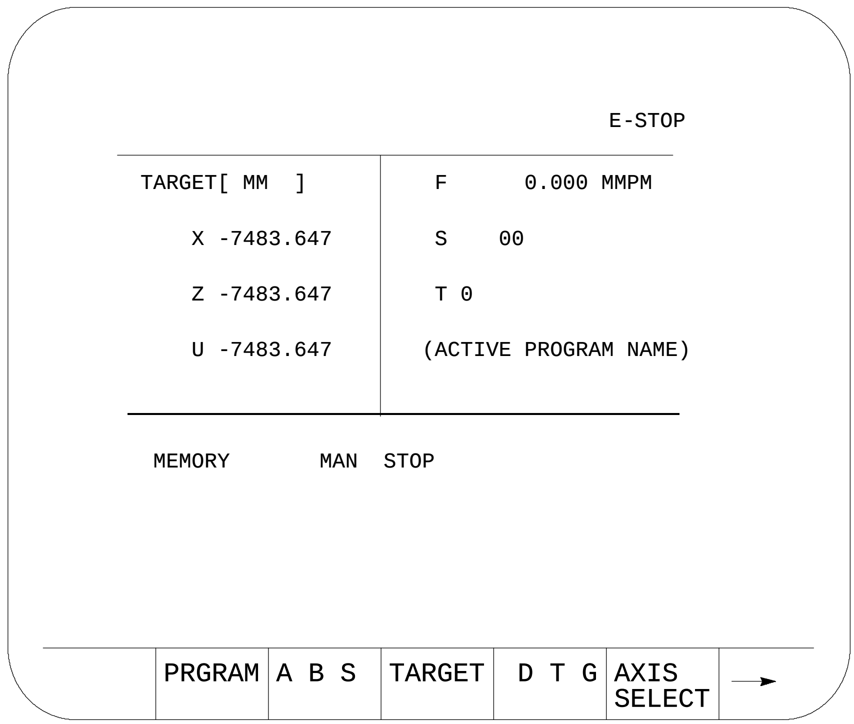
Figure 8.6
Results After Pressing {ABS} (Small Display) Softkey

```
ABSOLUTE  [ MM  ]

    X  -9999.647
    Y  -3333.647
    Z  -1111.647
    U  -2222.647
    V  -2222.647
    W  -2222.647
    A  -2222.647
    B  -2222.647
    C  -2222.647
    $X -2222.647
    $Y -2222.647
    $Z -2222.647


  F      0.000 MMPM           S    00

         PRGRAM | A B S | TARGET | D T G | AXIS SELECT | →
```

{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.

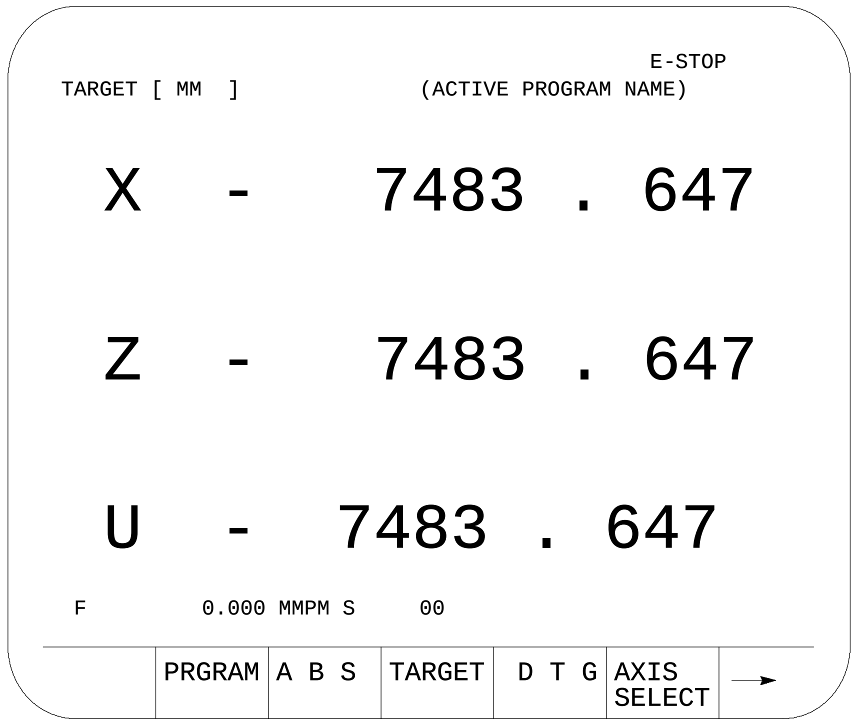
Figure 8.7
Results After Pressing {TARGET} Softkey



{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 axis are AMPed in the system, or more than 8 axis in the process for dual process systems).

Figure 8.9
Results After Pressing {TARGET} (Small Display) Softkey

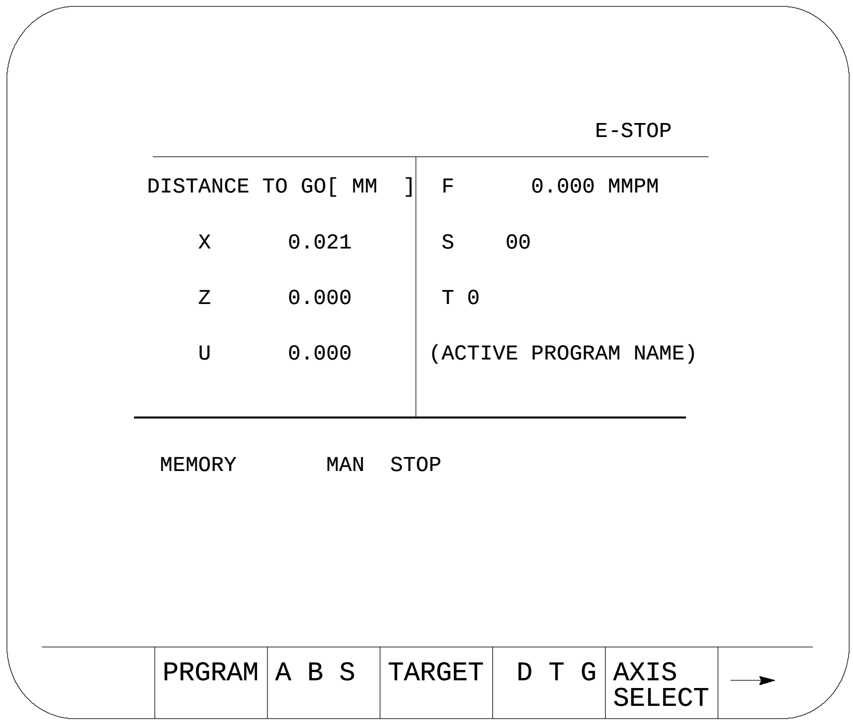
```
TARGET  [ MM  ]

    X  -9999.647
    Y  -3333.647
    Z  -1111.647
    U  -2222.647
    V  -2222.647
    W  -2222.647
    A  -2222.647
    B  -2222.647
    C  -2222.647
    $X -2222.647
    $Y -2222.647
    $Z -2222.647


  F      0.000 MMPM           S    00
```

	PRGRAM	A B S	TARGET	D T G	AXIS SELECT	→

{DTG}

The distance from the current position to the command end point, of the commanded axis in normal size characters.

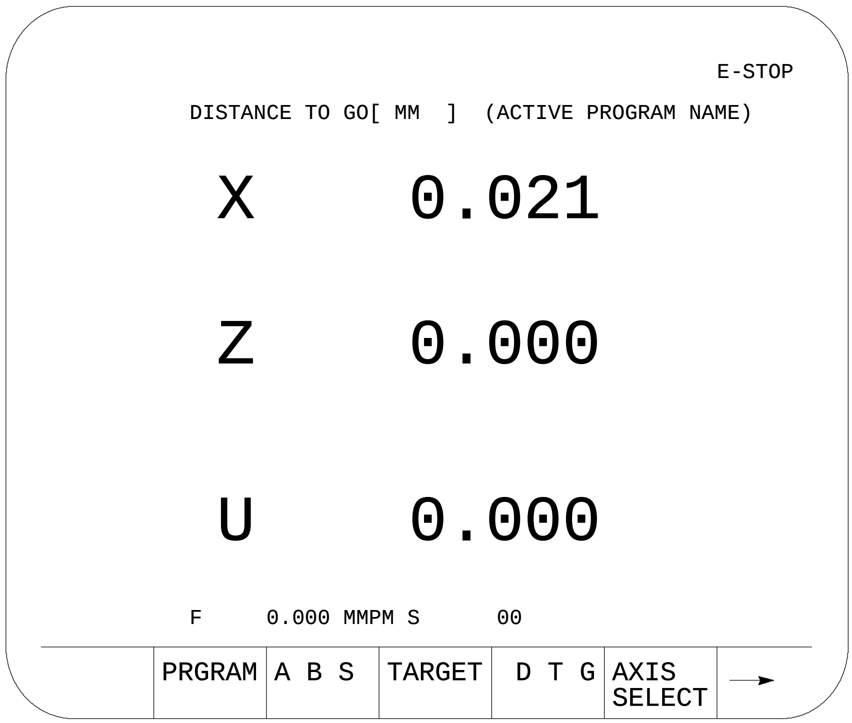
Figure 8.10
Results After Pressing {DTG} Softkey



{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 axis are AMPed in the system, or more than 8 axis in the process for dual process systems).

Figure 8.12
Results After Pressing {DTG} (Small Display) Softkey

```
Distance to Go  [ MM  ]

    X  0000.000
    Y  0000.000
    Z  0000.000
    U  0000.000
    V  0000.000
    W  0000.000
    A  0000.000
    B  0000.000
    C  0000.000
    $X 0000.000
    $Y 0000.000
    $Z 0000.000


  F      0.000 MMPM            S    00
```

	PRGRAM	A B S	TARGET	D T G	AXIS SELECT	→

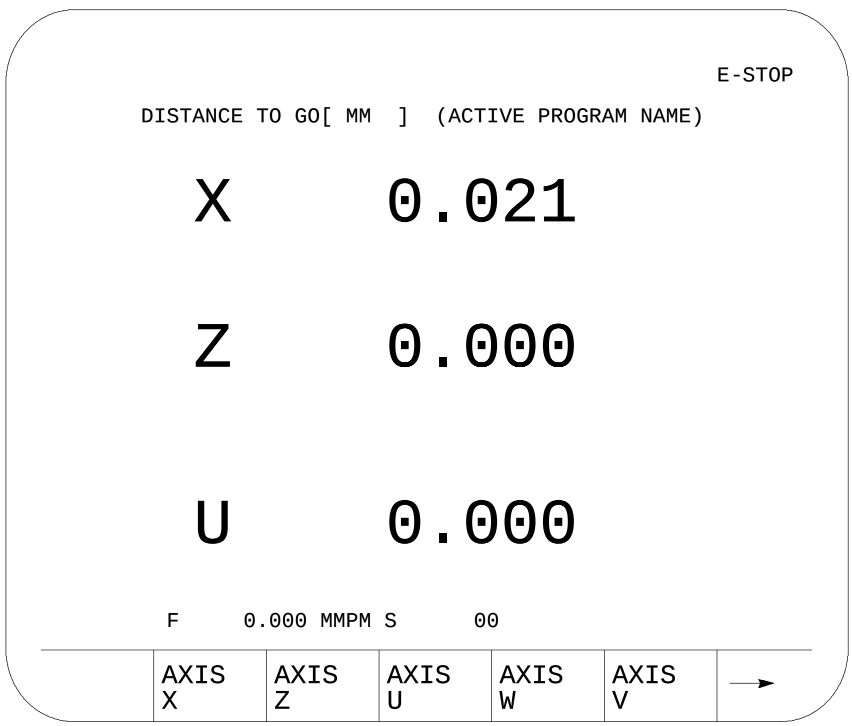
{AXIS SELECT}

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).

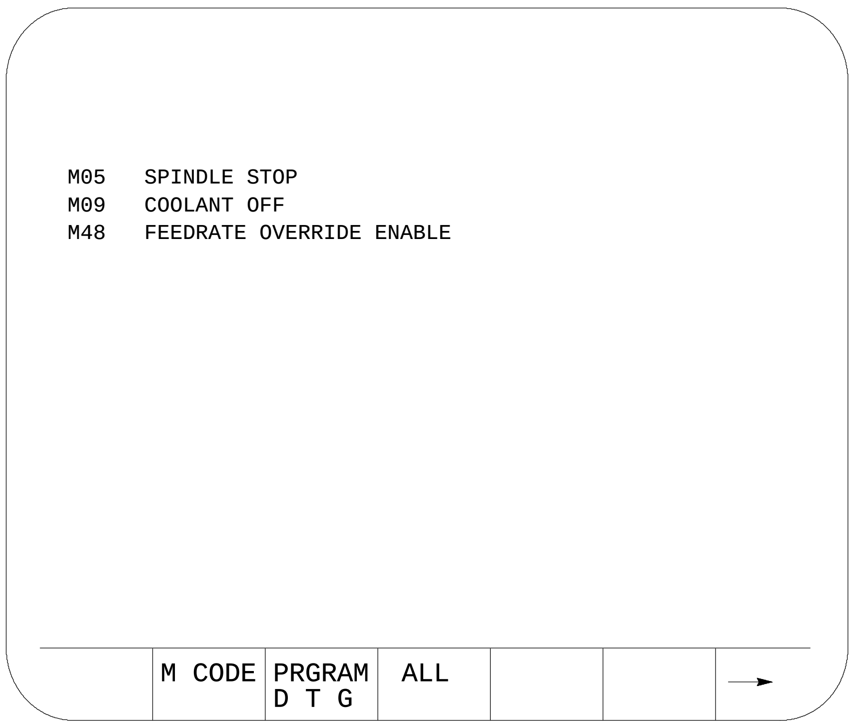
Figure 8.13
Results After Pressing {AXIS SELECT} Softkey



{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 programmers responsibility to make sure proper machine action takes place when the M code is programmed.

Figure 8.14
Result After Pressing {M CODE} Softkey



{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

```
                                          E-STOP
   PROGRAM          DISTANCE TO GO    [ MM ]

   X  - 7483.647     X      0.031

   Y  - 7483.647     Y      0.000

   Z  - 7483.647     Z      0.000

      F       0.000 MMPM     S    0

         | M CODE | PRGRAM | ALL |     |     |  →
         |        | D T G  |     |     |     |
```

{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
    $X -2222.647            $X 0000.000
    $Y -2222.647            $Y 0000.000
    $Z -2222.647            $Z 0000.000


  F      0.000 MMPM            S    00

        | M CODE | PRGRAM |  ALL  |       |       |  -->
        |        | D T G  |       |       |       |
```

{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 MMPM      S    0
```

	M CODE	PRGRAM D T G	ALL		G CODE 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

{G CODE STATUS}

The currently active G-codes are displayed.

Figure 8.18
Results After Pressing {G CODE} Softkey

```
PROGRAM STATUS
                              PAGE 2 OF 2
G50.1  MIRROR IMAGE CONTROL

G64    CUTTING MODE
G67    MACRO CALL CANCEL

G70    INCH PROGRAMMING
G80    CANCEL OR END FIXED CYCLE
G90    ABSOLUTE
G94    FEED/MIN
G97    CSS PROGRAMMING OFF
G98    FIXED CYCLE INITIAL LEVEL RETURN

                                      G CODE
                                      STATUS  →
```

```
PROGRAM STATUS
                              PAGE 1 OF 2
G01    LINEAR INTERPOLATION
G07    RADIUS PROGRAMMING
G12.1  PRIMARY SPINDLE CONTROLLING
G14    SCALING CANCEL
G15    POLAR/VIRTUAL CANCEL
G18    PLANE ZX
G23    ZONE 2 AND 3 OFF
G36    SHORT BLOCK ACC/DEC CHECK ON
G39    G41/42 LINEAR TRANSITION
G40    TOOL TIP RADIUS COMP CANCEL

       M CODE  PRGRAM  ALL          G CODE
       STATUS  D T G                STATUS  →
```

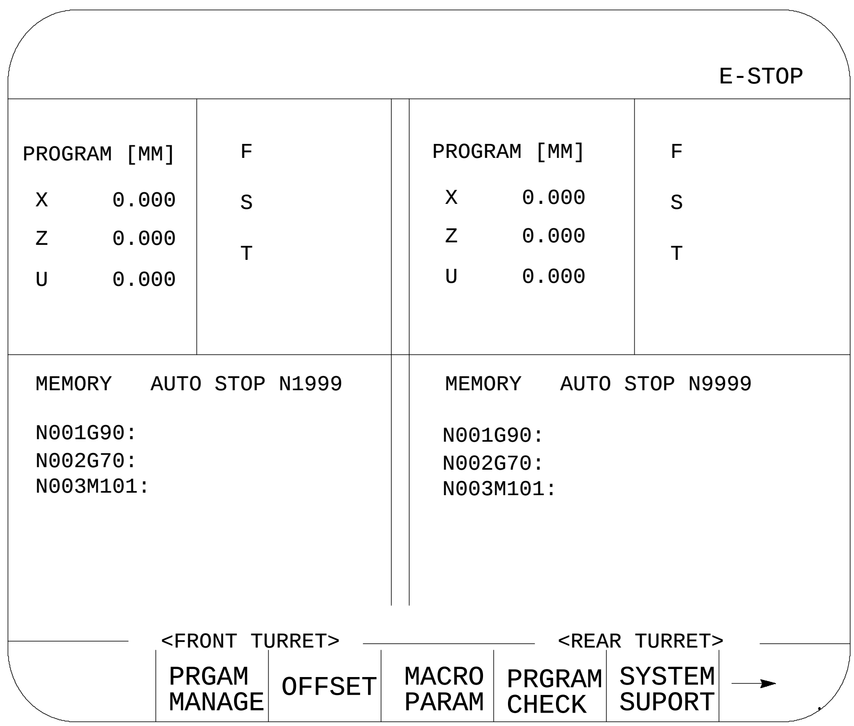
{SPLIT ON/OFF}

The split screen softkey is only available if your system installer has purchased the dual-process option.

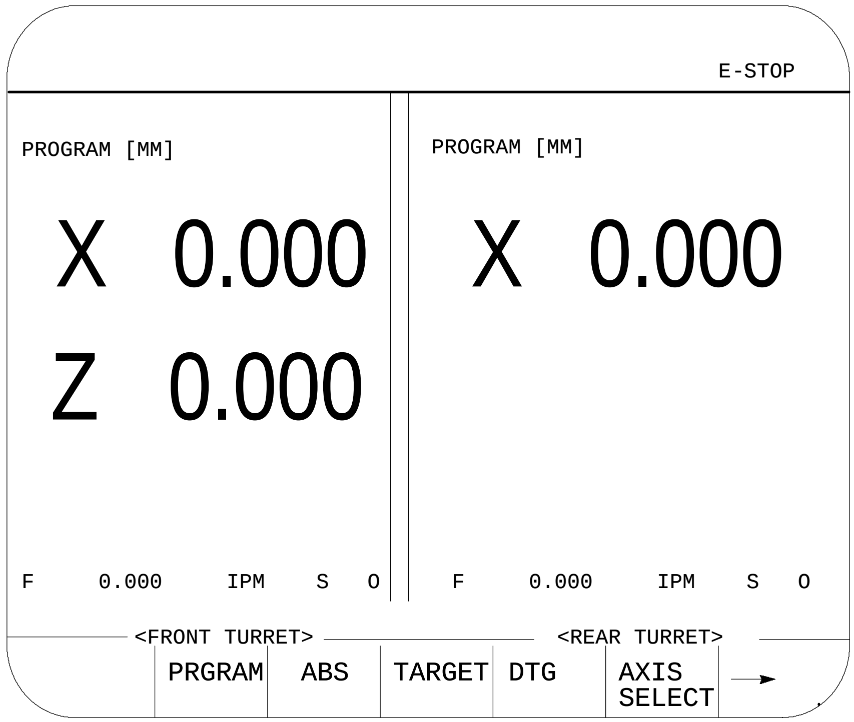
When you press the {SPLIT ON/OFF} softkey, you can view information for both processes. The screen displays two 40-column screens on one 80-column screen. Process 1 is displayed on the left, and process 2 is displayed on the right. The active process appears in reverse video.

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.



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 installer's 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	SWITCH 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, and 9/290 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 a program to execute, select one now, following these steps:

1. Press the `{PRGRAM CHECK}` softkey.

`(softkey level 1)`

PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	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. 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 MESAGE	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 keeps 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

(softkey level 2)
    ↑  |SELECT|QUICK |STOP  |T PATH|T PATH|
       |PRGRAM|CHECK |CHECK |GRAPH |DISABL|
```

```
QuickCheck Graphics

(softkey level 2)
    ↑  |SELECT|QUICK |STOP  |T PATH|T PATH|
       |PRGRAM|CHECK |CHECK |GRAPH |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 DISABL} softkey also appears on level three. If you press the {PRGRAM MANAGE} softkey on level one, followed by {ACTIVE PRGRAM} 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]
```

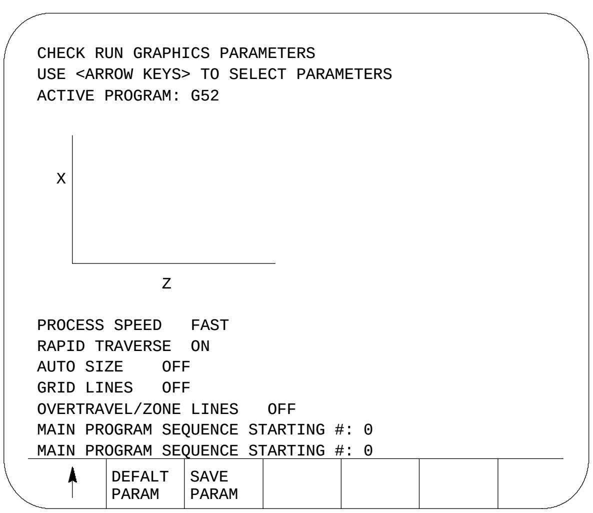
Follow these steps to change your parameters. We assume you have already completed the procedure to run graphics. See page 8-25 for details on running a graphics program.

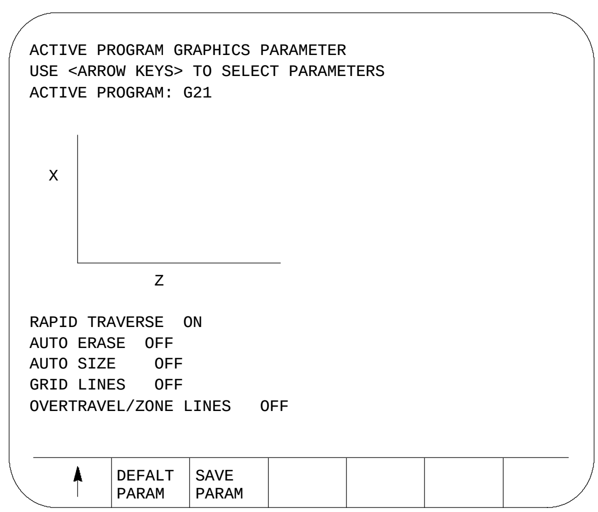
1. To load parameters, press {GRAPH SETUP}.

(softkey level 4)

↑	CLEAR GRAPHS	MACHIN INFO	ZOOM WINDOW	ZOOM BACK	GRAPH SETUP

The control displays these graphics parameter screens:





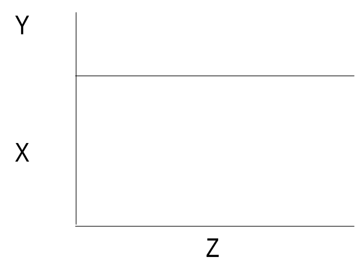
2. Set Select Graph. Use the up and down cursor keys to select the axes. Then set them by pressing the left or right cursor keys. The data for the selected axes change each time you press the left or right cursor key.

   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 QucikCheck 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.

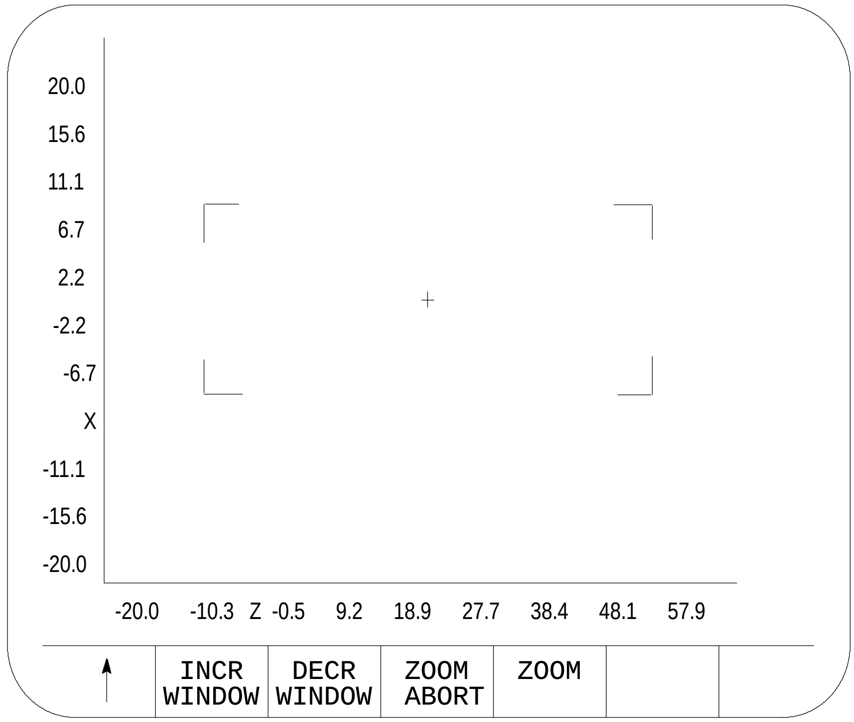
Press the {MACHNE INFO} softkey to activate the machine information window. The window toggles on and off each time you press the softkey.

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 WINDOW	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 WINDOW	DECR WINDOW	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.

3. 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.

4. 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)`

↑	INCR WINDOW	DECR WINDOW	ZOOM ABORT	ZOOM		

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 MESAGE	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	DEVICE SETUP	MONI-TOR	TIME PARTS	→

PTOM SI/OEM		SYSTEM TIMING			→

The control changes the screen to display the PTO screen, as shown in section 8.6.

* 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 MESAGE} 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 MESAGE	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
   - exit the PTO screen by pressing the up arrow softkey

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 MESAGE	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	QUICK CHECK	SYSTEM SUPORT	→

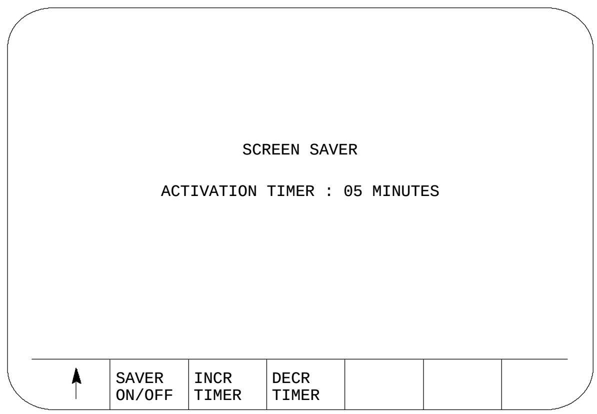
2. Press the [SCREEN SAVER] softkey.

(softkey level 2)

PRGRAM PARAM	AMP	DEVICE SETUP	MONI- TOR	TIME PARTS	→

PTOM SI/OEM		SYSTEM 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

Chapter 9

Communications

9.0 Chapter Overview

This chapter contains this information:

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 parameters separately for each device.

9.1.1 Setting Communication Port Parameter Values

This section describes how to change communication parameter values. A description of each of the parameters is presented on page 9-3.

Display the device setup screen as follows:

1. Press the {SYSTEM SUPORT} softkey.

```
(softkey level 1)

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

        FRONT   ERROR   PASS-   SWITCH          --->
        PANEL   MESAGE  WORD    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		SYSTEM TIMING			→

Figure 9.1
Device Setup Screen

```
                                                    E-STOP

      SERIAL PORT:   A
      DEVICE:        DECITEK AB 8000-XPDR
      PORT TYPE:     RS232
      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. See page 9-4 on selecting a device.

3. Use the up and down cursor keys to move the cursor to the parameter you want to change. The current value for each parameter appears in reverse video.

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 you press the left or right cursor key (or hold).

   Important: Select the SERIAL PORT (A or B) first and then select the DEVICE (see Figure 9.1). All other parameters are device-dependent.

5. To save the changes made to the communications parameters for a specific peripheral device, press the {SAVE CHANGE} softkey. The control displays the message “SAVE COMPLETED.” 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 by using the left and right cursor keys, 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 become your current settings.

9.1.2 Communication Port Parameters

SERIAL PORT (setting of input/output port)

The control has two communication ports: port A and port B. This parameter selects which of the two ports is going to be modified. Choose between ports A or B by pressing the left and right cursor keys.

All of the following parameters can be set independently for each communication port (A or B).

Important: If your system is a 9/230 grinder, it does not have port A. Port A options discussed in this chapter are not selectable and can not be configured for a 9/230 grinder.

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 USER PUNCH, USER PRINTER, or USER READER.

Table 9.A
Available Devices for Communication with the 9/SERIES Control

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 a more detailed list of the available peripheral devices and their function, see the integration/maintenance manual.

PORT TYPE

Port type options differ depending on the port you select.

Port	Type
`Port A`	RS232
`Port B`	RS232 or RS422

Port A is not available on 9/230 grinders.

BAUD RATE

You can set the baud rate at these speeds (in bits per second):

```
300, 600, 1200, 2400, 4800, 9600, 19200
```

See the documentation provided with your peripheral device.

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
```

See the documentation provided with your peripheral device.

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

See the documentation provided with your peripheral device.

STOP BIT (number of stop bits)

Select the number of stop bits with this parameter. You can select:

```
1, 1.5, or 2 bits
```

See the documentation provided with your peripheral device.

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. See the documentation provided with your peripheral device.

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 period, 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

UNLIMT (no time-out)
```

OUTPUT CODE

Select EIA (RS-244A) or ASCII (RS-358-B) as output codes for devices with data lengths configured as 8 bit. The output code can not be configured for data lengths configured as 7 bits and is displayed as N/A.

AUTO FILENAME

This parameter is valid only if you are inputting part programs to the control from a tape reader (see DEVICE for details). Use this parameter only if your tape contains more than one part program.

AUTO FILENAME lets you 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	The first program on a multiple program tape must have its program name 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. The control ignores all program names that exist as the first block of the program 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). The control generates an error if when reading program names into memory, the control needs to increment a program name that is greater than 99999. 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.
No	The program name for each program on a multi-program tape must be 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, the control generates an error. 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.

STOP PRG END

This parameter is available only if you have selected a tape reader as your device (see 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. See 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 stops 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 (see DEVICE for details). Use this parameter 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" (see 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." See the descriptions for M-codes in chapter 10 for details.

- M02, M30 – see the descriptions for M-codes in chapter 10 for details
- M99 – see the descriptions for M-codes in chapter 10 for details
- % – also used as end-of-tape code. See section on STOP PRG END (page 9-7)

  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 you use an N-code to designate the program name, be aware that the control interprets all part program blocks that contain N-codes as new part programs.

9.2 Inputting Part Programs from a Serial Peripheral

If you load a program 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 page 9-1).

2. Press the {PRGRAM MANAGE} softkey.

```
(softkey level 1)
```

PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

The Program Directory Screen is displayed:

```
SELECTED PROGRAM:

 DIRECTORY                  PAGE    1  OF    1

  NAME     SIZE          COMMENT

 O12345      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 PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

3. Select the directory into which you want the program to be copied. You can select the main directory or the protectable directory. The control displays the main directory as the default directory at power-up. The protectable directory must be selected using the {CHANGE DIR} softkey.

   The {CHANGE DIR} softkey controls access to the protectable part program directory. This softkey is password protected. You must have the proper password to be able to access this softkey. See page 5-42 for additional information on the protectable directory and the {CHANGE DIR} softkey.

4. Press the {COPY PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

5. If you have already entered the name in the program, skip step 5. and go to step 6. Otherwise, enter the program name to copy by either selecting it using the up and down cursor keys or typing it in by using the alphanumeric keys on the keyboard. The control displays the program name on the input line (line 2) of the screen next to the prompt “SELECTED PROGRAM:”.

   If you make an error while typing in the new program name, you can make corrections on the input line as described on page 2-41.

6. Select the device to copy from using the following 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 changed to the “COPY PARAMETERS” screen, as shown in Figure 9.2, that 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 on page 9-4.

Figure 9.2
Copy Parameter Screen

```
              COPY PARAMETERS

                FROM:  (SELECTED PORT NAME)
                  TO:  (PROGRAM 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

      ↑   SINGLE  MULTI   OUTPUT
          PRGRAM  PRGRAM  ALL
```

7. Specify if you want to copy one program or multiple programs.

   - Input Single Program
     Press {SINGLE PRGRAM} to copy one program from tape. Input terminates when the first program end or tape end code is encountered.

   - Input Multiple Programs
     Press {MULTI PRGRAM} 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, see the AUTO FILENAME, STOP PRG END, and PROGRAM END parameters described beginning on page 9-7.

When you input a program, the control does not erase the program from the peripheral device. If the program being read contains a comment <u>in the first block</u>, 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	

ATTENTION: If you select overwrite, the control erases 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 is lost, and the new one is not copied.

9.3 Outputting Part Programs to a Serial Peripheral

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 page 9-4).

2. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

The control displays the screen as shown in Figure 9.3.

Figure 9.3
Program Directory Screen

```
SELECTED PROGRAM:

 DIRECTORY                  PAGE    1  OF    1

  NAME     SIZE          COMMENT

 O12345      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 PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

3. Press the {COPY PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

4. Enter the program name to output from memory. Two ways to do this are available:

   - 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 and down cursor keys to position the cursor at the program you want to send. The selected program appears 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 9.4).

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 (see Figure 9.3) 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, see chapter 10.

ATTENTION: Be aware that the information in the directory such as program name and comment <u>are not</u> 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. See chapter 10 for details on "O" word program names.

- After the last program is output to the tape, press {FINISH TAPE} to punch the tape end 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 | MULTI  | OUTPUT |        |        |
      | PRGRAM | PRGRAM |  ALL   |        |        |
```

Figure 9.4
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 page 9-4).

2. Press the {PRGRAM MANAGE} softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS-WORD	SWITCH LANG		→

3. Press the {VERIFY PRGRAM} softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

↑	VERIFY PRGRAM	PRGRAM COMENT	DELETE PRGRAM	RENAME PRGRAM	INPUT DEVICE	→

↑	REFORM MEMORY	CHANGE DIR				→

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, you type the name of the program resident in the control only.

```
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)

↑	VERIFY PORT A	VERIFY PORT B	VERIFY MEMORY			

6. Press the {VERIFY YES} softkey. To abort the verify operation, press the {VERIFY NO} softkey.

```
(softkey level 4)
```

↑	VERIFY YES	VERIFY NO				

The control displays one of the following messages when you perform the verify operation:

"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 punch, 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

Chapter 10

Introduction to Programming

10.0 Chapter Overview

This chapter covers an introduction to programming part programs. The major topics described in this chapter include:

The control performs grinding 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, wheel dressing, and other CNC functions.

You can execute part programs from the control's memory or a CNC tape. You can execute programs on tape directly from the tape, or load them into the control and execute them 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:	See chapter:
enter and edit part programs through the keyboard	5
edit part programs off line and download them to the control	6
enter a part program from a peripheral device	9

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

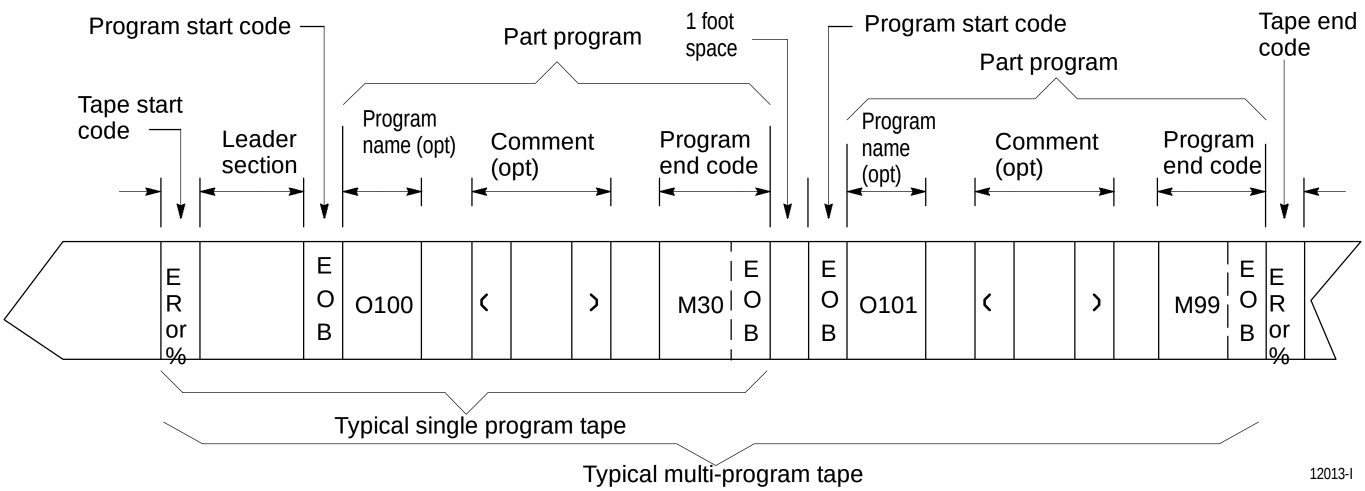
Tape with Program End = M02, M30, M99

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 (see chapter 10 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 if no filenames are included in the programs on the tape, you have to enter all filenames from the keyboard when prompted. For more information, see the section on O word programming in this chapter (page 10-34).

The control automatically recognizes EIA or ASCII during input when it reads the first EOB code from the tape.

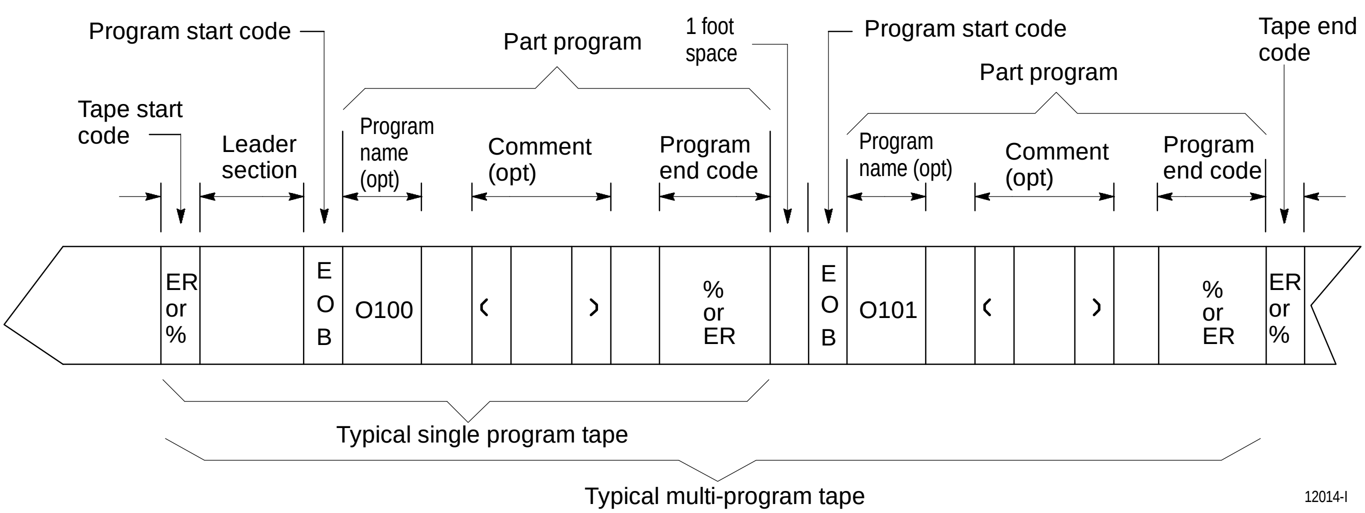
Figure 10.1
Tape Configuration (Program End = M02, M30, M99)



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). See this chapter 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 to be 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

(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 EAI format), or N word after the program start code.
Selected form 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 generates an O word block as the first block of the program written to tape.
No name is used	A program is 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. You can use these codes 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 individual grinding 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 grinding 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:

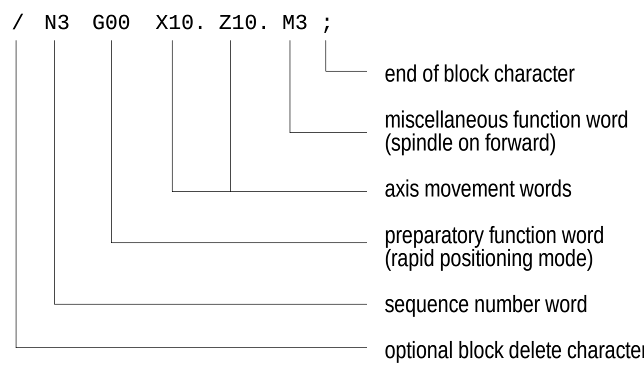
- characters -- A character is a number, a letter, or a symbol that means something specific to the control. For example, 1 G ; are characters the control recognizes as meaningful information.
- addresses -- An address is a letter that defines the instruction for the control. Examples of addresses are: G, X, Z, F.

- words -- A word consists of an address followed by a numeric value. Examples of words are: G01, X10.5, F.1., M2. Each word requires a specific format for its numeric part. These formats are given on page 10-21.

- codes -- There are industry standards for many of the G and M words used here. For that reason, they are often referred to as G or M “codes”.

- parameters -- The control has a number of fixed cycles that are initiated by a specific G word. 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 G81 reciprocation cycle, its value refers to the coordinate of the primary reversal point. In that case, it is a “parameter” of the G81 reciprocation 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 grinding operation.

Important: To make jumps, loops, or calculations within an executing program or subprogram, use the paramacro features as described in chapter 20.

A part program has a:

- beginning -- setting up the control and the grinder to perform the operations wanted

- middle -- performing the grinding operations

- end -- returning the grinding wheel to a stopping position and preparing the control for the next part program

The blocks programmed vary for each section of the program. Consider Example 10.2.

Example 10.2
Sample Part Program

`G91G21;`	beginning

`G00X28.;`
`G81Z5.K-2.;`
`G00X5;`
	middle

`G80;` `M02;`	end

A complete part program can consist of a main program and several subprograms. For details on subprograms, see page 10-12.

10.2.1 Program Names

Enter as many as 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 you enter a new program name 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.

The control first lists subprograms in numerical order, from lowest to highest. Following the subprograms are the main programs, which are listed in alphabetical order.

Entering Program Names

To enter a program name, do the following:

1. Press the softkey {PRGRAM 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 initiates 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
O00123	O00123
O123	O00123
123	O00123
12345	O12345

Example 10.4
Legal Program Name Blocks

```
O12345;
O12345(DRESSING PROGRAM);
O333
O2;
```

10.2.2 Sequence Numbers

You can assign each block in a part program 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.

You can assign sequence numbers at random to specific blocks or to all blocks. You can call blocks assigned sequence numbers later by designating their sequence number. Sequence numbers are necessary to make program jumps and to specify a block for paramacro calls and returns.

Example 10.5 shows 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 page 10-11). You do not have to program the N word first in the block; the control still finds it for jumps. However, it is not 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.

You can assign different blocks 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 sequence number that is closest to the calling block is used. 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 renumbers existing sequence numbers or assigns all block sequence numbers. For details, see 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 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 skips (does not execute) all remaining programmed commands if you turn on the block delete feature. Turn on this feature by pressing the {FRONT PANEL} softkey, or by activating the installed switch (optionally) on the MTB panel. If you use the {FRONT PANEL} softkey, only block delete /1 is available.

Your 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
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 is 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 <u>before</u> that block is read into buffer memory. Otherwise, it is not skipped.

The control considers a “/” without a number to mean “/1”. However, you must program a “/1” if you use more than one block delete number 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(es), see the PAL Reference manual and the documentation prepared by your system installer.

10.2.5 End of Block Statement

All program blocks must have an end of block statement 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.

To specify an end of block character “;” at the keyboard, use the [EOB] key on the operator panel. If editing part programs off line (see chapter 6), the end of block character should not be entered when blocks are keyed in. The control automatically inserts end of block ; when you download the program.

10.3 Using Subprograms

When the same series of blocks are repeated more than once, we recommend that you 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 20 for details on paramacros and local parameters.

This section explains the following:

- main and subprograms
- subprogram calls

To make jumps, loops, or calculations within an executing program or subprogram, use any of the paramacro features as described in chapter 20.

You can call a subprogram in an MDI command, however, an MDI command cannot contain an M99 code.

ATTENTION: Any edits that you made to a subprogram or paramacro program (as described on page 7-6) 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 described on page 7-6).

10.3.1 Subprogram Call (M98)

Generally the control executes programs sequentially. When you enter an M98Pnnnnn (“nnnnn” representing a subprogram number) command in a program, the control merges the subprogram, designated by the address P, with the main program immediately before the block that follows the M98 command.

The control first searches the main program directory for the subprogram called by an M98. If the control does not find the subprogram in the main program directory, it then checks the protectable program directory for the subprogram. If subprograms in the main and protectable directories have the same name, the control uses the subprogram in the main directory. The control issues the error message “CANNOT OPEN SUBPROGRAM” if it cannot find the subprogram designated by the M98 command.

For example,

```
M98 P00001 ;
```

causes execution to transfer from the current program to the subprogram numbered 00001.

Important: For you to use a program 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. The letter O is omitted.

Sometimes you want to execute a subprogram more than once. For example,

```
M98PnnnnnLmm;
```

causes the control to merge the subprogram numbered nnnnn in the main program mm times. When you enter an L command in a M98 command, the control merges 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. 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 cannot 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 stop program execution. The program resets or rewinds if an M30 code is executed or the program ends if the M02 code is executed.

10.3.2 Main and Subprogram Return (M99)

The M99 code acts as a return command in both subprograms and main programs; however, there are specific differences:

If you use M99 in a:	M99:
main program	• executes all commands in the block, regardless of 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
subprogram	• tells the control the end of a subprogram. The control does not execute any commands following the M99 as part of the subprogram • causes the control to return to the call 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.

if you programmed:	Then:
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;		

Example 10.8
Subprogram Calls and Returns (continued)

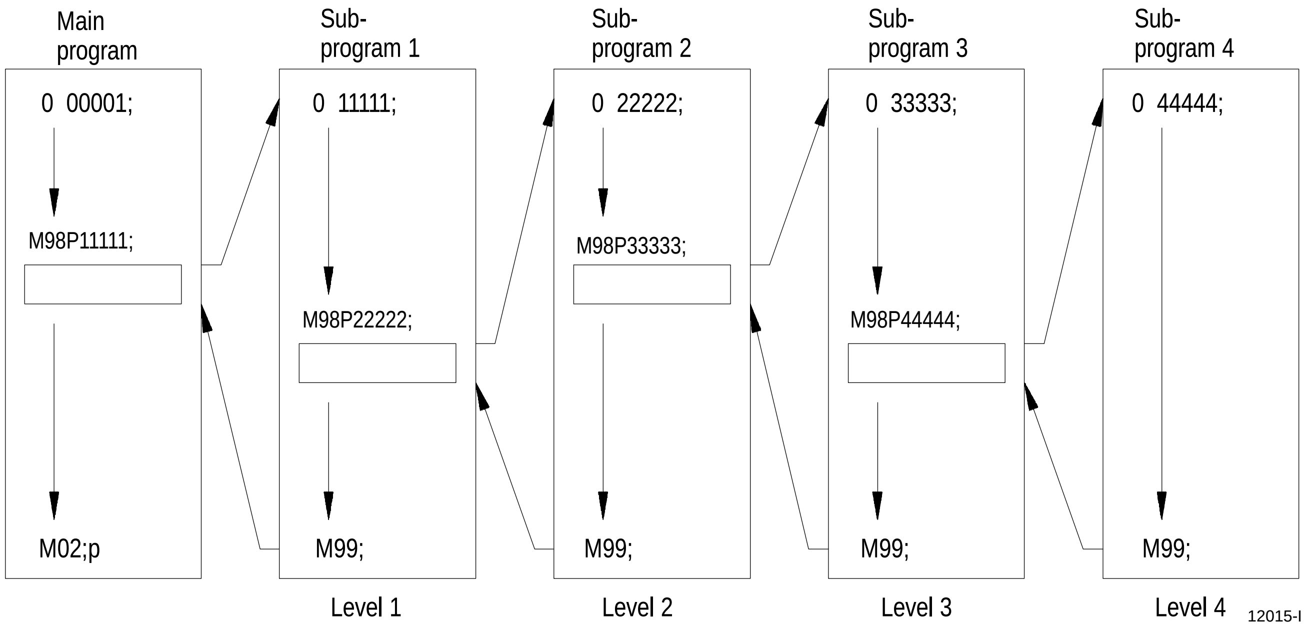
The following path of execution results when the main program above is selected as the active program.

```
(MAIN PROGRAM);
N00010...;
N00020...;
N00030M98P1;
(SUBPROGRAM 1);
N00110;
N00120...;
N00130M99;
N00040...;
N00050...;
N00060M98P2L2;
(SUBPROGRAM 2);
N00210;
N00220...M99;
(SUBPROGRAM 2);
NOO210;
N00220...M99;
N00070M30;
```

10.3.3 Subprogram Nesting

We use nesting as the term to describe one program calling another. The program called is said to be a nested program. When you call a subprogram 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 can be nested up to a maximum of 4 levels.

Figure 10.3
Subprogram Nesting



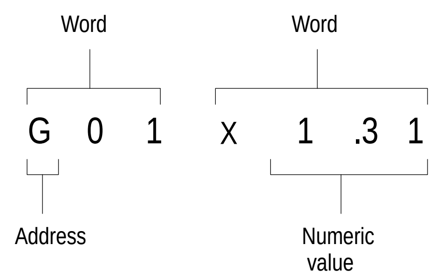
Important: Calling a macro (see chapter 20) does not add to the nesting level of any active subprograms. Up to 4 subprograms can still be nested though the combined total of nested macros and subprograms cannot exceed 8.

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 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 number to the left of the decimal point indicates the number of digits acceptable as integers, and the number to the right of the decimal point indicates the number of fractional digits acceptable.

The format "X3.4" for an X word, indicates that 3 digits to the left and 4 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.

Your 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 your 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 in Table 10.A, 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

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.

Your 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 you enable the feature, the error “NUMERIC MISSING” would have occurred upon execution of either of those blocks.

In alphabetical order, Table 10.B shows 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, see the documentation prepared by your system installer.

You can alter many of the addresses themselves 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 describe these words in more detail, including variations in their meanings when they are associated with certain G codes. All words described in this manual assume the formats and addresses in the following table have not been changed by your system installer.

Important: The formats in this table indicate the maximum number of digits to the left and 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. See your system installer's manual for specific formats.

Table 10.B
Word Formats and Descriptions

Address	Valid Range inch	Valid Range metric	Function
A	8.6 3.3	8.5 3.3	Rotary axis about X (AMP assigned) Angle in QuickPath Plus programming
B	3.0	3.0	Second miscellaneous function (AMP assigned)
C	8.6 8.6	8.5 8.5	Rotary axis about Z (AMP assigned) Chamfer length in QuickPath Plus programming
D	8.6 3.0	8.5 3.0	Fixed cycle parameter Number of dresses in fixed cycle
E	2.6 8.6	3.7 8.5	Thread lead Reciprocation Feedrate
F	8.6	8.5	Feedrate function (F word)
G	2.1	2.1	Preparatory function (G word)
I	8.6 8.6 8.6 8.6	8.5 8.5 8.5 8.5	X arc center in circular interpolation X lead in helical interpolation Parameter in fixed cycles (AMP assigned) Exit move vector in cutter compensation
J	8.6 8.6	8.5 8.5	Parameter in fixed cycles (AMP assigned) Exit move vector in cutter compensation
K	8.6 8.6 8.6	8.5 8.5 8.5	Z arc center in circular interpolation Parameter in fixed cycles (AMP assigned) 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 5.3	5.0 5.0 5.3	Subprogram/paramacro program name Dressing program name Length of dwell in G04 and fixed cycles
Q	8.6	8.5	Parameter in fixed cycles
R	8.6 8.6	8.5 8.5	Arc radius QuickPath Plus radius designation

Address	Valid Range inch	Valid Range metric	Function
S	5.3 3.3 4.3	5.3 3.3 3.3	Spindle rpm function Spindle Orient CSS
T	6.0	6.0	Tool selection function
U	5.3	5.3	Length of dwell in G04 and fixed cycles.
X	8.6 5.3	8.5 5.3	Main axis (AMP assigned) Length of dwell in G04
Y	8.6	8.5	Main axis (AMP assigned)
Z	8.6	8.5	Main axis (AMP assigned)

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 .00001mm. However, the actual range of programmable values depends on specifications determined by your system installer.

By using AMP to establish the format of numeric values for words, your 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

See your 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 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. Integrand words for these axes are assumed by be I, J, and K respectively. Incremental or parallel axis names for these axes are assumed to be U, V, W, respectively.

10.5.2 A_L_,R_,C_ (QuickPath Plus Words)

To simplify programming an angle, corner radius, or chamfer between two lines, all you need is the angle between the lines and the radius or chamfer size connecting them. You can use this method of programming to simplify the grinding of many complex parts.

QuickPath words are made up of the addresses below followed by the desired numeric value.

If you see:	It means:
A	angle
L	length
,R	corner radius
,C	chamfer size

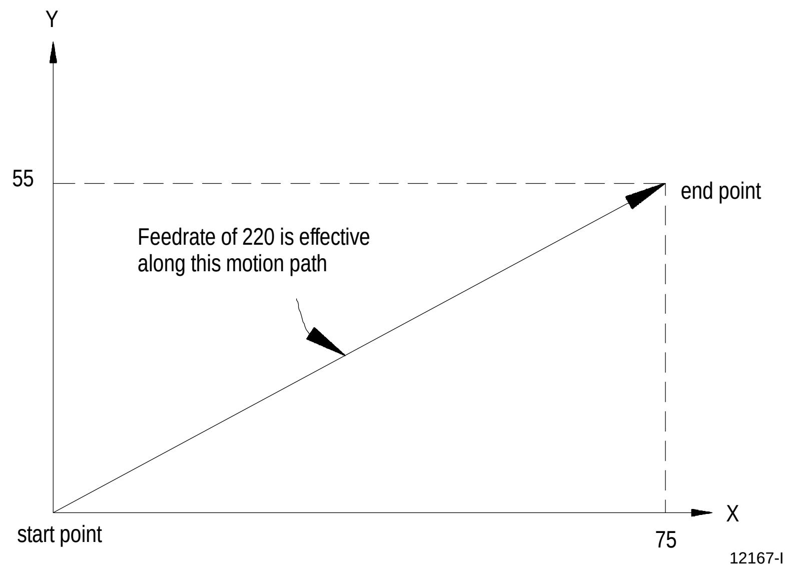
Important: A comma “,” must precede the R and C address characters for the control to recognize them as radius (,R) and chamfer (,C) words.

For details and examples using these words, see chapter 12.

10.5.3 F Words (Feedrate)

An F word with numeric values specifies feedrates for the grinding and dressing moves 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 Figure 10.5.

Figure 10.5
Feedrate Vectors



The term "feed" refers to moving the grinding wheel at a specific velocity along a path. "Feedrate" is the velocity programmed for the feed of a grinding wheel.

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	thread grinding
inches per rev	ipr	thread grinding

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 and special pre-assigned feedrates, see chapter 12.

Important: Feedrates programmed in any of the feedrate modes (G94 or G95) can be overridden by use of the <FEEDRATE OVERRIDE> switch. For details, see chapter 12.

10.5.4 G Words (Preparatory Functions)

The preparatory function is designated by a G word consisting of address G followed by a two-digit value. In some cases, the G word has an additional decimal digit. Because many of these are set by industry standards, they are usually referred to as G codes. 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.

Important: When programming more than one G code from the same group in a block, the control interprets the G codes in the block as they occur sequentially from left to right. Any mode that changes in a block affects only the interpretation of the values to the right of the G code. Characters to the left of the G code use the old operating mode.

When the control executes an M02 or M30, the modal M and G codes can be reset to their default values. Whether or not this occurs is determined by your system installer through AMP. These default values also become active at power up, E-STOP reset, or control reset. Your system installer determines these default G codes in AMP.

Important: You can also express G codes in terms of a parametric expression (for example G[#12+6]). For details, see chapter 20.

How the modal G codes are executed is explained in Example 10.9, using G00 and G01, both classified into the same G code group.

Example 10.9
Programming Modal G codes

`G00 X1. Z2.;`	G00 mode is effective
`Z3. ;`	G00 mode is effective
`G01 X2. Z1. ;`	G01 mode is made effective
`X3. Z3. ;`	G01 mode is in effect
`G00 X1.Z2. ;`	G00 mode becomes effective again
`G01 G00 Z3, ;`	G00 mode is in effect
`G01 G91 Z2 ;`	G01 and G91 both in effect

An example of non-modal G-code execution follows:

Example 10.10
Programming Non-Modal G Codes

`G00 X1. Z21 ;`	G00 mode is effective
`G28 X2.;`	G28 mode, this block only
`X2. Z1. ;`	G00 mode is effective
`G04 P2. X4. ;`	G04 active followed by move in G00

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)

Table 10.E breaks down the G codes into their 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 Code Table

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
G00	G00	01	Rapid Positioning	Modal
G01	G01		Linear interpolation	
G02	G02		Circular / helical interpolation CW	
G03	G03		Circular / helical interpolation CCW	
G04	G04	00	Dwell	Non-modal
G05	G05		Send Command and Wait for Return Status (used with 9/SERIES Data Highway Plus Communication Module)	
G05.1- G05.4	G05.1- G05.4		Send Command without Waiting for Return Status (used with 9/SERIES Data Highway Plus Communication Module)	
--	G07	18	Programming using radius values	Modal
--	G08		Programming using diameter values	
G09	G09	00	Exact stop	Non-modal
G10L2	G10L2		Setup Work Coordinate Offset Table	
G10L10	G10L10		Setup Tool Offset Values Geometry Table	
G10.2L1	G10.2L1		Communication Configuration Table (used with 9/SERIES Data Highway Plus Communication Module)	
G10.2L2	G10.2L2		Output Command Table (used with 9/SERIES Data Highway Plus Communication Module)	
G10.2L3	G10.2L3		Download Configuration Information (used with 9/SERIES Data Highway Plus Communication Module)	
G12.1	G12.1	21	Spindle 1 Controlling	Modal
G12.2	G12.2		Spindle 2 Controlling	
G12.3	G12.3		Spindle 3 Controlling	
G13	G13		QuickPath Plus use first intersection	Non-modal
G13.1	G13.1		QuickPath Plus use second intersection	
G14	G14	19	Disable Scaling	Modal
G14.1	G14.1		Enable Scaling	
--	G15	15	Angled-Wheel transformation off	Modal
--	G16.3		Angled-wheel normal motion	
--	G16.4		Angled-wheel two step motion	
G17	G17	02	Axis plane select (XY)	
G18	G18		Axis plane select (ZX)	
G19	G19		Axis plane select (YZ)	
G20	G20	01	Single pass O.D. and I.D. roughing	
G22	G22	04	Programmable Zone 2 and 3 (On)	
G22.1	G22.1		Programmable Zone 3 (On)	

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
G23	G23	04	Programmable Zone 2 and 3 (Off)	Modal
G23.1	G23.1		Programmable Zone	
G24	G24	01	Single pass rough facing cycle	
G27	G27	00	Machine home return check	Non-modal
G28	G28		Automatic return to machine home	
G29	G29		Automatic return from machine home	
G30	G30		Return to secondary home	
G31	G31		External skip function 1	
G31.1	G31.1		External skip function 1	
G31.2	G31.2		External skip function 2	
G31.3	G31.3		External skip function 3	
G31.4	G31.4		External skip function 4	
G33	G33	01	Constant lead thread grinding	Modal
G34	G34		Variable lead thread grinding	
G36	G36	22	Short block Acc/Dec clamp enable	
G36.1	G36.1		Short block Acc/Dec clamp disable	
G37	G37	00	Tool gauging skip function # 1	Non-modal
G37.1	G37.1		Tool gauging skip function # 1	
G37.2	G37.2		Tool gauging skip function # 2	
G37.3	G37.3		Tool gauging skip function # 3	
G37.4	G37.4		Tool gauging skip function # 4	
G39	G39	20	Dresser/Wheel compensation (linear gen. blocks)	Modal
G39.1	G39.1		Dresser/Wheel radius compensation (circular gen. blocks)	
G40	G40	07	Dresser/Wheel radius compensation cancel	
G41	G41		Dresser/Wheel radius compensation, left	
G42	G42		Dresser/Wheel radius compensation, right	
G50.1	G50.1	11	Programmable mirror image cancel	
G51.1	G51.1		Programmable mirror image	
G52	G52	00	Offset coordinate system zero point	Non-modal
G53	G53		Motion in Machine coordinate system	
G54	G54	12	Preset Work Coordinate System 1	Modal
G55	G55		Preset Work Coordinate System 2	
G56	G56		Preset Work Coordinate System 3	
G57	G57		Preset Work Coordinate System 4	
G58	G58		Preset Work Coordinate System 5	
G59	G59		Preset Work Coordinate System 6	
G59.1	G59.1		Preset Work Coordinate System 7	
G59.2	G59.2		Preset Work Coordinate System 8	

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
G59.3	G59.3	12	Preset Work Coordinate System 9	Modal
G61	G61	13	Exact stop mode	
G62	G62		Automatic corner override mode	
G64	G64		Cutting mode	
G65	G65	00	Paramacro Call	Non-modal
G66	G66	14	Modal paramacro call	Modal
G66.1	G66.1		Modal paramacro call	
G67	G67		Modal paramacro call cancel	
G68	G68	16	Part rotation	
G69	G69		Part rotation cancel	
G70	G70	06	Inch input	
G71	G71		Metric input	
G80	G80	09	Canned cancel	
G81	G81		Reciprocation on	
G81.1	G81.1		Reciprocation with predress on	
G82	--		Plunge grind cycle, axis 1	
--	G82		Incremental face grind cycle, axis 1	
G82.1			Plunge grind with predress, axis 1	
--	G82.1		Incremental face grind with predress, axis 1	
G83	--		Incremental plane grind, axis 1 plunge	
--	G83		Incremental plunge grind, axis 2 plunge	
G83.1	--		Incremental plane grind, axis 1 plunge with predress	
--	G83.1		Incremental plunge grind, axis 2 plunge with predress	
G84	--		Incremental plane grind, axis 2 plunge	
--	G84		Multi-pass face cycle	
G84.1			Incremental plane grind, axis 2 plunge with predress	
--	G84.1		Multi-pass face cycle with predress	
G85	--		Continuous plane grind, axis 1 plunge	
--	G85		Multi-pass diameter plunge	
G85.1	--		Continuous plane grind, axis 1 plunge with predress	
--	G85.1		Multi-pass diameter plunge with predress	
G86	--		Continuous plane grind, axis 2 plunge	
--	G86		Shoulder grind cycle	
G86.1			Continuous plane grind, axis 2 plunge with predress	
--	G86.1		Shoulder grinder cycle with predress	
--	G87		Face plunge shoulder cycle	
--	G87.1		Face plunge shoulder cycle with predress	
--	G88		Diameter plunge shoulder cycle	Modal

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
--	G88.1	09	Diameter plunge shoulder cycle with predress	
--	G89		Multi-step plunge with blend	
--	G89.1		Multi-step plunge with blend and predress	
G90	G90	03	Absolute mode	Modal
G91	G91		Incremental mode	
G92	G92	00	Maximum CSS spindle RPM, coordinate offset	Non-modal
G92.1	G92.1		Cancel offsets	
G92.2	G92.2		Cancel select offsets and G92 presets	
G93	G93	05	Inverse time feedrate mode	Modal
G94	G94		Feed per minute feedrate mode	
G95	G95		Feed per revolution feedrate mode	
G96	G96	17	Constant surface speed on	
G97	G97		RPM spindle speed mode (CSS off)	

A set of default G codes becomes effective at power-up, when the control is reset, or an emergency stop condition is reset. These default G codes are selected by your system installer in AMP. They can be seen on the status display screen after power-up or control reset.

10.5.5. I J K Integrand Words

Integrand words are typically used to define parameters that relate to a specific axis for a fixed cycle, or circular motion block; however, they are 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 an arc or circle.

Your system installer has the option of assigning either I, J, K, or none as the axis integrand name for a specific axis. This manual assumes the following:

Assign:	To the integrand name for the:
I	X axis
J	Y axis
K	Z axis

Important: See your system installer's documentation to make sure these assumptions are true. If not, it is important that the programmer realize that all examples and formats in this manual that use an I, J, K need to have their letter replaced with the correct integrand word for your system.

10.5.6 M Words (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 if the M codes are executed:

- at the same time as the axis motion
- before the axis motion
- after the axis motion is completed

You can also alter this order of execution by using the paramacro feature, system parameter #3003 (see page 20-19).

Your 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 revert back to their power-up condition and non-modal M and G codes are reset to their default values. If the control does not reset M and G codes, all modal and non-modal M and G codes remain at their present value.

Table 10.F shows the basic M codes for the control. A part program block can contain as many basic M codes as you want. If more than one M code from any group is programmed in the same block, the right-most M code in that block for that group is the active M code for the block.

Your system installer may have defined additional M codes in PAL. As many as 4 M-codes can be activated in any one block. If you program more than 4 in any one block, the right-most 4 in that block are activated. The control ignores other M-codes in the block. See documentation provided by your system installer for details on non-basic M codes and their operation.

Table 10.F
Basic 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
M30	NM	4	Program end and reset (tape rewind)
			SPINDLE 1
M03	M	7	Spindle positive rotation (cw)
M04	M	7	Spindle negative rotation (ccw)
M05	M	7	Spindle stop
M19	M	7	Spindle orient
			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
			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	CSS permit
M59	M	10	CSS prohibit
M98	NM	5	Sub-program call
M99	NM	5	Sub-program end and program jump

See pages 12-72 through 12-74 for descriptions of the spindle M–codes.

The following describes the basic M codes provided with the control.

(1) Program Stop (M00)

When the control executes M00, program execution is stopped after the block containing the M00 is executed. At this time, the CRT displays the "PROG STOP" message. To restart the operation, press the <CYCLE START> button.

(2) 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 your system installer.

ATTENTION: Once axis reciprocation begins, it continues through program block execution until stopped by a G80, an end of program (M02, M30, M99), a change to manual mode or an emergency stop. This means executing an M00 or an M01 in a part program does not necessarily stop the reciprocating axis.

(3) End of Program (M02)

If you execute a program from control memory, the M02 code acts the same as an M30. The control stops program execution and enters into the cycle stop state. The program is reset to the first block and a <CYCLE START> begins 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 the control executes M02, it stops program execution and enters into 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 can also result in a spindle and coolant supply stop. For details, see the instruction manual prepared by your system installer.

(4) End of Program, Tape Rewind (M30)

If you execute a program from control memory, the M30 code acts the same as an M02. The control stops program execution and enters the cycle stop state. The program is reset to the first block and a <CYCLE START> begins 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 the control executes M30, it stops program execution and automatically rewinds the tape to the beginning of the program just executed. After the tape is rewound, the control enters the cycle stop state. A <CYCLE START> begins part program execution over again (see M99 for auto cycle start).

With some machines, the M30 code can also result in a spindle and coolant supply stop. For details, see the instruction manual prepared by your system installer.

(5) Overrides Enabled (M48)

When M48 is executed, the feedrate override, rapid feedrate override, and the spindle speed override functions become effective. Normally, these are enabled on power-up without requiring this M code to be executed. An M48 cancels an M49 and your system installer can choose which is active upon power-up.

(6) Overrides Disabled (M49)

Use the override disable M code (M49) to ignore any override set by the operator on the MTB panel. When the control ignores the override setting, the axis feedrate, rapid feedrate, and the spindle speed override values are all set to 100%. An M49 cancels an M48 and your system installer can choose which is active upon power-up.

(7) Constant Surface Speed Mode Enable (M58)

M58 cancels M59 mode and allows the control to recognize programmed G96 constant surface speed mode with S words. The spindle resumes the speed it was revolving at prior to the designation of M59.

ATTENTION: Restoring the constant surface speed mode might cause the spindle speed to increase or decrease rapidly, depending on the grinding wheel position.

(8) Constant Surface Speed Mode Disabled (M59)

M59 cancels M58 and G96 making the constant surface speed mode ineffective. The spindle continues to revolve at the speed it was at the moment the M59 executed.

The spindle speed can be directly programmed using an S word.

(9) Subprogram Call (M98)

When M98 is executed, a subprogram is called and executed. You can use this word in any program including an MDI program. For details on programming an M98, see page 10-12.

(10) 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 can be contained in a subprogram called by an MDI command. For details on programming an M99, see page 10-12 or chapter 20.

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. Pressing <CYCLE START> automatically starts program execution.

ATTENTION: The M99 code is commonly used as the end of program for fully automated systems that automatically load the next part. Typically this code requires that some PAL interface be written that make sure the part is fully loaded and ready for grinding before block execution is allowed to restart. Failure to do so can cause injury to operators or damage to equipment.

Other M codes are described in later sections that deal specifically with their function.

Important: When the miscellaneous function lock feature is activated, the control ignores 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 on page 7-2.

2nd Miscellaneous Function (B Word)

Your system installer can use the 2nd miscellaneous functions to distinguish a set of miscellaneous functions from the normal M code miscellaneous functions. This manual assumes the B word is used to call second auxiliary functions. Any alphabetic character which is not used for other functions can be used instead of B by setting the proper AMP parameter. For details, see documentation prepared by your system installer, or the AMP reference manual.

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 ignores 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 on page 7-2.

10.5.7 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.

You can assign sequence numbers at random to specific blocks or to all blocks. You can call blocks assigned sequence numbers later by designating their sequence number. Sequence numbers are necessary to make program jumps and to specify a block for subprogram calls and returns. For details on sequence number, see page 10-9.

10.5.8 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 5 numeric characters following it.

10.5.9 P,L (Main Program Jumps and Subprogram Calls)

When you repeat the same series of blocks more than once, we recommend that you program them using a subprogram.

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 described in chapter 20.

Use P words in a subprogram call (M98) or paramacro call to designate the specific program being called. The P address is followed by the program name being called.

Use L words in a subprogram call (M98) and some cycles to designate a repeat count for a subprogram. The number following the L address designates the number of times a subprogram is executed consecutively before execution is returned to the main program.

For details on subprograms, see page 10-12.

10.5.10 S Word (Spindle Speed)

The spindle function has two modes:

- Constant Surface Speed Mode (G96) maintains the speed of the part surface relative to the grinding wheel surface at a constant rate. A change in part or wheel diameter results in a change in spindle speed as needed to maintain the surface speed programmed with a G96 S word. Wheel diameter is only monitored if S axis is selected in AMP as the CSS axis.

- RPM Spindle Speed Mode (G97) maintains a constant spindle speed equal to the programmed S word, making the actual surface speed dependent on the working diameter.

Spindle speed and surface speed values are programmed by an S word followed by as many as four digits. S words are modal and remain active until another S word is executed.

Important: Your system installer sets a maximum speed for each gear range in AMP. If you program an S word requesting a spindle speed that exceeds this limit, the spindle speed is held at the AMP defined maximum. A new value can be set for this maximum RPM by programming a G92 followed by an S word (see page 12-43).

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 can search for and select the appropriate gear range to attain the programmed spindle RPM. The operation of gear changing and how it is implemented is PAL dependent. See your system installer's documentation for details on how a gear change operation is performed.

For details on programming spindle speeds, see page 12-34.

Important: When the miscellaneous function lock feature is activated, the control ignores 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 as described on page 7-2.

You can override programmed spindle speeds with the <SPINDLE SPEED OVERRIDE> switch on the MTB panel. This switch can be positioned in 5% increments within a range of 50 - 120%. For details, see your system installer's documentation.

Use the override disable M code (M49) to ignore any override set on the MTB panel. When the override settings are ignored, the axis feedrate, rapid feedrate, and the spindle speed values are all set to 100%. For more information on spindle functions, see page 12-34.

10.5.11 T Words (Length, Radius, and Orientation Offsets)

Modern grinders typically require the selection of multiple control points on the wheel for grinding different contours and dressing different profiles. Use the wheel length offset feature to shift the control point to the various cutting locations about the grinding wheel. Activate wheel length offsets with a T word as described in chapter 13.

Also, use the T word to select an active dresser/wheel radius compensation offset number. This offset compensates for the radius of wheel corners or dresser tip and orientation relative to the part/wheel surface. Dresser/wheel radius compensation is described in detail in chapter 15.

The control can store as many as 32 wheel length offsets and 33 radius/orientation offsets. The actual number of available offsets on your system is configured in AMP by your system installer. Typically each time a different profile is dressed into the wheel or a different part of the wheel is used as the grinding surface, you need to activate different offsets.

The selection and activation of offsets is done by programming a T word. A T word can be programmed in any block in a part program.

Important: When the miscellaneous function lock feature is activated, the control ignores M, B, S, and T words in the part program with the exception of M00, M01, M02, M30, M98, and M99. This miscellaneous function lock feature is activated through the front panel screen (as described on page 2-11) or by an optional switch installed by your system installer.

The format for a T word is as follows:

```
Trrll;
```

Where:	Is:
rr	The dresser/wheel radius and orientation offset number. The radius/orientation offset number is the first two digits of the T word. See chapter 3 for details on radius/orientation data.
ll	The wheel length offset number. The wheel length offset number must be the last two digits of the T word. See chapter 3 for information on length offset values.

An offset number selected as zero (for either length or radius/orientation) selects an offset value of zero and effectively cancels wheel length offsets and dresser/wheel radius compensation. Dresser/wheel radius compensation is not canceled when a radius offset number of zero is programmed (G41 or G42 remain active). In this case, a radius value of zero is simply used. Dresser/wheel radius compensation must be canceled by programming a G40.

Important: T words are always right-justified. This means that programming a two-digit T word would only enter a wheel length offset value and not specify a radius/orientation offset number. Table 10.G gives examples of T words and the resulting offset numbers activated.

Table 10.G
Programming T Words

Program this T word	to activate this radius/orientation offset	and this length offset
T0102;	#01	#02
T1223;	#12	#23
T0;	cancels offset	cancels offset
T123;	#01	#23
T12;	cancels offset	#12
T1;	cancels offset	#01
T1200;	#12	cancels offset
T0012;	cancels offset	#12

From Table 10.G, you can see you cannot program a T word without inadvertently programming both a length offset and a radius/orientation offset. By not programming one of the offsets, the control assumes an offset of 00 is programmed and cancels any active offset value. We recommend always programming a T word followed by all 4 digits thus making sure you are not inadvertently canceling one offset while trying to activate another.

Important: Your system installer determines in AMP if all wheel length or radius/orientation offsets are canceled when the control is reset or an M02, M30, or M99 end of program block is executed.

Your system installer can also write PAL to automatically select and activate a wheel length and radius/orientation offset number. See your system installer's documentation and the PAL reference manual for details.

END OF CHAPTER

Chapter 11

Coordinate Control

11.0 Chapter Overview

This chapter describes the control of the coordinate systems on the control. G words in this chapter are among the first programmed because they define the coordinate systems of the machine in which axis motion is programmed.

A thorough understanding of this group makes programming easier by allowing full control of the coordinate systems.

11.1 Machine (Absolute) Coordinate System

The control has two types of coordinate systems:

- machine coordinate system. This is often referred to as the absolute coordinate system, which is unique to the individual grinding machine
- work coordinate system. This is defined based on the coordinate system used in the part drawing of a part to be ground by the machine

Programs are usually written based on the work coordinate system.

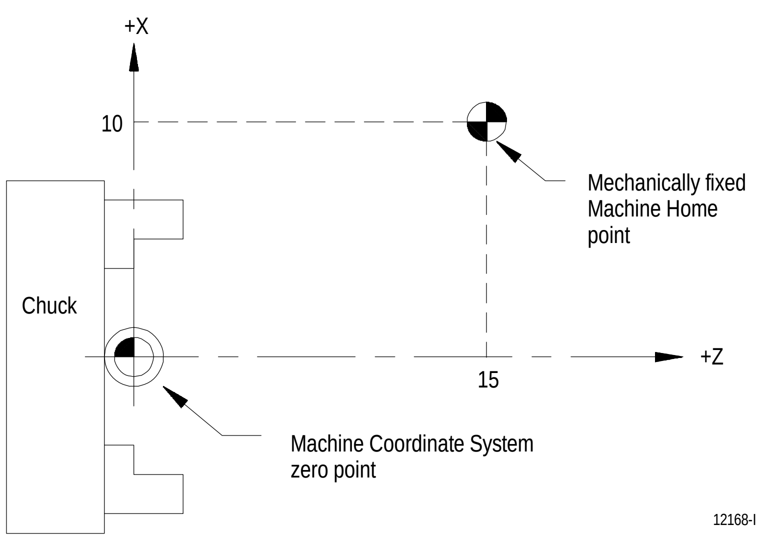
The machine coordinate system is the basic coordinate system set for every CNC grinder. It is established after completion of the machine homing operation. It cannot be offset or shifted in any way. Its position is determined in AMP by your system installer.

Important: You must home the machine before you can activate any coordinate system. 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, see page 4-11.

The zero point of the machine coordinate system is referenced from the machine home point. This is done by assigning a coordinated location to the machine home point. 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, your system installer has defined the zero point of the machine coordinate system by assigning the machine home point to have the coordinates X=10 and Z=15 in the machine coordinate system.

Important: 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 your system installer.

Normally the control displays the current position of the axes in respect to the active work coordinate system. The control can display the position in the machine coordinate system if you select the absolute screen as described 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___Z___;
```

The X 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 wheel travels to this position in G00 or G01 mode depending on which is active when the control executes the G53 block. Any attempt to execute a G53 block in G02 or G03 mode generates 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 coordinate system that was in effect prior to the G53 blocks execution.

Important: The control must be in absolute mode (G90) when it executes the G53 block. If you execute a G53 while in incremental mode (G91), the control generates the error “G53 NOT ALLOWED IN INCREMENTAL MODE” and the block is not executed.

Example 11.1
Motion in the Machine Coordinate System

Program block	Comment
`N1 G00X30Z30;`	axis motion in work coordinate system.
`N2 G53X25Z10;`	axis motion in machine coordinate system.
`N3 X20Z50;`	axis motion in work coordinate system.

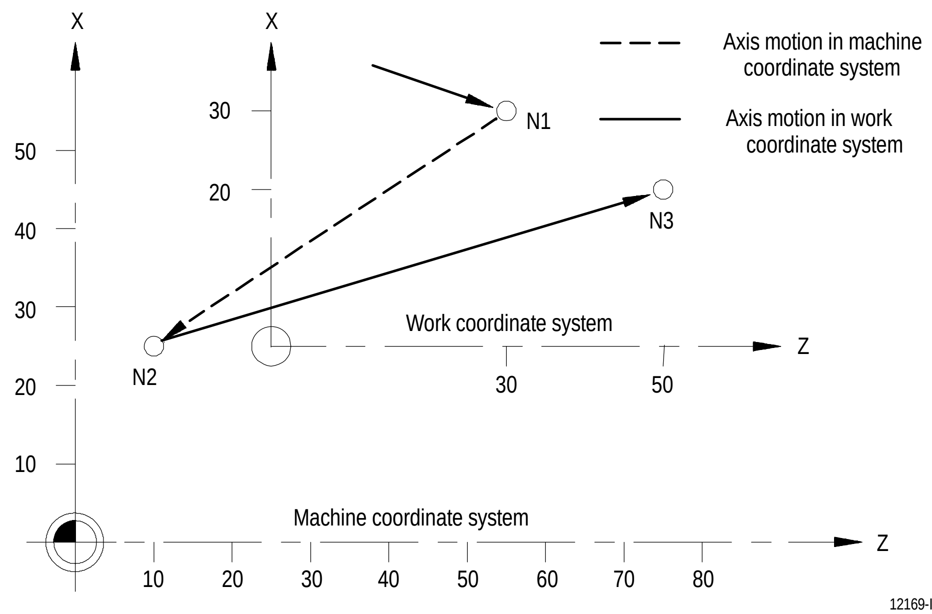


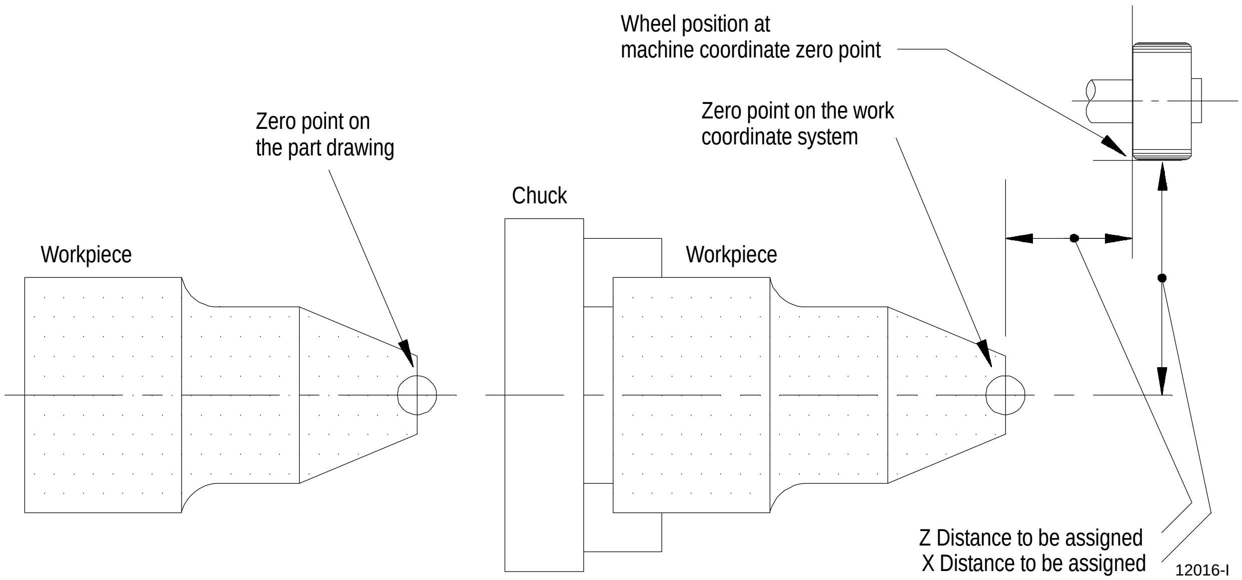
Figure 11.2
Results of Example 11.1

11.2 Preset Work Coordinate Systems (G54-59.3)

When grinding a part using a part program made from a part drawing, it is sometimes 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



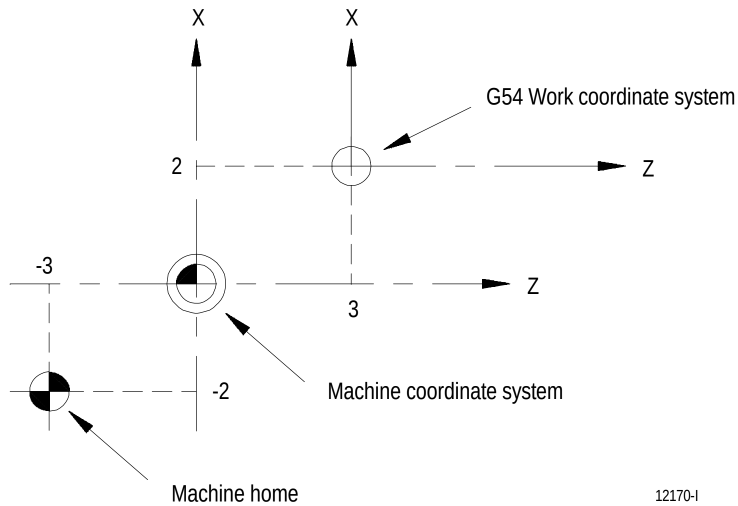
There are 9 preset work coordinate systems selectable using G54 - G59.3. You can select the required work coordinate system by specifying any of these G codes in the program.

Work coordinate systems called out by G54 - G59.3 have zero points that are entered in a work coordinate system table (see page 3-18). These zero points are in the form of offset values from the machine coordinate system zero point.

The control establishes the machine coordinate system immediately after it completes the machine home operation. The default work coordinate system, determined in AMP by your system installer, is activated simultaneously. The default work coordinate system is activated upon execution of a control reset operation, G92.1, or power-up. E-STOP reset can also activate the default system if your system installer has AMPed the control to perform a control reset at E-STOP reset.

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 in AMP as “none,” the default work coordinate system is the same as the machine coordinate system. This manual assumes G54 is the default coordinate system and no external offset has been entered.

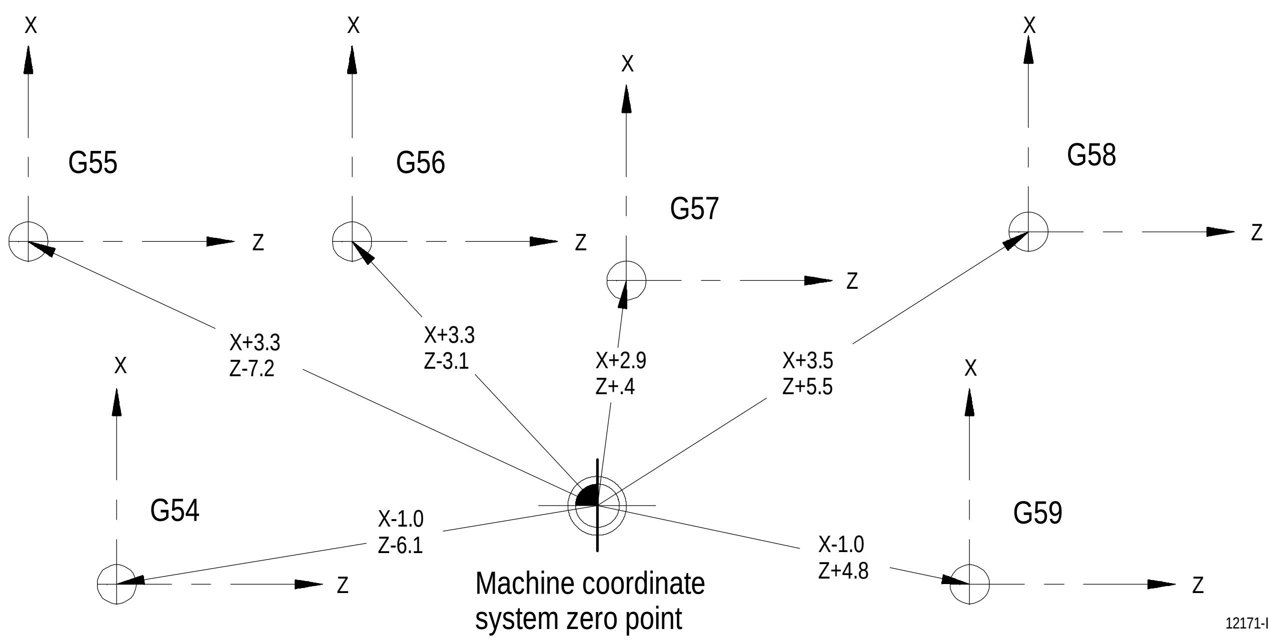
Figure 11.4
Work Coordinate System Definition



In Figure 11.4, the machine coordinate system was defined by declaring the fixed position machine home as the point X=-2., Z=-3. Then the G54 work coordinate system zero point is defined by the coordinates X=2, Z=3 in the machine coordinate system.

Coordinate positions in a part program are manipulated as coordinate values in the default work coordinate system unless another coordinate system is selected by programming G54-G59.3.

Figure 11.5
Examples of Work Coordinate System Definition

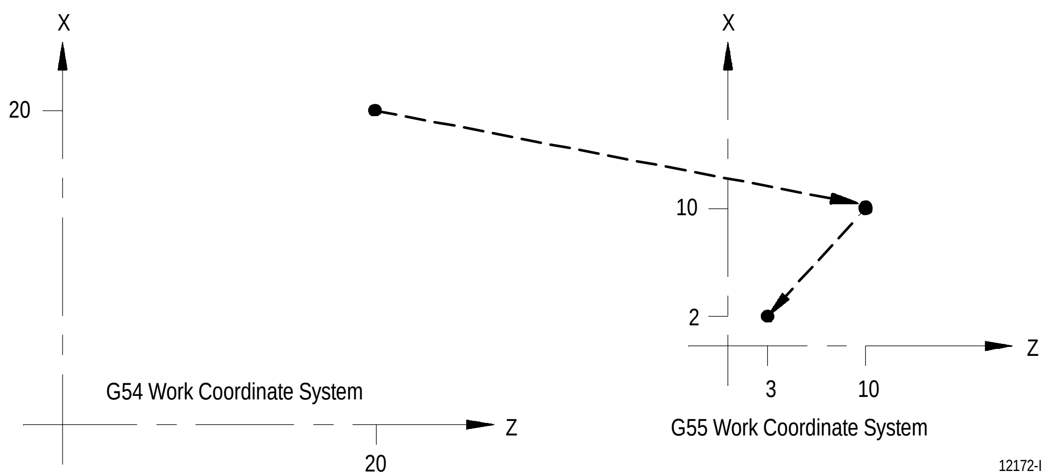


To change work coordinate systems, specify the G code corresponding to the work coordinate system in a program block. Any axis motion commands in a block that contains a change from one work coordinate system to another are executed in the work coordinate system specified in that block.

Example 11.2
Changing Work Coordinate Systems

Program Block	Comment
`G54;`	
`G00X20.Z20.;`	axis motion in the G54 work coordinate system.
`G55X10.Z10.;`	axis motion to the point X10. Z10. in the G55 work coordinate system.
`X2.Z3.;`	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 3 methods to change the value of a work coordinate system zero point in the work coordinate system table. You can find two methods in these chapters:

Method:	Chapter:
Manually alter the work coordinate system table	3
Alter the paramacro system parameter values 5221 - 5386	20

The third method, the one described in this chapter, alters the work coordinate system table through G10 programming. Changing the values in the table by 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:

```
G10 L2 P__ O__ X__ Z__;
```

Where :	Tell(s) the control:
L2	that you want to alter the coordinate system tables.
P__	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
O__	whether the value entered for the diameter axis is a radius or diameter value. (O is non-modal, and it applies to cylindrical grinders only) O1=value entered for the diameter axis is a radius value. O2=value entered for the diameter axis is a diameter value. Important: If you program O1 or O2 in a G10 code, the G10 code is not affected by a previously programmed G07 or G08 (radius/diameter programming). However, if no O code is specified, or if the O code is out of range (for example, O3), then the G10 code is affected by a G07/G08.
X_Z_	the location of the zero point of the specified work coordinate system relative to machine coordinate system.

Important: You cannot program G10 blocks when dresser/wheel radius compensation 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.

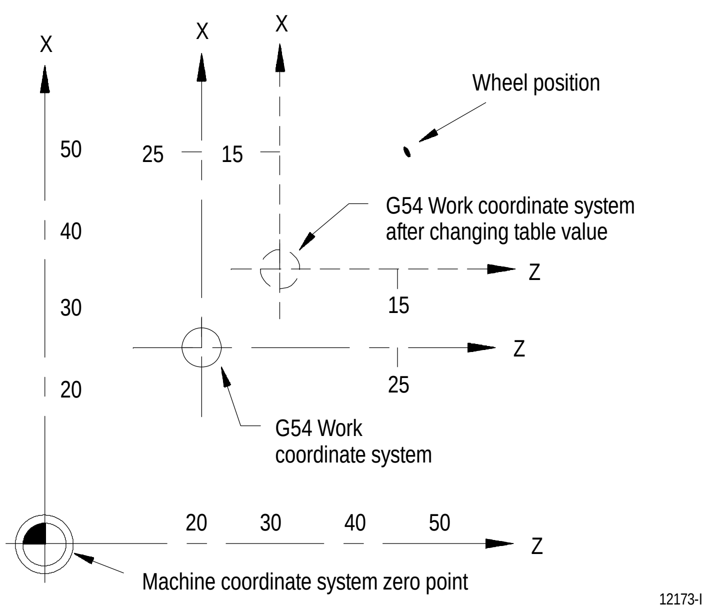
Example 11.3 and Figure 11.7 illustrate how the work coordinate system is shifted using G10.

Example 11.3
Work Coordinate System Shift Using G10

Program block	Work coordinate Position	Absolute coordinate Position
`G54G01X25.Z25.;`	X25 Z25	X50 Z45
`G91;`		
`G10L2P1O2X10.Z10.;`	X15 Z15	X50 Z45
or		
`G54G01X25.Z25.;`	X25 Z25	X50 Z45
`G90;`		
`G10L2P1O2X35.Z30.;`	X15 Z15	X50 Z45

Important: This modification is permanent. The control saves the new table values for the work coordinate systems even when control power is turned off.

Figure 11.7
Results of Example 11.3

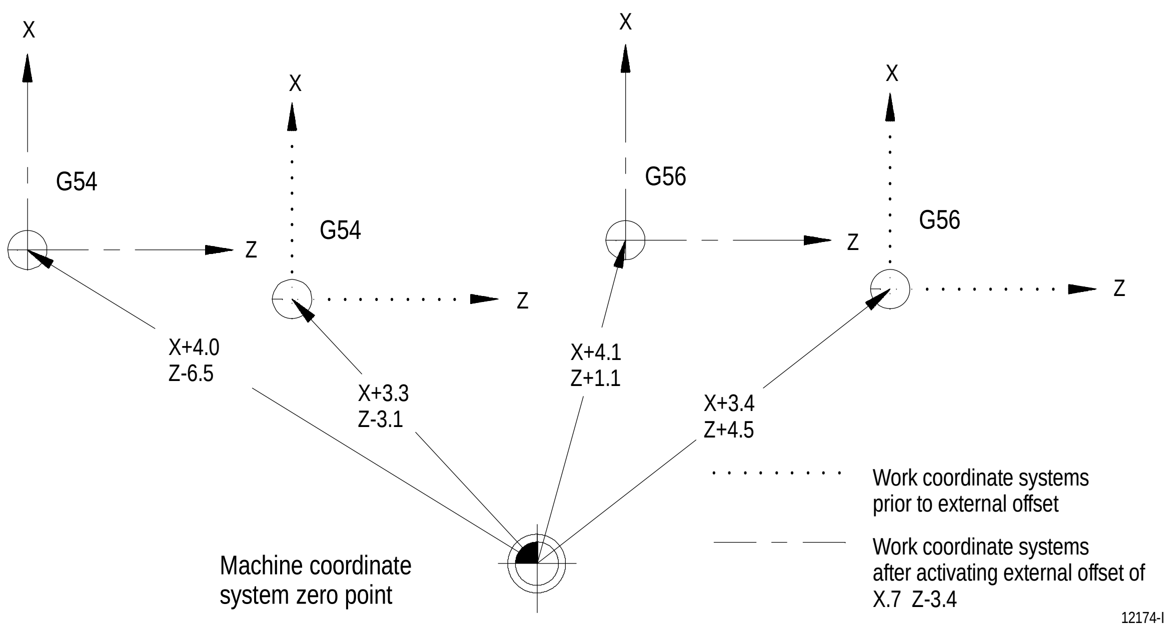


11.3 Work Coordinate System External Offset

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 chuck or mandrel is installed.

Also, use the external offset 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 dependent.

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 (unless “none” is selected as the default coordinate system).

11.3.1 Altering External Offset (G10L2)

There are 3 methods to change the value of an external offset in the work coordinate system table. Two methods can be found in the following chapters:

Method:	Chapter:
Manually alter the external offset value in the work coordinate system table	3
Alter the paramacro system parameter values 5201 - 5206	20

The third method, the one described in this chapter, 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 all work coordinate systems 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:

```
G10 L2 P0 O__ X__ Z__;
```

Where :	Tell(s) the Control:
`L2`	that you want to alter the coordinate system tables.
`P0`	the external offset is the offset to update.
`O__`	whether the value entered for the diameter axis is a radius or diameter value. (O is non-modal and it applies to cylindrical grinders only) `O1`=value entered for the diameter axis is a radius value. `O2`=value entered for the diameter axis is a diameter value. Important: If you program O1 or O2 in a G10 code, the G10 code is not affected by a previously programmed G07 or G08 (radius/diameter programming). However, if no O code is specified, or if the O code is out of range (for example, O3), then the G10 code is affected by a G07/G08.
`X_Z_`	the location of the zero point of the specified work coordinate system relative to machine coordinate system.

Important: G10 blocks cannot be programmed when dresser/wheel radius compensation is active.

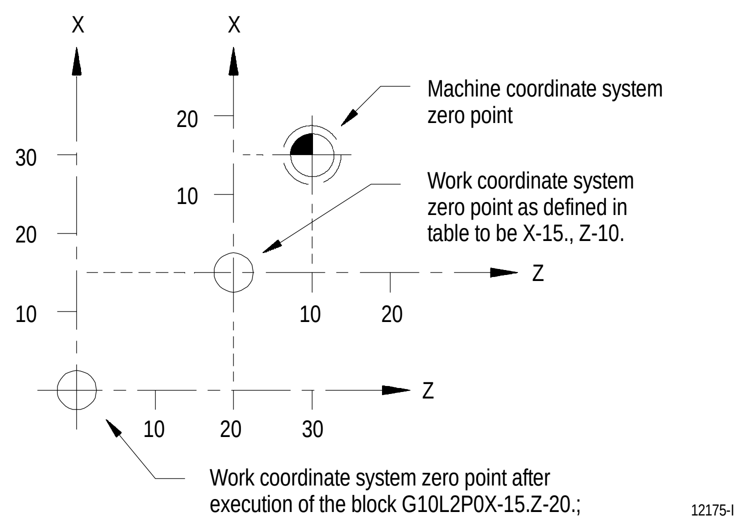
Example 11.4 and Figure 11.9 illustrate how the work coordinate system is shifted using G10.

Example 11.4
Changing the External Offset Through G10 Programming

Program Block	Comments
`G10L2P1O1X-15.Z-10.;`	defines work coordinate system zero point to be at X-15, Z-10 from the machine coordinate system zero point
`G90;` `G10L2P0O1X-15.Z-20.;`	sets external offset of X-15, Z-20 moving work coordinate system zero point to be at X-30, Z-30 from the machine coordinate system zero point
`G90;` `G10L2P0O1X-30.Z-30.;`	sets external offset of X-30, Z-30 moving work coordinate system zero point to be at X-30, Z-30 from the machine coordinate system zero point

Important: This modification is permanent. The control saves the new table values for the work coordinate systems even when control power is turned off.

Figure 11.9
Results of Example 11.4



11.4 Offsetting the Work Coordinate Systems

This section describes the more temporary ways of offsetting the work coordinate systems. These offsets are activated through programming and are canceled when a control reset is performed, or power to the control is turned off. These offsets can also be canceled when the control executes an end of program command (M02, M30, or M99) if your system installer has selected this option in AMP.

This section contains the following subsections:

Important: All of the offsets described in this section are global in nature. This means that they apply to all of the work coordinate systems. When changing work coordinate systems (programming G54-G59), you must consider the effects of these offsets on the new work coordinate system.

Wheel geometry offsets are not affected by an offset made to the work coordinate system.

Important: We recommend that wheel length offsets be canceled before any work coordinate system offsets are executed. If wheel length offsets are not canceled, the work coordinate system offset is added to the active wheel length offset. This can cause confusion when changing wheel length offsets later in the program (see page 13-4 on canceling wheel length offsets).

11.4.1 Coordinate Offset Using Wheel Position (G92)

Use the G92 command in a part program to offset the currently active work coordinate system relative to the current wheel position. A G92 block in a program offsets the zero point of the work coordinate system a specified distance from the current wheel position.

G92.2 cancels G92 without canceling any other work coordinates. This differs from G92.1, which cancels all coordinate system offsets.

When the control executes a G92 command in a program, it cancels any other active work coordinate system offsets including G52 offsets, jogged offsets, and set zero offsets. The external offsets are not affected.

Important: Wheel length offsets are not automatically canceled when a G92 block is executed. This can result in undesired effects on the work coordinate system when wheel offsets are changed later.

The following G92 block offsets the work coordinate system so that the current wheel position takes on the coordinate values programmed in the G92 block.

```
G92 X___ Z___;
```

For example, specifying values of zero for all axes in a G92 block causes the current wheel 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 is not offset, and the current coordinate position for that axis remains unchanged.

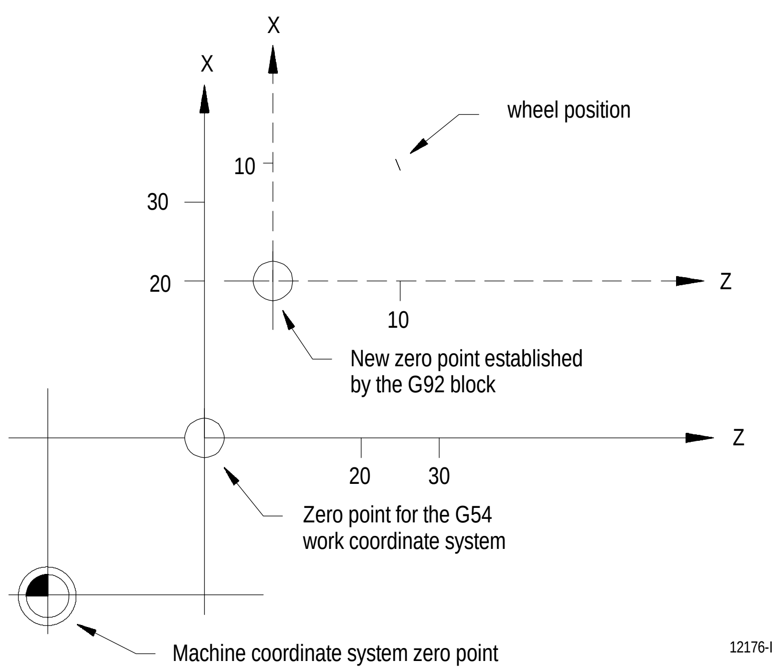
Once the work coordinate system is offset, all absolute positioning commands in the program are executed as coordinate values in the offset coordinate system.

Example 11.5
Work Coordinate System Offset (G92)

Program Block	Comment
`G54 G00;`	G54 work coordinate system.
`X35. Z25.;`	Rapid move to X35, Z25 in the G54 work coordinate system.
`G92X10.Z10.;`	Redefines current axis position to have the coordinates X10, Z10

The zero point of the offset G54 work coordinate system is 10 units away from the current wheel location in both the X and Z directions. If the Z value had not been entered in the G92 block, the Z coordinate location would have remained unchanged (Z25).

Figure 11.10
Results of Example 11.5



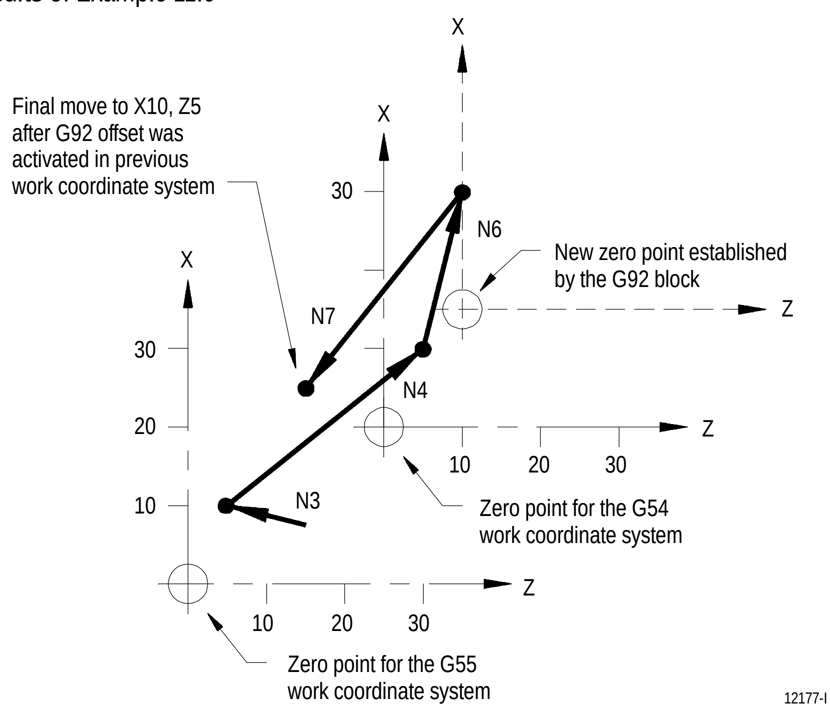
ATTENTION: G92 offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not program a change in coordinate systems (G54-G59.3) unless the effects of the offset have been considered.

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 G10L2P1X0Z0;`	Define G54 work coordinate system zero point to be positioned X0, Z0 away from the machine coordinate system
`N2 G10L2P2X20.Z25.;`	Define G55 work coordinate system zero point to be positioned X20, Z25 away from the machine coordinate system
`N3 G55X10.Z5.;`	Move to X10, Z5 in the G55 work coordinate system
`N4 G54X10.Z5.;`	Move to X10, Z5 in the G54 work coordinate system
`N5 G92X-5.Z-5.;`	Offset current wheel position to be at X-5, Z-5
`N6 X15.Z0.;`	Move to X15, Z0 (offset still active)
`N7 G55X10.Z5.;`	Move back to X10, Z5 in the G55 work coordinate system with the G92 offset still active

Figure 11.11
Results of Example 11.6



In Example 11.6, the G92 offset, entered while the G54 work coordinate system was active, has also shifted the G55 coordinate system. Any offsets described in this section alter all of the work coordinate systems (G54 - G59) 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___ Z___ ;
```

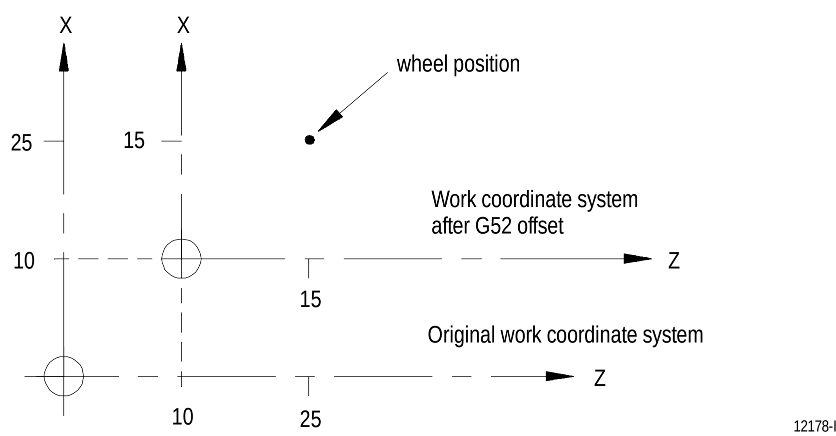
The above command offsets the current work coordinate system by the axis values that follow the G52 command.

Example 11.7
Work Coordinate System Offset by G52

Program Block	Machine Coordinate Position	Work Coordinate Position
G01X25.Z25.;	X25 Z25	X25 Z25
G52X10.Z10.;	X25 Z25	X15 Z15

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.

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 canceled. 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 can be activated after the G92 block is executed even if a G92 offset is still in effect.

ATTENTION: G52 offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not program a change in coordinate systems (G54-G59.3) unless the effects of the offset have been considered.

11.4.3 {SET ZERO} Offset

When you perform a set zero operation, the control shifts the current work coordinate system so that the current wheel position is the zero point of the coordinate system. The axis that set zero is performed on is selected through PAL (see your system installer's 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 Z0 block, with one exception. Unlike a G92, the set zero does <u>not</u> cancel a G52 offset. The G52 remains active and continues to offset the current wheel position in the work coordinate system. When the G52 offset is canceled later, the coordinate system shifts.

You can cancel the set zero offset by executing a G92.1, a control reset operation, an E-STOP reset operation, or by programming an end of program M30 or M02 command.

ATTENTION: Set zero offsets are global. This means that changing from one coordinate system to another does not cancel the offset. Do not program 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

You can perform the set zero offset through an optional switch installed by your system installer or through the {SET ZERO} softkey as described on page 2-11 on using the {FRONT PANEL} softkey.

11.4.4 Jog Offset

You can equip your control with an optional jog offset feature. The jog offset feature allows the operator to manually create an offset by jogging the axes. When this function is active, all jog moves made are added as offsets to the current work coordinate system.

Important: This feature functions only if your system installer has supplied the appropriate PAL programming. See the "Jog Offsets" and "Jog-on-the-fly" PAL flags in your PAL reference manual or the documentation supplied by your system installer.

Normally, jogging occurs in the manual mode. Your system installer has the option to enable a "Jog on the Fly" feature (through PAL) that allows jogging in automatic or MDI modes for the purpose of jogging an offset or performing manual gap elimination. See page 4-6 for details on manual gap elimination. To jog in an offset in automatic or MDI mode, both the "Jog on the Fly" and jog offset features must be active. See documentation provided by your system installer for details on activating these features. You can jog an offset at any time during automatic or MDI operation, even while blocks are executing.

To use this feature, follow these directions:

1. Turn on the switch to activate the jog offset function (see documentation provided by your 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, see documentation prepared by your system installer.

3. Jog the axis using any of the available jog types, with the exception of homing and arbitrary angle jog. 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 offset move is made, the axis position displays do not change on the screen (unless the currently active screen is displaying absolute position coordinates). This is because the coordinate values in the work coordinate system are being offset as the axes are being jogged.

You can cancel the jog offset by programming a G92.1, G92.2, executing a control reset operation, executing an E-STOP reset operation, or programming an end of program (M30 or M02, or M99 command).

11.4.5 Canceling Coordinate System Offsets (G92.1)

The G92.1 command cancels these offsets:

- G92 work coordinate system offset
- G52 zero point offset
- {SET ZERO} offset
- Jog offset
- Reset G54 - G59.3 coordinate systems to default conditions

G92.1 does not cancel an external offset (see page 11-10).

The G92.1 block also re-establishes the default work coordinate system as set in AMP by your system installer. It cancels or activates the coordinate system (G54-G59.3) as set in AMP to establish the default coordinate system.

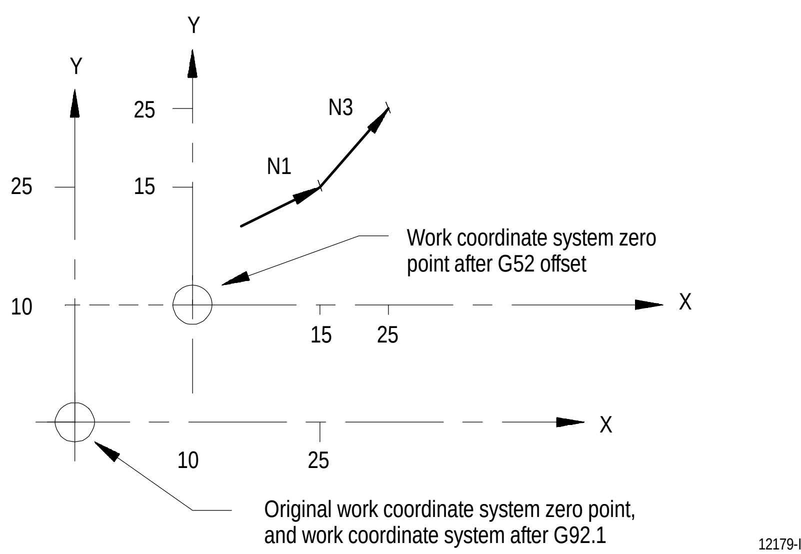
You must program the G92.1 block with no axis words. Axis words in a G92.1 block generates an error. When the control executes the G92.1 block, the control cancels all G92, G52, {SET ZERO}, and Jog offsets on all axes. You cannot cancel the offsets on individual axes.

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.

Example 11.9
G52 Offset Canceled 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 canceled, 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 these offsets:

- G92 work coordinate system offset
- {SET ZERO} offset
- Jog offset

G92.2 does not cancel an external offset (see page 11-10), reset the current work coordinate system (G54-G59.3) or cancel a G52 offset.

You must program the G92.2 block with no axis words. Axis words in a G92.2 block generate an error. When the control executes the G92.2 block, the control cancels all G92, {SET ZERO}, and Jog offsets on all axes. You cannot cancel the offsets on individual 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

Your system installer has the option to activate, deactivate, or alter the value of these offsets through PAL:

- Work coordinate systems
- External offset
- Wheel length offsets
- Dresser/wheel radius offsets
- Wheel orientation

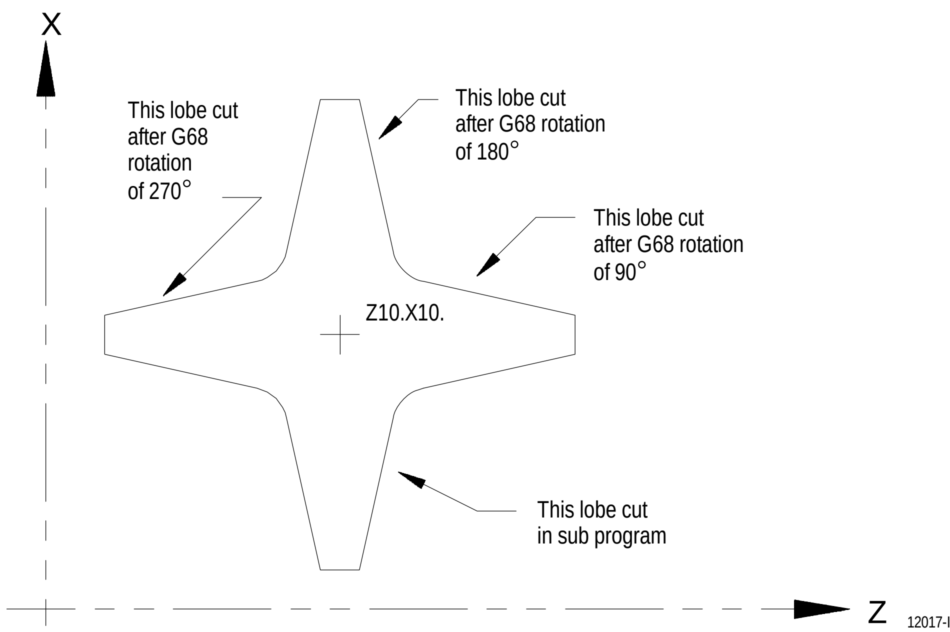
Your system installer can modify these offsets through a PAL display page created by your system installer or through some other input to PAL.

Be aware that there can 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 control's look-ahead buffer use the newly modified offset value. However, if a dresser/wheel radius compensation offset has been modified by PAL, the control does not update the look-ahead buffer. See documentation supplied by your system installer for details on specific PAL offset operations.

11.6 Rotating the Coordinate Systems

The control has a feature (G68) that can rotate the work coordinate system. Another feature, external part rotation, rotates all work coordinate systems by simulating a rotation of the machine coordinate system. Rotating the coordinate systems can prove to be useful when grinding a part that contains symmetrical geometries (see Figure 11.14).

Figure 11.14
Part with Symmetrical Geometry



Important: Reciprocation is not permitted on any axis that is rotated. You cannot:

- perform G68 work coordinate rotation and external part rotation on a reciprocating axis
- rotate the work coordinate system of a plane in which one of the axes assigned to that plane is the reciprocating axis
- start reciprocation on one of the axes assigned to the plane in which you rotated the work coordinate system

11.6.1 Rotating the Current Work Coordinate System (G68, G69)

To rotate the current work coordinate system, program this command:

```
G68 X__ Z__ R__;
```

Where :	Is:
X, Z	the center of rotation using only the two 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	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 on page 11-30.

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 11.10). The first motion command following a G68 block cannot be a circular move.

The control ignores any unnecessary parameters in the G68 block. 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 results in an error.

The G68 block does 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 rotates only the currently active work coordinate system. When changing to a different work coordinate system, the rotation does not apply to the new work coordinate system. When changing back to the rotated coordinate system, the rotation is still 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 11-30. 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.

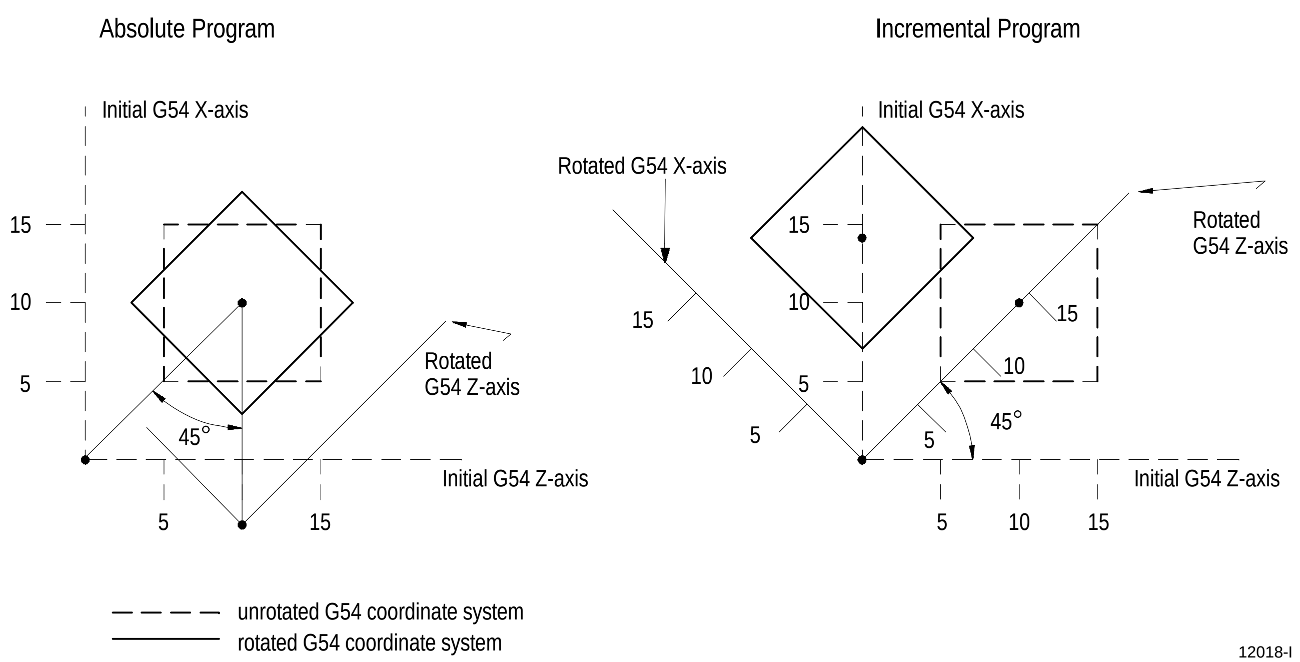
Example 11.10
Rotating the Current Work Coordinate System

These program blocks cause the rotation of the active work coordinate system as shown in Figure 11.15.

ABSOLUTE PROGRAM	INCREMENTAL PROGRAM
`N1 G54 G17 G00;`	`N1 G54 G17 G90;`
`N2 G90 Z0. X0. F500;`	`N2 G00 Z0. X0.;`
`/N3 G68 Z10 X10 R45;`	`/N3 G68 Z10 X10 R45;`
`N4 G90 G00 Z5. X5.;`	`N4 G91 G00 Z5. X5.;`
`N5 G01 Z15. F100;`	`N5 G01 Z10 F100;`
`N6 X15.;`	`N6 X10;`
`N7 Z5.;`	`N7 Z-10;`
`N8 X5.;`	`N8 X-10;`
`N9 M30;`	`N9 G69;`
	`N10 M30;G54 G00;`

If optional block delete 1 is set to “ON,” the control follows the path shown with a dashed line in Figure 11.15. If optional block 1 is set to “OFF,” the control follows the path shown with a solid line in Figure 11.15.

Figure 11.15
Results of Example 11.10



In Figure 11.15, 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: You can rotate all of the work coordinate systems at once by using the external part rotation, see page 11-28.

If rotating the coordinate system again in the same plane using another G68 command, the angle of rotation is taken from the current rotated coordinate position, not the original position (see Figure 11.16).

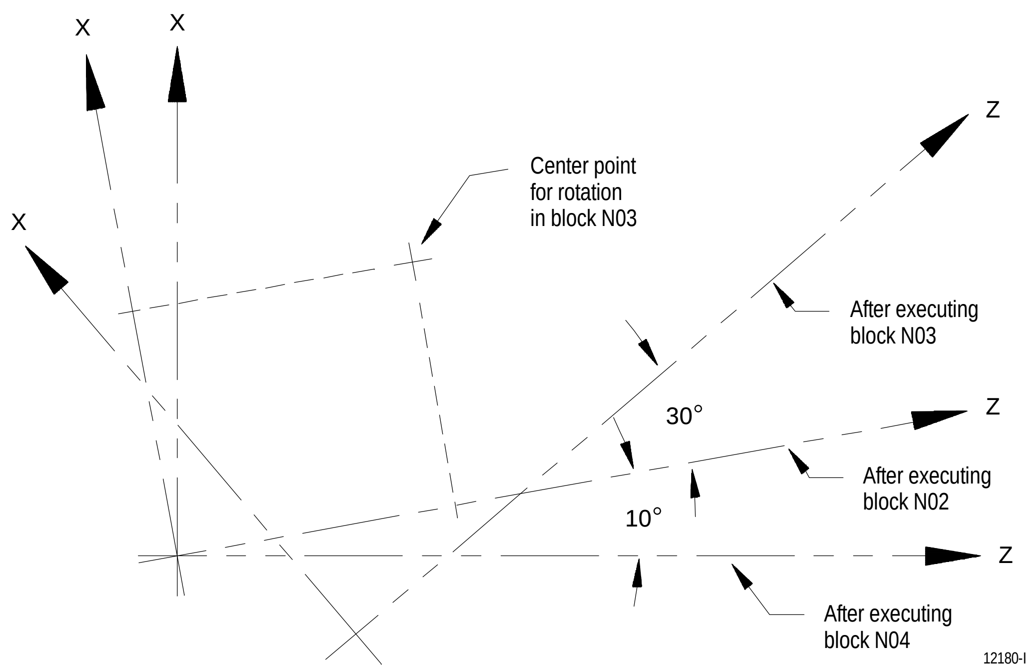
Rotating the coordinate system again in a different plane using another G68 is allowed. The resulting work coordinate system is rotated in both planes.

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 11.11
Canceling G68 Rotations With G69

Program Block	Comment
`N01 G54;`	
`N02 G68Z0X0R10;`	Rotates the current work coordinate system 10°.
`N03 G68Z5.X4.R30;`	Rotates the current work coordinate system 30° about a center point of Z5., X4. for a total rotation from its original position of 40°.
`N04 G69;`	Returns the work coordinate system to its original position of 0°.

Figure 11.16
Results of Example 11.11

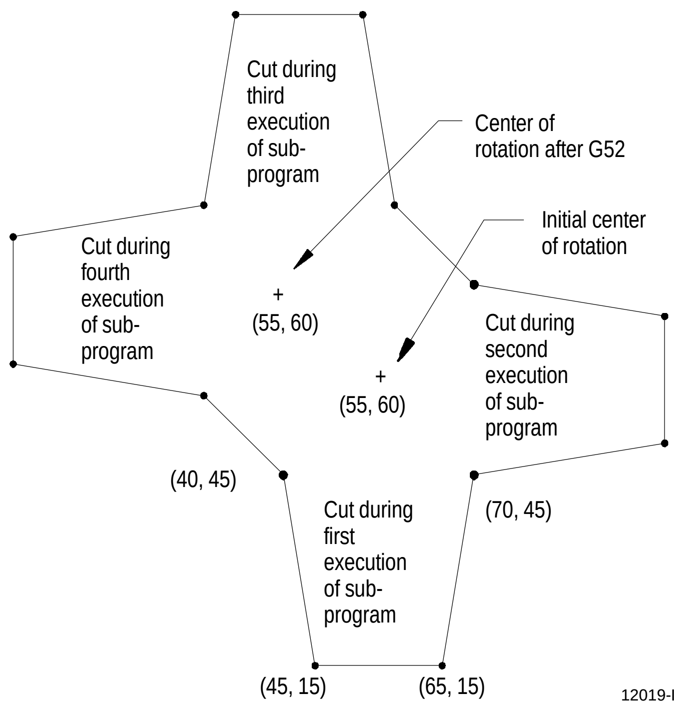


Rotating the work coordinate system can be helpful anytime a part has a repetitive shape. This feature combined with the G52 work coordinate system shift can reduce the size of a part program appreciably. Example 11.12 is an example of this.

Example 11.12
Rotating the Work Coordinate System with G52

Main Program	Subprogram 1000
`G17 G90 G00 Z0 X0;`	`G01 Z45. X15. F500.;`
`G00 G90 Z40 X45.;`	`Z65.;`
`M98 P1000 L4;`	`Z70. X45.;`
`M30;`	`G68 Z55. X60. R90.;`
	`M99;`

Figure 11.17
Results of Example 11.12



11.6.2 External Part Rotation

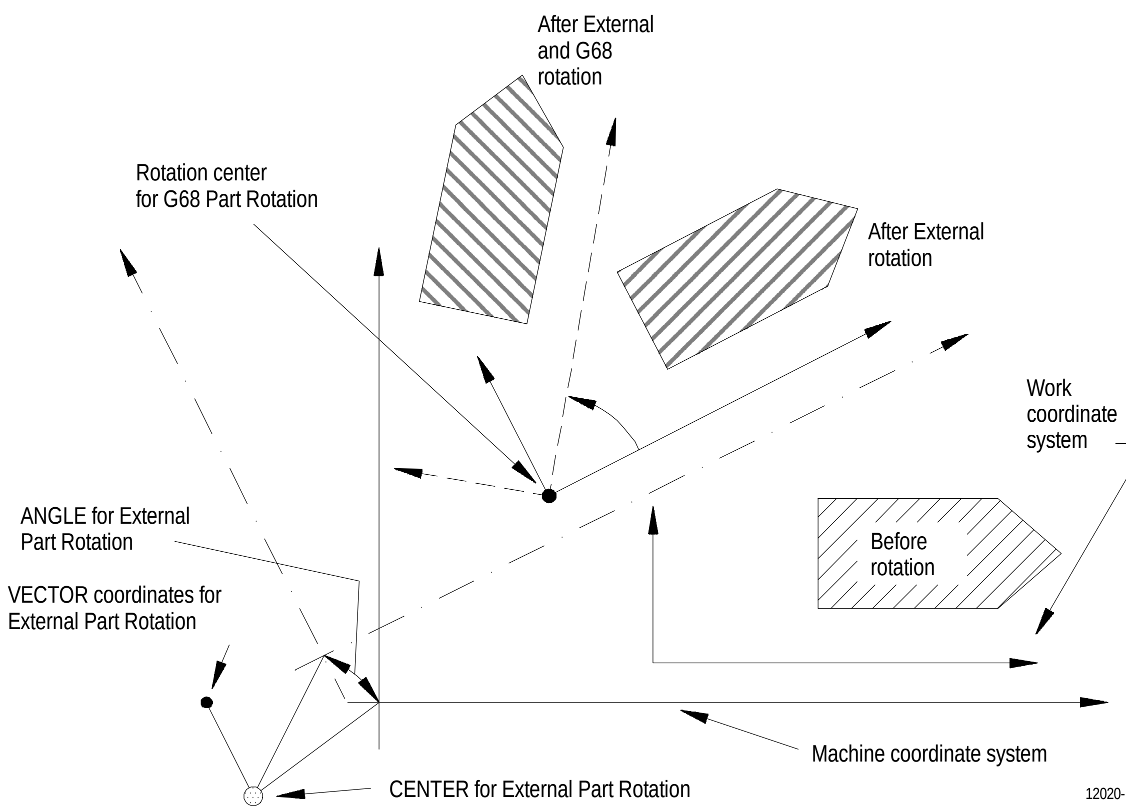
The external part rotation feature simulates a rotation of the machine coordinate system. Since all work coordinate systems are referenced from the machine coordinate system, rotating it would effectively rotate all work coordinate systems. However, software overtravels, programmable zone limits, homing, 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, see page 11-24. If a G68 is programmed to rotate the current work coordinate system, an additional rotation of coordinates result as shown in Figure 11.18.

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 an external part rotation. If you want to interrupt a program to perform an external part rotation, the rotation does not become effective until the end of program (M02 or M30) command is read, a control reset, or E-STOP reset is performed.

Figure 11.18
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 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	PRGRAM CHECK	SYSTEM SUPORT	→

FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the {COORD ROTATE} softkey.

```
(softkey level 2)
```

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→

↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→

The external part rotation parameters screen appears:

```
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 parameter you want to change by pressing the up, down, left, right cursor keys. The selected parameter appears in reverse video.

4. Enter the new value for the parameter by using the keys on the keyboard. The entered value appears on the input line (lines 2 and 3) of the CRT. You can modify the value on the input line as described in chapter 2.

5. When the value appears on the input line of the CRT, press the [TRANSMIT] key. The value on the input line replaces 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 toggles between on and off each time you press this softkey.

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 is rotated as described by the external part rotation parameters.

The “PROGRAM” and “TARGET” position displays (as described in chapter 8) do 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 reflect an external part rotation.

External Part Rotation Parameters

PLANE

Use this parameter to select a plane for rotation. 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.

CENTER

Use this parameter to assign the center of rotation. The center of rotation is a point on the machine coordinate system about which all the work coordinate systems are rotated. Enter a coordinate value for each axis in the selected plane. The default value for the center of rotation is (0, 0).

VECTOR

This parameter is optional, and is superseded by any entered value for the ANGLE parameter. If used, the values entered here define the coordinates of the "head" of a vector. The "tail" of this vector is the center of rotation (described above). The resulting angle between this vector line and the first plane axis (described above) is the angle that the work coordinate systems are to be rotated. Again, all work coordinate systems are rotated about the point defined by the CENTER parameter. The default values for VECTOR are (0, 0). If values are entered for Vector, the angle of rotation that is generated is displayed for the Angle parameter.

ANGLE

Use this parameter to assign the angle that the work coordinate systems rotates about the above defined CENTER point. If you enter a value here, the control cancels any value entered for the VECTOR parameter. The ANGLE parameter has units of degrees, with a default of zero. Positive angles result in rotation clockwise from the primary axis for that plane (as defined in plane above). If no value is entered, the control displays the current angle as defined by the Vector parameter here.

11.7 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 code.

This is especially true for surface grinding machines. Cylindrical grinders are generally limited to using the G18 plane only.

Some of the features that are plane dependant are:

- Circular interpolation
- Dresser/wheel radius compensation
- Most fixed cycle operations

Important: Your system installer determines the axis names, axis integrands, and parallel axes for each plane in AMP. Your system may not have planes assigned exactly as listed below. See the documentation prepared by your system installer.

Typical axis names and their corresponding plane assignment are shown in Table 12.A and Table 12.B (this manual assumes these configuration throughout):

Table 12.A
Typical Surface Grinder Plane Configuration

PLANE	1st Axis	2nd Axis	1st Integrand	2nd Integrand	1st Parallel	2nd Parallel
G17	X	Y	I	J	U	V
G18	Z	X	K	I	W	U
G19	Y	Z	J	K	V	W

Table 12.B
Typical Cylindrical Grinder Plane Configuration

PLANE	1st Axis	2nd Axis	1st Integrand	2nd Integrand	1st Parallel	2nd Parallel
G17	none	none	none	none	none	none
G18	Z	X	K	I	W	U
G19	none	none	none	none	none	none

You can alter planes to accommodate additional axes parallel to the principle axes by programming those axes in a G17, G18, or G19 block. See Example 11.13.

Example 11.13
Altering Planes for Parallel Axes

```
Assuming the system installer has made these assignments in AMP:

G18    -- the ZX plane
W axis -- parallel to Z axis
U axis -- parallel to X axis
```

Program Block	Plane Selected	Axis Motion
`G18;`	selects ZX plane	none
`G18 U;`	selects ZU plane	U axis moves to zero
`G18 W;`	selects WX plane	W axis moves to zero
`G18 UW;`	selects WU plane	U and W axes move to zero

This manual assumes that your system installer has selected the G18 plane to be activated when an end-of-program block is read (M02 or M30), when a control or E-STOP reset is performed, or when power to the control is turned off.

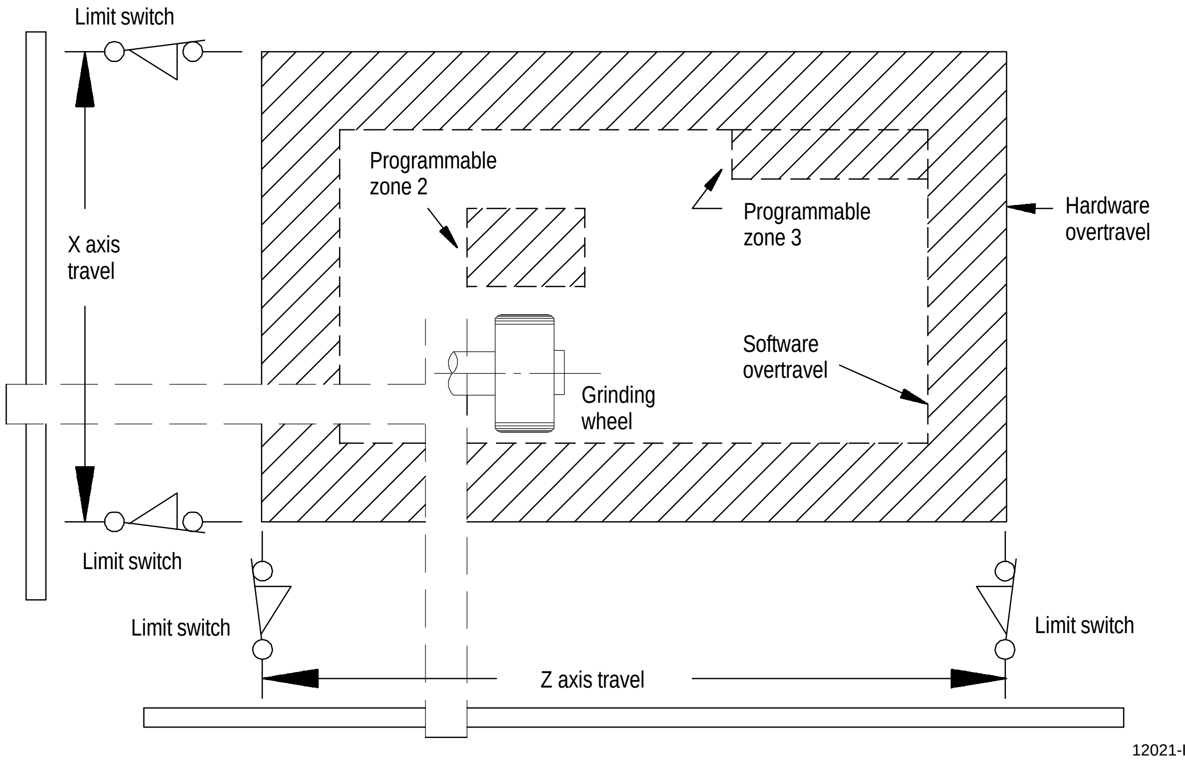
Important: Any axis word in a block with plane select G-codes (G17, G18, G19) causes axis motion on that axis. If no value is specified with that axis word, the control assumes a value of zero or generates an error, depending on how your system is AMPed.

11.8 Overtravels and Programmable Zones

This section contains the following subsections:

Overtravels and programmable zones define areas that restrict the movable range of the grinding wheel. The control is equipped to establish two overtravel areas and two programmable zones as illustrated in Figure 11.19.

Figure 11.19
Overtravel Areas and Programmable Zones



Two types of overtravels are available:

- Hardware overtravels –– Established by your system installer by mounting mechanical limit switches on the movable range of the axes.
- Software overtravels –– Established in AMP by your system installer by assigning coordinate values in the machine coordinate system.

Two types of Programmable Zones are available:

- 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 are 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 are turned on and off in the part program.

11.8.1 Hardware Overtravels

When the grinding machine is set up, your 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 your system installer, the limit switch trips. When the limit switch trips, axis motion stops. The area defined by these limit switches is referred to as the hardware overtravel.

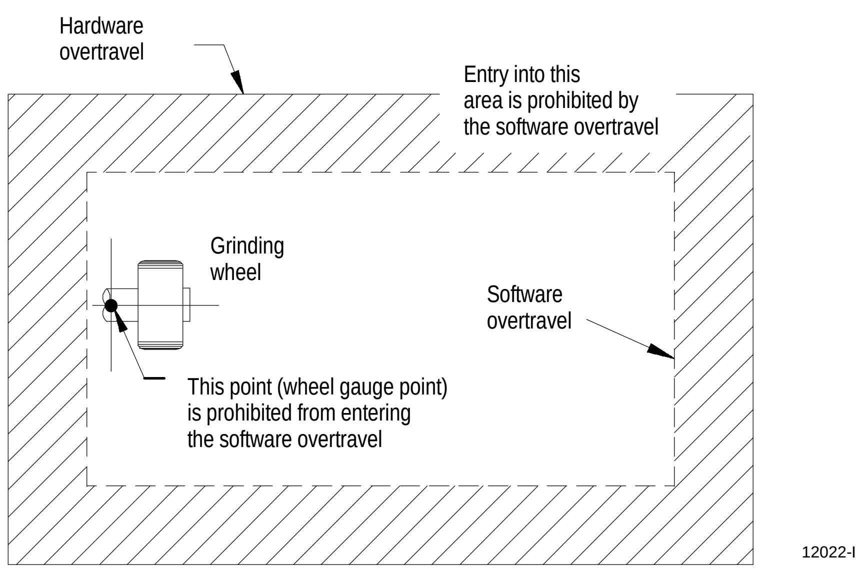
ATTENTION: The area defined by a hardware overtravel does not take into account any wheel offsets. This may allow the grinding wheel to enter a restricted area without the axes entering it.

Install these switches to prevent the machine from motion that exceeds a range that might cause damage to the machine. Frequently your system installer wires the hardware overtravel directly into the E-STOP string. This stops all motion and disable the axis drives. See the literature provided by your system installer for instructions on moving axes out of hardware overtravel.

11.8.2 Software Overtravels

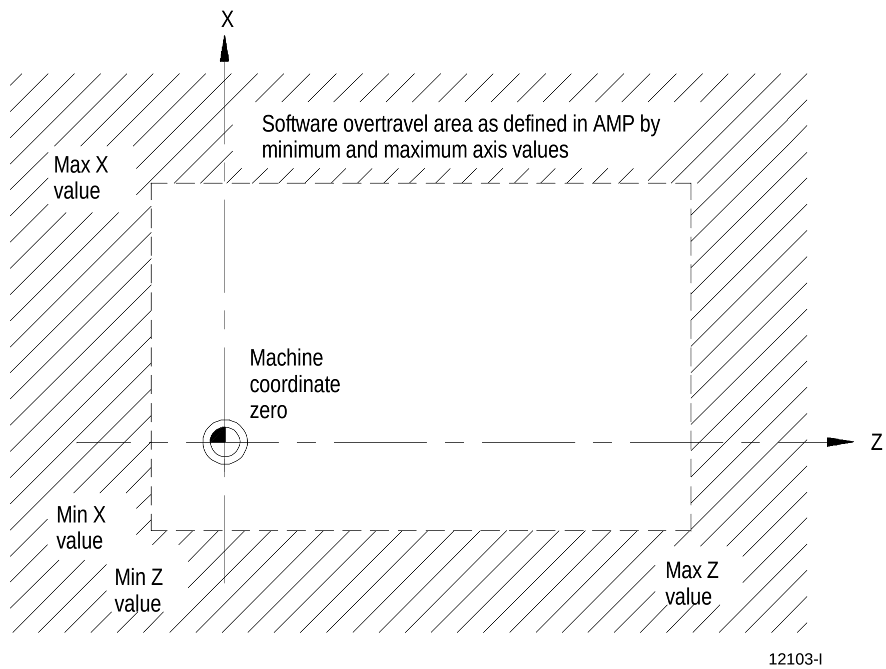
The coordinate values of the points defining the software overtravels are set in AMP by your system installer. This overtravel can be disabled only by your system installer in AMP. If your system installer has enabled the software overtravels, the control does not allow the axes to enter the area defined by the software overtravels.

Figure 11.20
Software Overtravels Established in AMP



Your 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. The 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 11.21
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 your 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 page 11-43.

ATTENTION: The area defined by a software overtravel does not take into account any wheel length or radius offsets. This may allow the grinding wheel to enter a restricted area without the axes entering it. Make sure this is considered when programming motions with wheel offsets active.

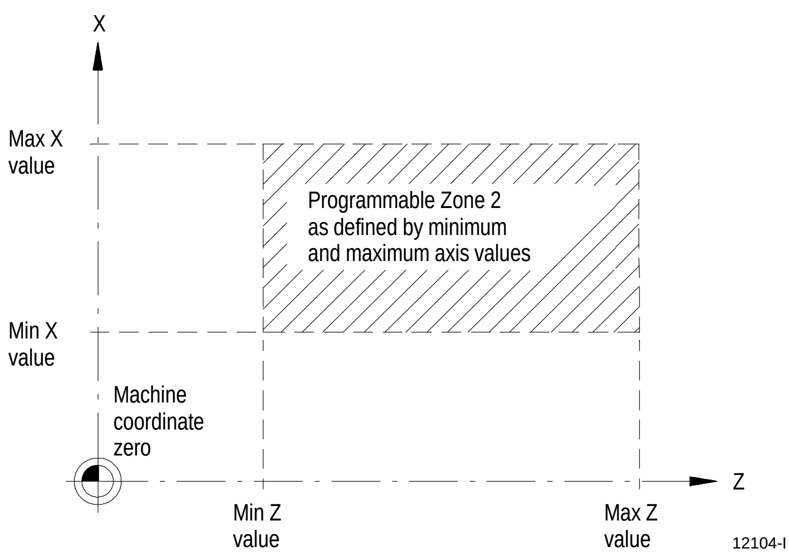
11.8.3 Programmable Zone 2 (G22, G23)

Programmable zone 2 defines an area which the grinding wheel cannot enter. Generally, you use zones 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 take into account any active wheel length offset but not wheel radius offsets.

Values for programmable zone 2 are entered in the programmable zone tables as described on page 3-25. These values represent a maximum and a minimum machine coordinate value for each axis. The area defined by these points establishes the boundaries for programmable zone 2.

Figure 11.22
Area Defining Programmable Zone 2



Programmable zones 2 and 3 become active when a G22 block is executed and are canceled when a G23 is executed. Both G22 and G23 are modal commands.

Important: When made active the current grinding wheel location must be outside of the area defined by programmable zone 2.

Important: Special programmable zone considerations must be made when using programmable zones on an angled-wheel grinder. Refer to page 14-18 for details.

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.

G23 is normally made active at power-up, though this is ultimately determined by your 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 are stored as the coordinates for programmable zone 3 (see page 11-40).

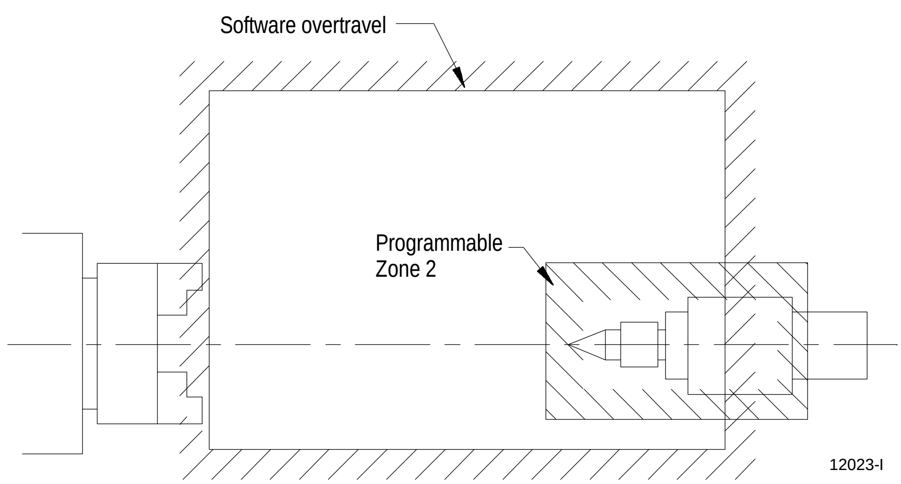
If an attempt is made to program some other command in a G22 or G23 block, other than a G code in the same modal group, for example:

```
G22 G01 X12.;
```

the control issues the error message:

"UNUSABLE WORDS IN ZONE BLOCK"

Figure 11.23
Programmable Zone 2



For details on how the control reacts to entry into a prohibited area, see page 11-43 on resetting overtravel.

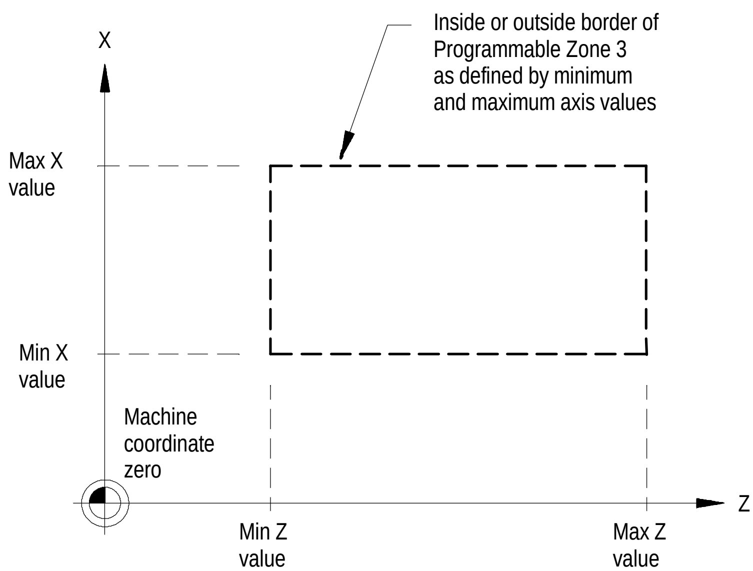
11.8.4 Programmable Zone 3 (G22.1, G23.1)

Programmable zone 3 can define an area which the grinding wheel cannot enter or cannot exit. This is determined by the current wheel location when programmable zone 3 is made active. Generally, you use zones 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 take into account any active wheel length offsets but not wheel radius offsets.

Values for programmable zone 3 are entered either in the programmable zone table (described on page 3-25) 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 11.24
Area Defining Programmable Zone 3



Unlike the software overtravels, which define the perimeter of an area from which the grinding wheel cannot exit, programmable zone 3 can define either an area that the grinding wheel:

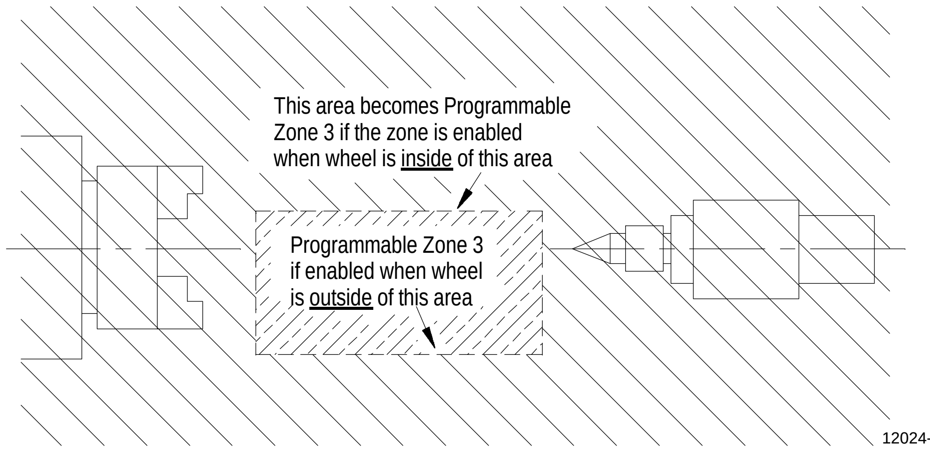
- cannot exit if the wheel is inside the zone when the zone is activated

  or

- cannot enter if the wheel is outside the zone when the zone is activated

This area is determined by the current wheel location when programmable zone 3 is made active.

Figure 11.25
Programmable Zone 3



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.

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.

You can reassign values for the parameters that establish programmable zone 3 by programming axis words in a G22 program block. Main axis words (normally X and Z) define maximum zone limits. Integrand axis words (normally I and K) define minimum zone limits. These axis words can vary. See documentation prepared by your system installer.

For example, the following block:

```
G22 X10 K2;
```

redefines the maximum X coordinate of programmable zone 3 at a value of 10, and the minimum Z coordinate of programmable zone 3 at a value of 2. Any unspecified axis parameters remain at their currently defined value. The coordinate 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 whether it is the minimum or maximum value that is incorrect.

Important: You must program G22.1, G23, and G23.1 in blocks without other commands. If programming a G22, the control stores any axis words included in the block as the coordinates for programmable zone 3.

If other commands are programmed in a G22, G22.1, G23, or G23.1 block, other than a G code in the same modal group, the control issues this error message:

"UNNECESSARY WORDS IN ZONE BLOCK"

For details on how the control reacts to entry into a overtravel area, see page 11-43.

11.8.5 Resetting Overtravels

The control stops grinding wheel travel 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 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 your system installer.
- programmable zone overtravel –– The wheel 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 appears:

If you see this error message:	It means that:
"HARDWARE OVERTRAVEL (-) AXIS (X)"	the specified axis has tripped either the + or - hardware limit switch mounted on the machine
"SOFTWARE OVERTRAVEL (+) AXIS (X)"	the specified axis has entered the overtravel area defined by the software overtravel limits in either a positive or negative direction
"VIOLATION OF ZONE (2) AXIS (X)"	the wheel 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 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. Eliminate any other possible conditions that can 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 4-2 except homing to manually move the axis away from the limit.

11.9 Absolute/Incremental Modes (G90, G91)

There are two methods for programming axis positioning commands:

- absolute positioning
- incremental positioning

In 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 moves the axes to a position X40, Z20 as referenced on the active coordinate system.

G90 is a modal G code and remains active until canceled by a G91.

In incremental mode, coordinates are referenced from the current axis position. Incremental mode is established by programming a G91.

```
G91X40.Z20.;
```

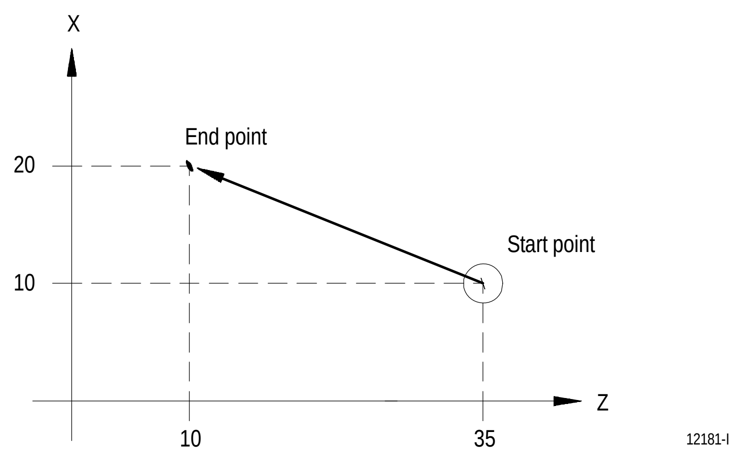
In the above block, the control moves the grinding wheel a distance of 40 units on the X axis and 20 units on the Z axis away from the current axis position.

G91 is a modal G code and remains active until canceled by a G90.

Example 11.14
Absolute vs. Incremental Commands

Absolute Command	Incremental Command
G90X20.Z10.;	G91X10.Z-25.;

Figure 11.26
Results of Example 11.14



11.10 Inch/Metric Modes (G70, G71)

You can program a G70 to select the inch system or a G71 to select the metric system. These unit system G codes should be among the first blocks written in a program.

Both G70 and G71 are modal, and cancel each other. The default unit system selected by the control at power-up, end of program (M02, M30, or M99) and control reset is determined in AMP by your 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 (G71 or G70) is displayed among the active system G codes.

Following is a list of some of the functions that 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)

11.11 Radius/Diameter Modes (G07, G08)

Usually workpieces on cylindrical grinders are cylindrical in shape. The control allows programming of workpiece dimensions as radius or diameter values. It also allows data to be entered into the offset tables as either radius or diameter values.

This feature is only available on cylindrical grinder configurations.

G08 places the control in diameter programming mode. This mode remains active until canceled by a G07.

G07 places the control in radius programming mode. This mode remains active until canceled by a G08.

Your system installer can select G08 or G07 in AMP to be the active mode at power-up. You can display the currently active G code by selecting the status screen.

Example 11.15
Diameter/Radius Programming

```
Assume X is the diameter axis
```

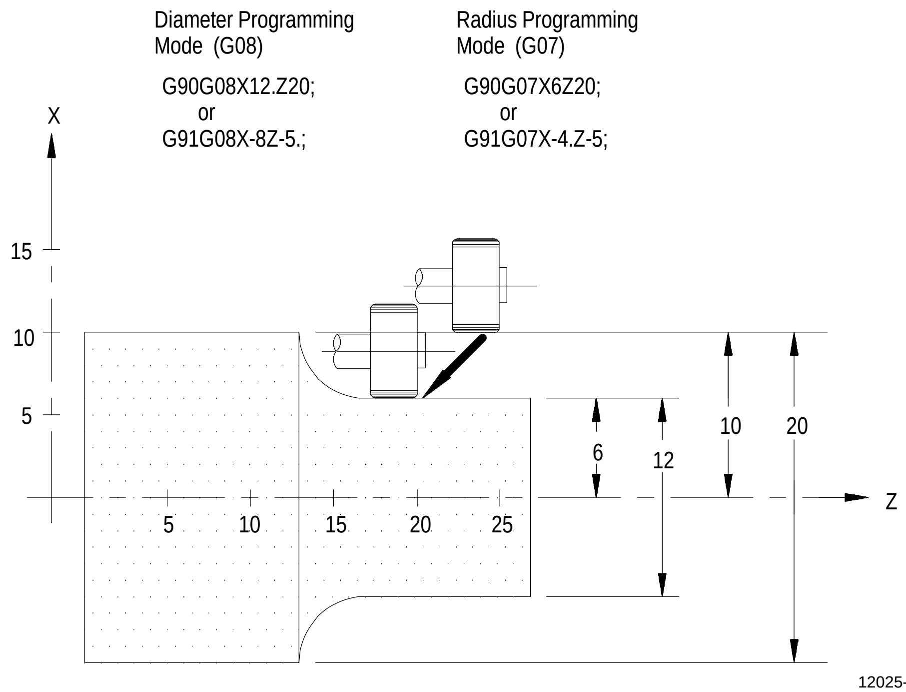
Diameter Programming Mode (G08)	Radius Programming Mode (G07)
Incremental	Incremental

`G90G00X0Z35;`
`G91G01F.1X12;`
`Z-10`
`G02Z-4X8I4;`
`G01Z-15;`
	`G90G00X0Z35;`
	`G91G01F.1X6;`
	`Z-10`
	`G02Z-4X4I4;`
	`G01Z-15;`

Absolute	Absolute

`G90G00X0Z35;`
`G01F.1X12;`
`Z25;`
`G02Z21X20I4;`
`G01Z6;`
	`G90G00X0Z35;`
	`G01F.1X6;`
	`Z25;`
	`G02Z21X10I4;`
	`G01Z6;`

Figure 11.27
Results of Example 11.15



Important: You must always program the following as radius value, regardless of whether G07 or G08 is active:

- Center point assignment (addresses R, I, K) for circular interpolation
- Feedrates in the X-axis direction (change in radius per revolution G95 or radius per minute G94)
- The threading cycle parameter E or F when face threading is being programmed

Position displays are also impacted by radius diameter mode. Diameter/radius axes (selected in AMP) are displayed with either an R or a D next to them indicating which mode is active and currently represented on the CRT. This even applies to the machine coordinate system (absolute display) and offset tables.

Refer to your system installers documentation for details on which axes on your system have been configured to allow both diameter and radius programming. This manual assumes only the X axis is configured to allow diameter and radius programming.

11.12 Scaling

This section contains the following subsections:

Use the scaling feature to reduce or enlarge a programmed shape. This feature is enabled by programming a G14.1 block as shown below:

```
G14.1 X__ Z__ P__;
```

Where :	tell(s) the control :
`X, and Z`	the axis or axes to be scaled and the center of scaling for those axes. Can be any valid axis name(s).
`P`	the scaling magnification factor for the specified axes.

The axes programmed in the G14.1 block determine which axes are scaled. The corresponding axis word values specify the center of scaling for each axis. This position is the axis coordinate around which the scaling operation is performed.

The scaling magnification factor (P) is the amount of scaling to be applied to the programmed axes. Each axis can 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 reduces a programmed move while a scale factor greater than one enlarges a programmed move.

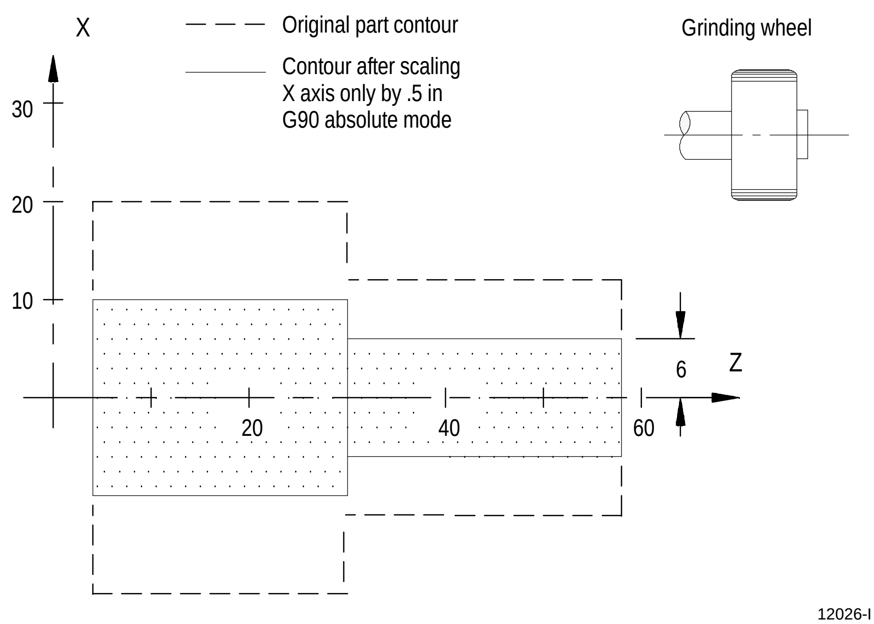
If no P word is programmed or if P0 is programmed in the G14.1 block, the control uses the default magnification factor. If the programmed P word value is out of range, an error message is displayed on the CRT.

When absolute mode (G90) is active, scaling moves are referenced from the programmed center of scaling.

Example 11.16
Scaling with Absolute Mode Active

Program block	Comment
`G07 G90 G00 X30. Z60.;`	radius mode, absolute mode
`G14.1 X0 P.5;`	scale X axis only, by .5
`G01 X12.;`	feedrate move X
`Z30.;`	feedrate move Z
`X20.;`	feedrate move X
`G14;`	cancel scaling
`G00 X30. Z60.;`	rapid return

Figure 11.28
Results of Example 11.16



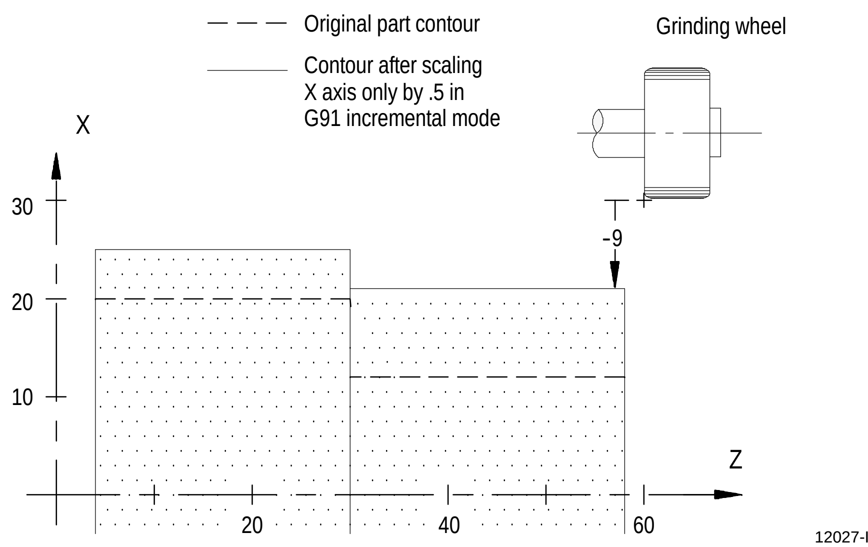
When incremental mode (G91) is active, the control ignores the programmed centers of scaling. The control performs scaling on the axes programmed in the G14.1 block, but the scaling moves are referenced from their current axis positions not the programmed center of scaling or the active coordinate zero point.

Important: The center of scaling can be specified in either incremental or absolute mode (G90/G91) in the G14.1 block. But unlike other features in the control, you cannot program both modes in the same block.

Example 11.17
Scaling with Incremental Mode Active

Program block	Comment
`G07 G90 G00 X30. Z60.;`	radius mode, absolute mode
`G91;`	incremental mode
`G14.1 X1.023 P.5;`	scale X by .5 (X value is ignored)
`G01 X-18.;`	feedrate move X
`Z-30.;`	feedrate move Z
`X8.;`	feedrate move X
`G14;`	cancel scaling
`G00 X5. Z30.;`	rapid return

Figure 11.29
Results of Example 11.17



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 disables scaling. The system powers-up with scaling disabled.

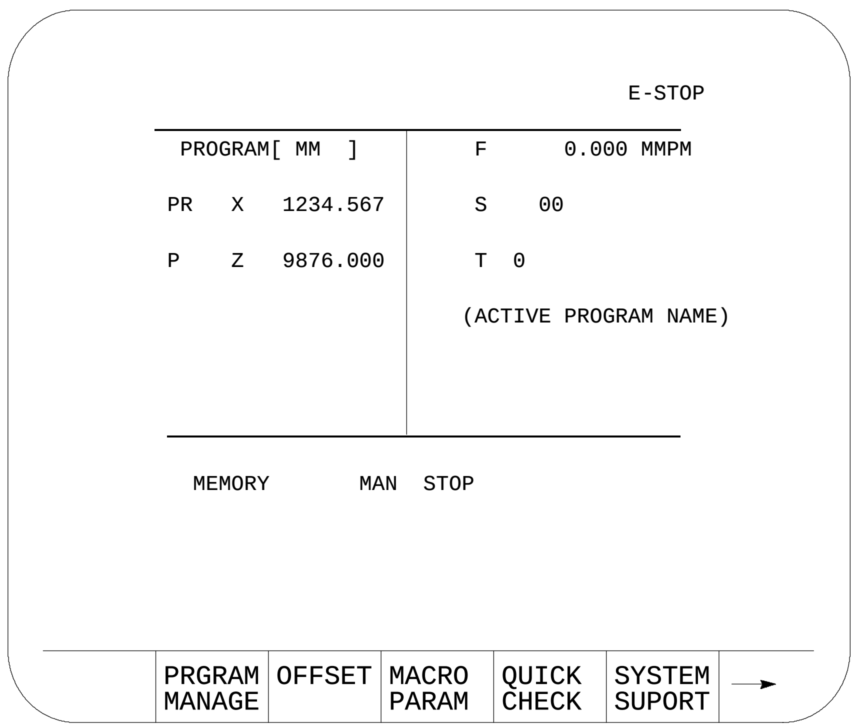
Your system installer specifies in AMP, on an axis by axis basis, whether scaling is allowed. See the literature provided by your system installer for additional information.

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. See the PAL Reference manual (publication 8520-4.2) for additional information.

11.11.1 Scaling and Axis Position Display Screens

When scaling is enabled for a particular axis, the letter "P" appears next to the axis name on all axis position display screens. Figure 11.30 shows scaling enabled on all axes.

Figure 11.30
Axis Position Display Screen Showing Scaling Enabled



11.11.2 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 only be monitored through this screen. The default scaling magnification amount for the axes can be monitored or changed through this screen.

You should only change the default scaling magnification values when the control is in a stopped state. If you change the default values, the new default values do not become active until the control executes the next G14.1 block.

Access the scaling magnification data screen 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	WHEEL GEOMET	RADIUS TABLE	DRESSR TABLE	SCALNG	→

↑	COORD ROTATE	BACKUP OFFSET				→

The scaling magnification data screen appears:

```
SCALING MAGNIFICATION

       CENTER        CURRENT       DEFAULT

X    ±123.00000      2.00000       1.00000

Z    ±123.00000      2.00000       1.00000
```

↑	REPLCE VALUE					

Important: If you configure an axis as a rotary axis, the scaling magnification display screen displays dashes instead of numbers for that axis. You cannot scale rotary axes.

The left column lists the current center of scaling for each axis. When you cancel scaling, 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 you cancel scaling, 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 backup memory when the control is powered down. When the control is powered-up, these values are restored from memory.

3. Use the up and down cursor keys to move the block cursor to the default value you want to change. The selected default value appears in reverse video.

4. To replace stored default scaling magnification value, key-in the new default value and press the {REPLCE VALUE} softkey.

11.11.3 Scaling Restrictions

While scaling is enabled, the following restrictions apply:

- Scaling only affects 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 is scaled if the control is in inverse time mode (G93). Scaling while in inverse time mode is applied as follows:

```
                                                Programmed F word
Scaled F word  (when in G93 mode)  = ----------------------
                                                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 wheel position in the old work coordinate system to the wheel position in the new work coordinate system is not scaled
- Scaling is applied to G52 and G92 offsets. The center of scaling is 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 is adjusted to the new local coordinate systems
- Scaling is not applied to external offsets, wheel radius, wheel geometry, or wheel length offsets.
- Scaling is not 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 is not scaled unless the data is used inside of the macro for motion.
- G22, programmable zone 2 check on and data setting is not scaled
- G53, absolute positions moves is not scaled
- Rotary axes cannot be 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 is applied to the following fixed cycles as shown below. The axis letters can vary depending on how AMP is configured.
  - G31, G31.1 - G31.4

    Gxx X__Z__

    X (scaled)
    Z (scaled)
  - G37, G37.1 - G37.4

    Gxx Z__

    Z (scaled)
  - G80, G81, G81.1, G82, G82.1, G83, G83.1, G84, G84.1, G85, G85.1, G86, G86.1, G87, G87.1, G88, G88.1

    Gxx X__ I__Z__K__R__Y__J__Q__L__F__E__P__D__

    X (scaled)
    I (scaled)
    Z (scaled)
    K (scaled)
    R (scaled)
    Y (scaled)
    J (scaled)
    Q (scaled)
    L (not scaled)
    F (not scaled)
    E (not scaled)
    P (not scaled)
    D (not scaled)

  Important: R uses the scale factor associated with the axis that is perpendicular to the active plane

- G33

  G33 Z_F_E_Q
  G33 X_Z_F_E_Q
  G33 X_F_E_Q

  X (scaled)
  Z (scaled)
  E (not scaled)
  F (not scaled)
  Q (not scaled)

- G34

  G34 Z_F_E_Q_K
  G34 X_Z_F_E_Q_K
  G34 X_F_E_Q_K

  X (scaled)
  Z (scaled)
  E (not scaled)
  F (not scaled)
  Q (not scaled)
  K (scaled)

- G20

  G20 X_Z_I_

  X (scaled)
  Z (scaled)
  I (scaled)

ATTENTION: This cycle cuts more metal when scaling is enabled.

- G24

  G24 X_Z_K_

  X (scaled)
  Z (scaled)
  K (scaled)

ATTENTION: This cycle cuts more metal when scaling is enabled.

END OF CHAPTER

Chapter 12

Axis Motion

12.0 Chapter Overview

This chapter covers the group of G–codes that generate axis motion or dwell data blocks. The major topics covered include:

12.1 Positioning Axes

This section covers the following topics:

Use the 4 basic G–codes to produce axis motion:

G-code:	Function:
G00	Rapid Positioning
G01	Linear interpolation
G02	Circular interpolation (clockwise)
G03	Circular 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 their in-position band. Your system installer sets the in-position band width in AMP. See page 12-70 for details on the G–codes that you can use to modify the in-position band check.

12.1.1 Rapid Positioning Mode (G00)

Format for the rapid positioning mode is as follows:

```
G00X__  Z__ ;
```

Where :	Is :
G00	the G00 code establishes the rapid positioning mode. In this mode, the grinding wheel is fed along a straight line at the rapid feedrate determined in AMP by your system installer.
XZ	the end point of the move generated by the G00 block in the current work coordinate system.

You can perform rapid positioning in the absolute mode (G90), or the incremental mode (G91).

Your system installer determines the feedrate for each axis for the rapid positioning mode in AMP. See page 12-53 for details on feedrates. The acceleration/deceleration type for each axis (linear or exponential) is also defined in AMP. See page 12-63 for details on Acc/Dec.

In rapid positioning mode, as in linear interpolation mode (G01), when more than one axis is commanded, they are driven together to produce a linear move in which all axes start and stop at the same time.

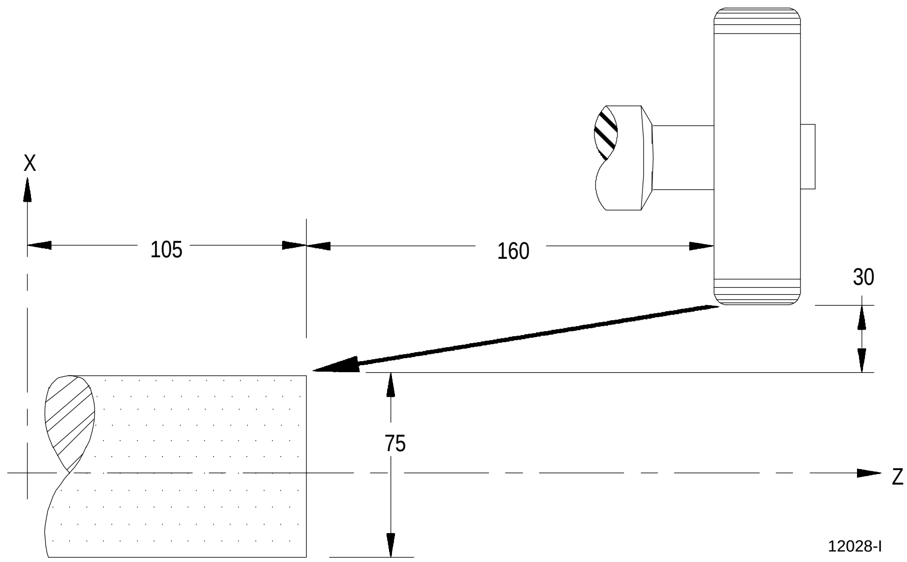
G00 is a modal command and remains in effect until canceled by a G–code of the same group. For a listing of G–code groups, refer to appendix C.

Example 12.1 illustrates a G00 rapid positioning move. This example first establishes diameter programming mode with the G08 block. Diameter programming mode is available on cylindrical grinders only.

Example 12.1
Positioning (G00)

Absolute command	Incremental command
G08;	G08;
G90G00X75.Z105.;	G91G00X-60.Z-160.;

Figure 12.1
G00 Positioning, Results of Example 12.1



Important: The control stores all F-words programmed in the positioning mode as the active feedrate in control memory, but the control ignores them during positioning mode (G00).

12.1.2 Linear Interpolation Mode (G01)

The format for linear interpolation mode is as follows:

```
G01X ____  Z ____ F ____ ;
```

Where :	Is :
G01	G01 establishes the linear interpolation mode. In linear interpolation mode, the wheel is fed along a straight line at the currently programmed feedrate.
XZ	the location of the end point of the linear move in the current work coordinate system.
F	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 the F-word is not programmed a feedrate must have been made active in some previous block.

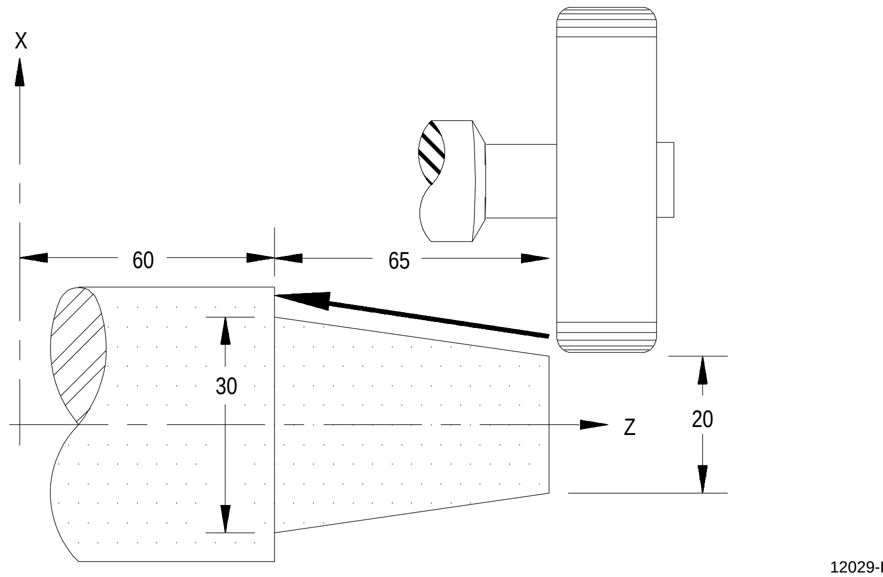
You can perform linear interpolation in the absolute mode (G90), or the incremental mode (G91).

Example 12.2 illustrates a G01 rapid positioning move using linear interpolation. This example first establishes diameter programming mode with the G08 block. Diameter programming mode is available on cylindrical grinders only.

Example 12.2
Linear Interpolation

Absolute command	Incremental command
`G08;`	`G08;`
`G90G01X30.Z60.F.1;`	`G91G01X10.Z-65.F.1;`

Figure 12.2
Results of Linear Interpolation (G01) Example 12.2



Once you program the feedrate F, it remains effective until you program another feedrate (F is modal). You can override programmed F–words. For details on feedrates, see page 12-53.

Example 12.3
Modal Feedrates

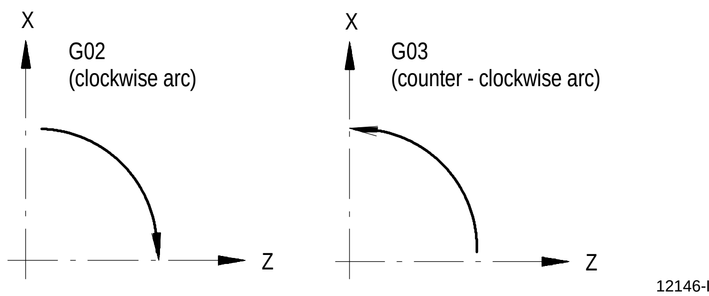
Program Block	Comment
`G91G01X10.Z20.F.1;`	feedrate of .1 is requested
`Z35.;`	feedrate of .1 is effective
`X40.Z35.;`	feedrate of .1 is effective
`Z44.F.3;`	feedrate of .3 is requested

Specify the feedrate for a multi-axis move as the vectorial feedrate. The control adjusts the individual axis feeds to obtain the programmed feedrate. For details on feedrates, see page 12-53.

12.1.3 Circular Interpolation Mode (G02, G03)

G02 and G03 establish the circular interpolation mode. In G02 mode, the grinding wheel moves along a clockwise arc; in G03 mode, the wheel moves along a counterclockwise arc. Figure 12.3 shows clockwise and counterclockwise orientation relative to the positive X and Z axes.

Figure 12.3
Circular Interpolation Direction



Your system installer must first define a plane; typically assign the Z and X axes to the G18 plane. For cylindrical grinders, your system installer typically makes G18 the default plane that the control assumes when power is turned on, E-Stop is reset, or the control is reset.

You can perform circular interpolation in the absolute (G90) or incremental (G91) mode.

Important: S-Curve Acc/Dec mode is not available with circular interpolation mode.

The format for circular interpolation in the ZX plane is as follows:

$$\begin{Bmatrix} G02 \\ G03 \end{Bmatrix} \quad X\_\_\ Z\_\_ \begin{Bmatrix} I\_\_\ K\_\_ \\ R\_\_ \end{Bmatrix} F\_\_\ ;$$

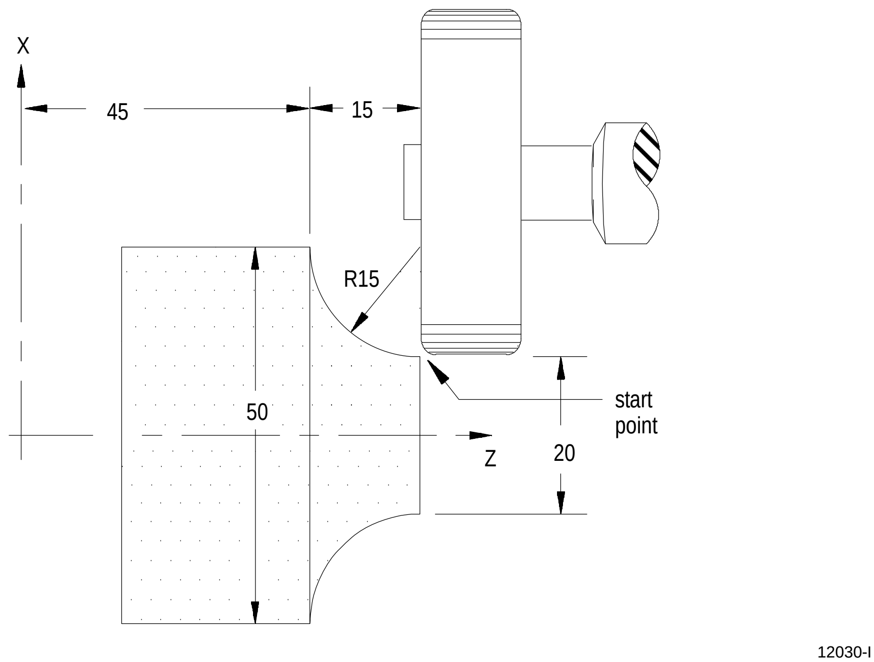
Where :	Is :
X, Z	in absolute (G90) mode, these are the work coordinate values of the end point. In incremental (G91) mode these are the positions of the end point in reference to the start point.
I, K	these determine the position of the arc center. They are the incremental distance on each axis from the start point of the arc to the center point. These values are always incremental, regardless of the established positioning mode (absolute or incremental). I is parallel to X axis, and K is parallel to Z axis; though this can be configured in AMP. These are not necessary if programming the R parameter.
R	rather than defining a center with I, 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 is located such that an arc less than 180 degrees is generated. If R is programmed as a negative value, the center point is located such that an arc greater than 180 degrees is generated. Refer to Figure 12.5 for an example.
F	another option is to enter a feedrate tangential to the arc. If omitted the control uses the feedrate active prior to this block

Example 12.4
Circular Interpolation G18 (ZX Plane)

Absolute Mode	Incremental Mode
G08G02; X50.Z45.I15.K0F.1;	G08G02; X30.Z-15.I15.K0F.1;
or	or
G08G02; X50.Z45.R15.F.1;	G08G02; X30.Z-15.R15.F.1;

In Example 12.4, you can omit the K–word. If you omit I or K from the circular block, the control assumes that they have a value of 0 unless an R–word is present.

Figure 12.4
Results of Circular Interpolation Example 12.4

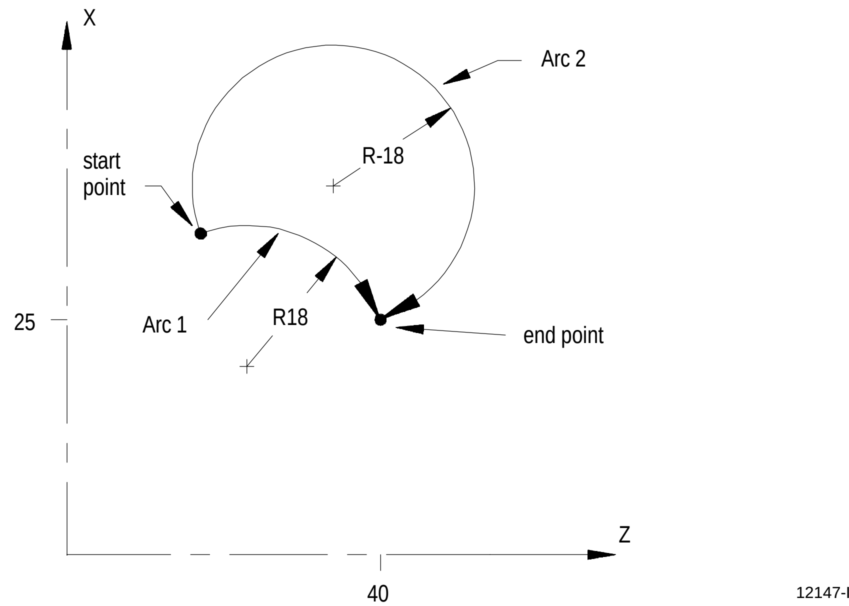


When programming an arc using the radius (R) value, two arcs are possible (Figure 12.5). Program the R–word with a positive or negative value to distinguish between these arcs.

Example 12.5
Arc Programmed Using Radius

Arc 1 center angle less than 180 degrees	Arc 2 center angle greater than 180 degrees
G90G02X25.Z40.R18.F.1;	G90G02X25.Z40.R-18.F.1;

Figure 12.5
Results of An Arc Programmed with Radius Command, Example 12.5



Important: Any axis that is not specified when programming a circle remains at its current axis position value. This results in the end point of an arc having the same coordinate value as the start point of the arc for that axis.

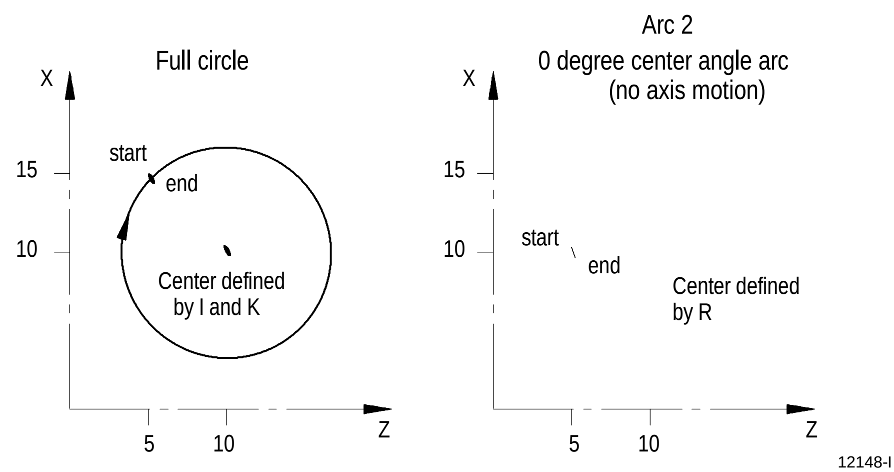
If you do not specify the end point of the arc, or if the end point is the same as the start point, two different results are possible:

- if you use I and/or K to program the arc center, a full circle is made
- if you use R to program the radius of the arc, no arc is made (the control does not move either axis)

Example 12.6
Arc End Points Same as Start Points

Arc 1-Full Circle	Arc 2-No Motion
`G02I-5.K5.F.1;`	`G02R10.F.1;`
or	or
`G02X15.Z5.I-5.K5.F.1;`	`G02X10.Z5.R10.F.1;`

Figure 12.6
Results of An Arc with End Point Equal to Start Point, Example 12.6



If programming a radius command R in the same block as I, and/or K, the control gives the R priority and ignores the I–, and/or K–words.

Important: Your system installer can specify the maximum allowed difference between the starting radius of the arc and the ending radius of an arc programmed with an I, K center. If the difference exceeds the allowed value set in AMP, an error occurs.

12.1.4 Positioning Rotary Axes

The following section describes how to program a rotary axis. 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 spindle orient (M19) cannot move simultaneously with the other axes in the system. A rotary axis is capable of rotating when other axes are being moved.

Your system installer determines which axes are rotary axes or linear axes, in AMP. Also defined in AMP, is the address used to command a rotary axis. This manual assumes that the C–word is used to program a rotary axis. See the documentation prepared by your system installer for the rotary axis words used in a specific system.

Program a rotary axis in units of degrees. This manual assumes that your system installer has configured the rotary axis to “rollover” at 359.99°. Rollover means that after the rotary axis exceeds 359.99° of rotation, its position displays rollover to 0° and starts increasing. If the axis rotates to a position less than 0° its position displays rollover to 359.99° and start decreasing.

Typically you program a rotary axis in a block by itself or with linear moves (rapid G00 or linear G01 moves). You can, however, program a rotary axis in a block that contains circular moves (G02 or G03).

Programming in Absolute or Incremental

You can program rotary axes 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°. The sign given to this angular position determines the direction that the rotary axis travels to reach the programmed angle. For example, if you program:

```
G90C25;
```

in a part program, that causes the rotary axis C to rotate to an angle of 25° (referenced from a position 0 determined by your system installer) and to rotate the axis in the positive direction to reach this position. If you program:

```
G90C-25;
```

in a part program, that causes the rotary axis C to rotate to an angle of 25° and to rotate the axis 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 your system installer. The sign of the angle determines the direction the rotary axis rotates in. For example, if the current C axis position is 25° and you program the following block:

```
G91C50;
```

the C axis would rotate 50° in the positive direction. The new C axis position would be 75°.

If the current C axis position is 25° and you program this block:

```
G91C-50;
```

the C axis would rotate 50° in the negative direction. The new C axis position would be 335°.

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.

If circular interpolation mode is active (G02 or G03) you cannot program a rotary axis move unless the following 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 moved made with the axes in the active plane

12.1.5 PAL Axis Mover

Your 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 over that axis. Jog commands as well as part program commands are typically ignored unless PAL has been written to somehow accept and act on those commands.

You can disable axis position displays on the CRT for an axis under PAL control. See 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.

12.2 QuickPath Plus

This section covers the following topics:

The QuickPath Plus feature (QPP) is offered as a convenient programming method to simplify programming with the control. This method of programming can prove useful in simplifying the programming of a part directly from a part drawing.

The most significant advantage to the QuickPath Plus feature is that the programmer no longer needs to calculate the endpoint of every block or every point of intersection. QuickPath Plus determines these points from the angles and lengths entered.

QuickPath Plus uses these addresses:

A	Angle	This word, which defines the angle of the wheel's path, is determined in AMP by your system installer. This manual assumes that the A-word is used. The angle is always measured counterclockwise from the first axis defining the currently active plane. The angle is in units of degrees.
L	Length	This word defines the length of the wheel's path, the direction of which is determined by the angle (A). It is always interpreted as an incremental value.

The control offers a variety of sample patterns with prompting to aid in the programming of QuickPath Plus. These are found under the QuickView feature described in chapter 5.

Remember these points when programming QuickPath Plus:

- Your system installer may have assigned ,A (rather than A) as the angle word.
- 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 assume the ZX plane is active (angles are measured relative to the Z axis)
- 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 (G07 - G08) has no effect on the A– or L–word
- If you must program more than one block to perform the QuickPath Plus operation being used and an error is made in one of the program blocks, the control always shows the error as being in the first block regardless of which block it is really in. If programming in <SINGLE BLOCK> mode, the control stops after the execution of the first block as normal
- If you must program more than one block to perform the QuickPath Plus operation being used, a maximum of 4 non-motion blocks can be programmed between those blocks. A non-motion block is any block that does not generate axis motion in the current plane
- These G–codes cause a syntax error if programmed in any QuickPath Plus 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, refer to the G-code table in appendix C of this manual.
- If you must program more than one block to perform a QuickPath Plus operation, it causes an error if the current plane is changed to some other parallel plane in between those blocks
- If you program an angle in a circular QuickPath Plus block, an error occurs

- If you program an L–word in a G13, or G13.1 block, an error occurs

12.2.1 Linear QuickPath Plus

This section describes 3 programming situations in which QuickPath Plus can be used:

- Only one end coordinate known
- No end coordinate known (L)
- No intersection known

One End Coordinate Known

Many times part drawings give a programmer only one axis endpoint for a path and require that the other axis endpoint be calculated by using a given angle. The following QuickPath Plus feature eliminates the need for this calculation. This must be a linear block. See page 12-17 for circular QuickPath Plus.

The format for this block is as follows:

$$A\_\_ \quad \left\{ \begin{array}{c} X\_\_ \\ Z\_\_ \end{array} \right\} ;$$

Where :	Is :
`A   Angle`	this word, determined in AMP by your system installer, defines the angle of a linear path. This manual assumes that the A-word is used. The angle is a positive value when measured counterclockwise from the first axis defining the currently active plane and a negative value when measured clockwise. The angle is in units of degrees.
`X,Z End Point`	this word is used to program one of the coordinates of the end point of a linear path. The control calculates the other end point automatically. This can be any axis word that is in the current plane.

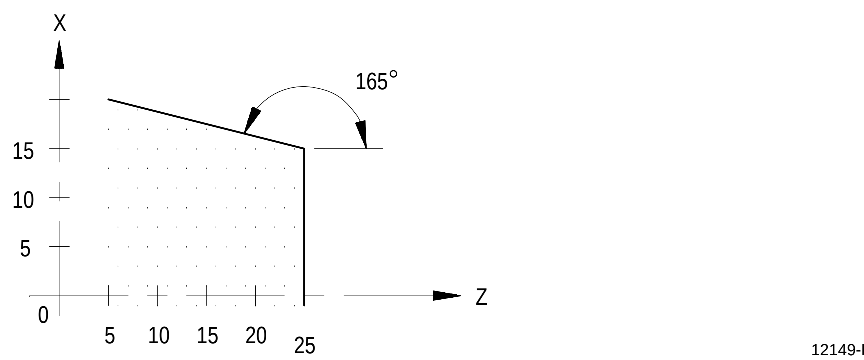
You can only program one axis word from the active plane in this block. Any axis word that is not in the current plane is executed as a normal linear move to that coordinate and combined with the QuickPath Plus generated path. If both axis words from the current plane are entered in the block, the angle is ignored and the control moves to the coordinate position programmed with the axis words. All examples in this section assume the ZX plane is active.

Important: If the value of the angle (A–word) is 0° or 180°, the second axis in the active plane must be programmed in the block as the endpoint. If the value of the angle is 90° or 270°, the first axis in the active plane must be programmed in the block as the endpoint.

Example 12.7
Angle Programmed:

```
N10 G01 X0.0 Z25.0 F.1.;
N20 X15. A90;
N30 Z5.A165;
```

Figure 12.7
Results of Angle Programmed, Example 12.7



Important: Circular QuickPath Plus can also use an angle (A) in a program block. This is described on page 12-17.

No End Coordinate Known (L)

This feature of QuickPath Plus lets the programmer define a path by using only the angle and length of the path. This must be a linear block.

The format for this block is as follows:

```
A__ L__;
```

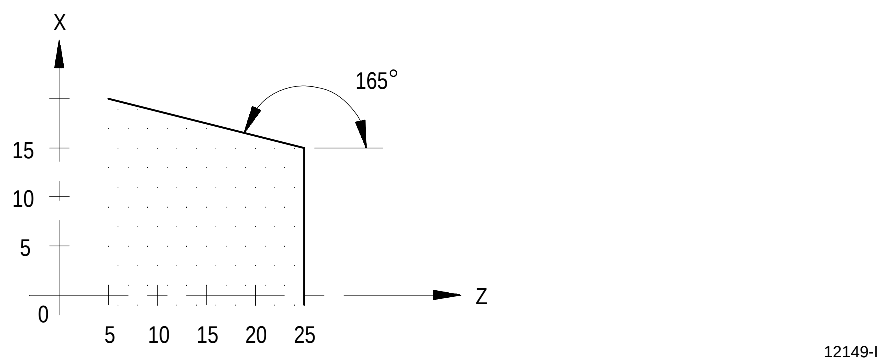
Where:	Is:
A Angle	this word, determined in AMP by your system installer, defines the angle of a path. This manual assumes that the A-word is used. The angle is a positive value when measured counterclockwise from the first axis defining the currently active plane and a negative value when measured clockwise. The angle is in units of degrees.
L Length	this word determines the length of the 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 from the current plane is programmed in the block, the L-word is ignored and the control calculates the end point from the angle and the programmed axis coordinate. If an angle (A) or length (L) is programmed in a block that also includes axis words for both axes in the current plane, then QuickPath Plus is not performed and the control ignores the A– and the L–words in the block.

Example 12.8
Angle and Length Programmed:

```
N10 G01 X0. Z25. F.1.;
N20 A90 L15;
N30 A165 L20.7;
```

Figure 12.8
Results of Angle and Length Programmed, Example 12.8



No Intersection Known

This feature of QuickPath Plus lets the programmer define two intersecting, consecutive, linear paths without knowing the point at which the actual intersection takes place. Both of these blocks must be linear blocks and programmed in absolute mode. The angle of both of these paths must be known.

This is done with a sequence of two linear blocks (in the current plane) in which QPP calculates the end point of the first block. The start point of the first block is the current axis position.

Important: The second block of these two blocks must be programmed in absolute mode. Any attempt to program the second block in incremental generates an error.

The format for these blocks is as follows:

```
N1 A__;

N2 A__Z__X__;
```

Where :	Is :
A Angle	this word, determined in AMP by your system installer, defines the angle of a linear path. This manual assumes that the A-word is used. The angle is a positive value when measured counter-clockwise from the first axis defining the currently active plane and a negative value when measured clockwise. The angle is in units of degrees.
ZX End Point of second block	a representation of the actual coordinates of the end point of the second block. They must be programmed as absolute values and must be axes in the current plane.

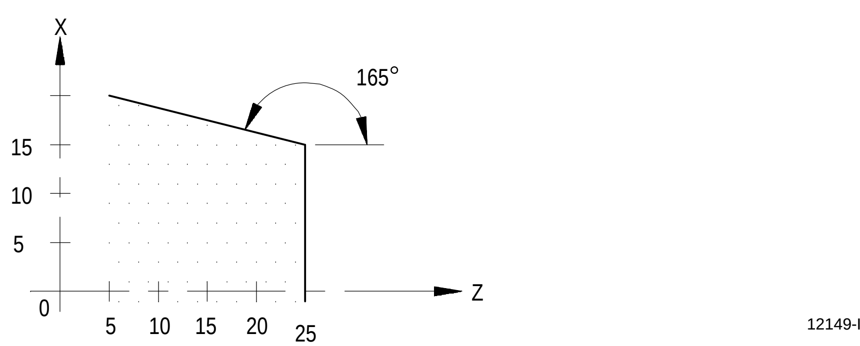
Important: There can be as many as 4 program blocks between the two QPP blocks; however, those blocks cannot generate axis motion in the current plane.

Both of these blocks must be programmed in the same plane. If the current plane is changed between blocks, the control generates an error.

Example 12.9
QuickPath Plus When An Intersection is Unknown

```
N10 G01 X0. Z25. F.1;
N20 A90;
N30 A165 X20.Z5.;
```

Figure 12.9
Results of Unknown Intersection, Example 12.9



If the control cannot determine an intersection point for the two linear paths (for example, if the paths are parallel), an error occurs.

12.2.2 Circular QuickPath Plus (G13, G12.1)

Circular QuickPath Plus helps the programmer when a drawing does not call out the actual intersection of two consecutive paths and at least one of the paths is circular. This provides the programmer with the option of not having to do any complex calculations to determine end points and start points when an arc is involved.

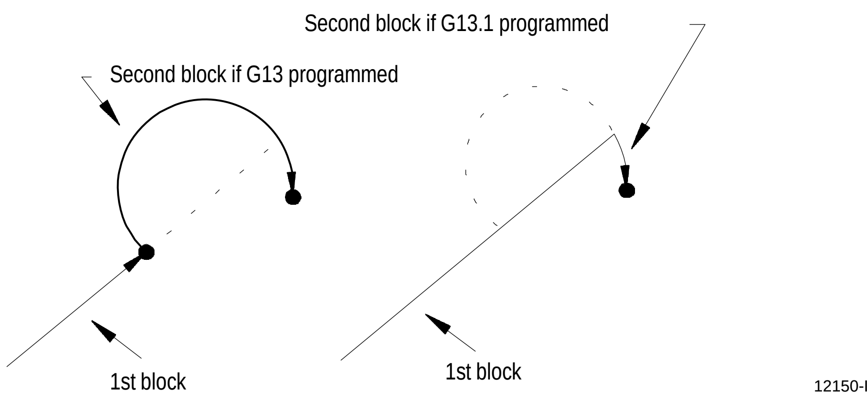
For most cases of circular QuickPath Plus there are two possible intersection points for the two defined blocks. Which intersection is used is determined by programming either G13 or G13.1 in the first of the two blocks:

G13 First Intersection

G13.1 Second Intersection

Programming G13 defines the first intersection that occurs when the path of the first block intersects with the second block. Programming a G13.1 defines the second intersection that occurs when the path of the first block intersects with the second block.

Figure 12.10
G13 versus G13.1 Intersections



When programming Circular QuickPath Plus, remember:

- When there is only one intersection involved with the paths, the G13 and G13.1 can be programmed interchangeably. One of these G–codes still must be programmed
- The G13 or G13.1 must be programmed in the first of the two blocks defining the two paths
- If the arc is programmed with an R–word, the two paths must be tangent. The sign (+ or -) of the R–word determines the arc center location as described on page 12-5

- The angle word (A) cannot be programmed in a circular block
- The absolute coordinate values for both axes in the current plane must be programmed in the second block. Both must be programmed regardless of whether there is axis motion 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. A G13 or G13.1 must be programmed and absolute coordinate values must be used.

Format:

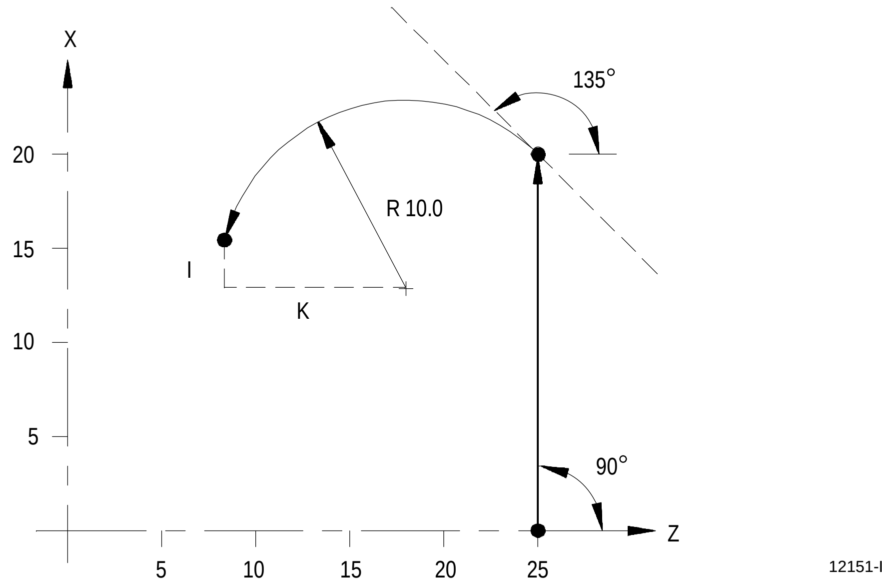
```
G13G01 A__;          or    G13G01 A__;
G02 Z__X__K__I__;          G02 Z__X__R__;
```

Important: If the second block is an arc and it is programmed using I and K integrands, the values programmed with I and K are not measured from the start point of the arc as normally done. 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 12.10
Line Into Arc Without Programming Intersection

```
G00Z25.X0.;
G01G13.1A90;
G03Z7.X15.K9.21I-2.;
```

Figure 12.11
Results of Line into Arc Without Intersection, Example 12.10



Important: R cannot be programmed to specify the arc radius for linear-to-circular block combinations unless the two 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. A G13 or G13.1 must be programmed in the first of the two blocks and absolute coordinate values must be used.

Format:

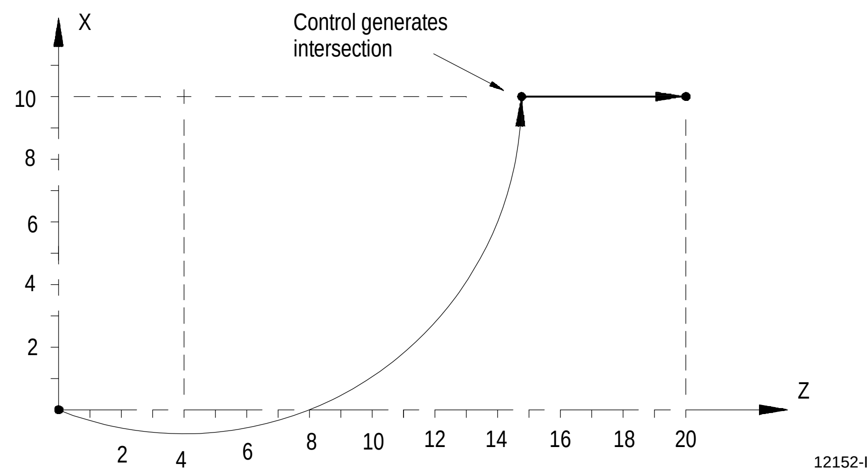
```
G13G02I__K_;   or    G13G02R__;
G01A__Z__X__;        G01A__Z__X__;
```

Important: K values are the normal integrand values when using this format (measured from start point of arc to arc center). These are described on page 12-5.

Example 12.11
Arc into Line Without Programming Intersection Point

```
G0X0Z0.;
G13G03K4I10F.1;
G01A0X10Z20;
```

Figure 12.12
Results of Arc into Line Without Intersection, Example 12.11



Important: R cannot be programmed to specify the arc radius for linear-to-circular block combinations unless the two 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. A G13 or G13.1 must be programmed. If using this format the R-word cannot be used to specify the radius of an arc in either of the circular blocks. These blocks must be programmed in absolute.

Format:

```
G13G02K__I__;
G02 Z__X__K__I__;
```

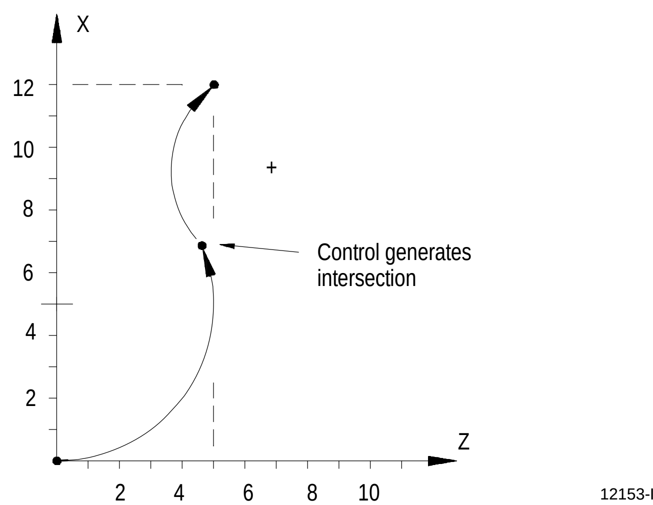
Important: The I and K integrands in the second circular block of this QPP format are not the same values as used for normal circular interpolation. In this case, they specify the distance from the end point of the second arc to the center of that arc. At least one of these integrand words must be programmed in each of the two circular blocks.

Neither circular block can contain an angle word (A) when using this format.

Example 12.12
Arc Into Arc Without Programming Intersection

```
G0X0.Z0.;
G13.G03I5F.1;
G02X12Z5I-2.75K2;
```

Figure 12.13
Results of Arc Into Arc Without Intersection, Example 12.12



12.3 Chamfering and Corner Radius

During cornering, the control can perform a chamfer (a linear transition between blocks) or a corner radius (an arc transition between blocks).

,C	Chamfer size	This word defines a chamfer length that connects two intersecting paths. Its value determines the distance that the chamfer begins and ends from the intersection point.
,R	Corner radius	This word defines the radius of an arc that is tangent to two intersecting paths.

Both the chamfer and the corner radius are generated between two motion blocks which must be programmed in the same plane. The motion block with the corner chamfering (,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.

If more than one ,C– or ,R–words are programmed in the same block, the control uses only the right–most word, and ignores others. The second block can also have a corner chamfering or corner rounding word in it. If it does, the second block is also used as the first block of the next corner chamfering or corner rounding.

The chamfering and corner radius features are often used in conjunction with QuickPath Plus. They can be programmed in either absolute (G90) or incremental (G91) modes.

ATTENTION: If a programming error of some type is made in the block defining the second path, the chamfer or radius is not made. Instead the first block is executed to its programmed endpoint without a chamfer or radius. This can cause damage to the part or grinding wheel.

Three common error conditions that occur with corner programming are:

- if a programmed block with a ,C or ,R is not followed by another motion block in the same plane prior to the end of program, an error results
- if an MDI block with a ,C or ,R is entered it must be immediately followed by an MDI block with motion in the same plane, or else an error results
- if more than 4 non-motion blocks are programmed immediately after a ,C or ,R block, an error results. A non-motion block is any block that does not generate axis motion in the current plane

Chamfering

Program a ,C–word to grind a chamfer between two consecutive intersecting paths. The chamfer word C must follow a comma (,) and is programmed in the first of two paths to be connected by the chamfer.

The value programmed with the ,C determines the chamfer size:

- if the path is linear, then the distance programmed with the ,C–word is measured from the intersection of the paths to the point where the chamfer begins or ends
- if the path is circular, then the chamfer distance programmed is applied as a chord length on the arc measured from the intersection between the two paths

This applies regardless of the combination of arcs and lines to be chamfered.

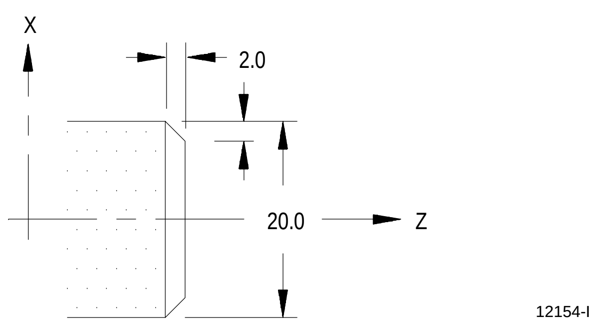
The ,C–word can be programmed anywhere in a block as long as no space is programmed before or after the ,C–word. The value programmed with the ,C is an incremental distance <u>not</u> affected by radius or diameter programming (G07 or G08).

Example 12.13 illustrates a chamfer between two linear paths. This example first establishes diameter programming mode with the G08 block. Diameter programming mode is available on cylindrical grinders only.

Example 12.13
Basic Chamfering Using ,C

```
N10G90G08;
N20Z25.0X0.0F.1.;
N30G01X20.,C2.0;
N40Z5.0;
```

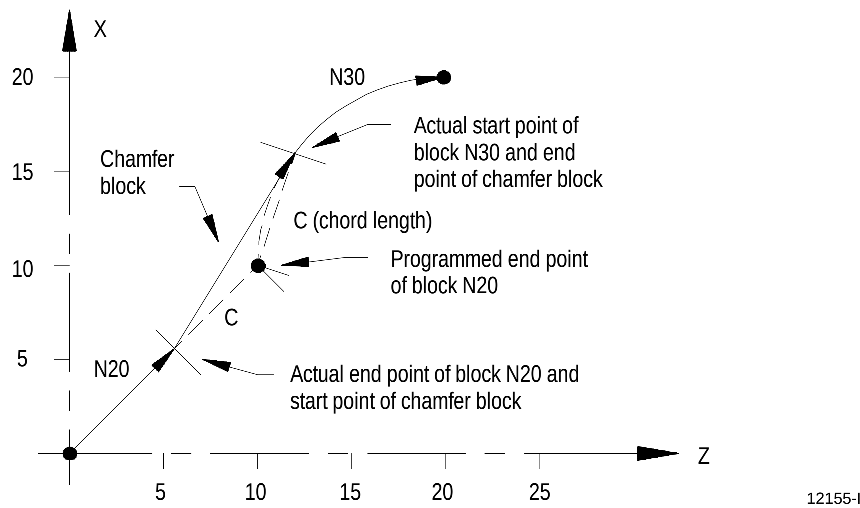
Figure 12.14
Results of Chamfering Using ,C from Example 12.13



Example 12.14
Linear-to-Circular Motions with Chamfer

```
N10X0.Z0.F.1;
N20X10.Z10.,C5;
N30G02X20.Z20.R10;
```

Figure 12.15
Results of Linear-to-Circular Motions with Chamfer, Example 12.14



Corner Radius

Program a ,R–word to grind a radius between two consecutive intersecting paths. The radius word R must follow a comma (,) and is programmed in the first of two paths to be connected by the radius.

The value programmed with the ,R determines the radius size. Regardless of whether the programmed paths are linear or circular, a circular move of the programmed radius is executed tangent to both paths. If the control cannot make an arc that is tangent to both paths, an error is generated.

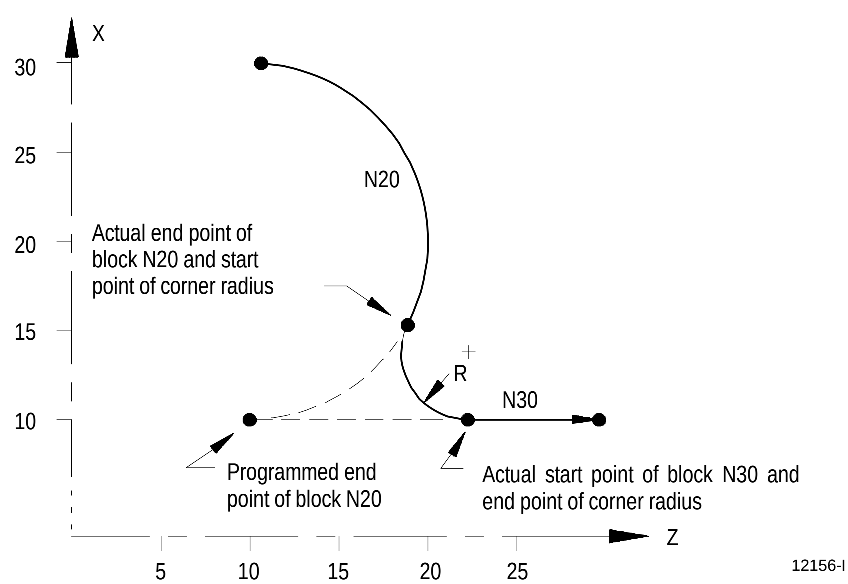
Important: If the two motion blocks are tangent to each other, then any corner rounding commands are ignored.

The ,R–word can be programmed anywhere in a block as long as no space is programmed before or after the ,R–word. The value programmed with the ,R is an incremental distance not affected by radius or diameter programming (G07 or G08).

Example 12.15
Programming a Radius for a Circular Path into a Linear path.

```
N10Z10X30.F.1;
N20G02X10.Z10.R10,R3;
N30Z30.X10.;
```

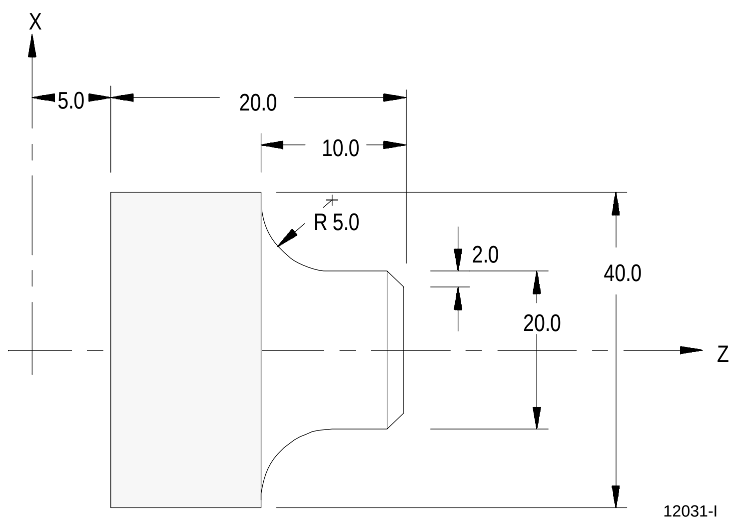
Figure 12.16
Results of Radius for a Circular Path into a Linear path, Example 12.15



Example 12.16
Radius and Chamfer with QuickPath Plus

```
N10Z25.X0.F.1;
N20G01A90,C2.;
N30Z15.X20.A180,R5.;
N40X40.;
N50Z5.;
```

Figure 12.17
Results of Radius and Chamfer, Example 12.16



Considerations with Chamfering and Corner Radius

- If the control is executing in single block mode, the control enters 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 executes 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 this word is used to cut the chamfer or radius. For example ,C-10 is executed 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 generates an error
- ,C and ,R cannot be programmed in a block that contains any of the following:
  - any fixed cycle G–codes
  - any Dwell commands
  - thread grinding blocks
  - programmable zone G–codes
- Your system installer determines in AMP the resolution of the ,C– and ,R–words for both inch and metric programming. Refer to the documentation prepared by your system installer for details

12.4 Automatic Motion To and From Machine Home

This section covers the following topics:

Machine tools have a fixed machine home position 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 on the MTB panel provided solely for this purpose. Manual homing is described in detail in chapter 4
- Automatic machine home operation that uses a programmed machine home code

12.4.1 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 automatically homes any axes programmed in the G28 block that have not yet been homed. Only axes that have their axis words programmed in the G28 block are homed.

Homing follows the sequence of homing events described in chapter 3.

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 considers 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".

12.4.2 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__ 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, Z) in memory to be used as the point of return for the automatic return from machine home operation called out by G29.

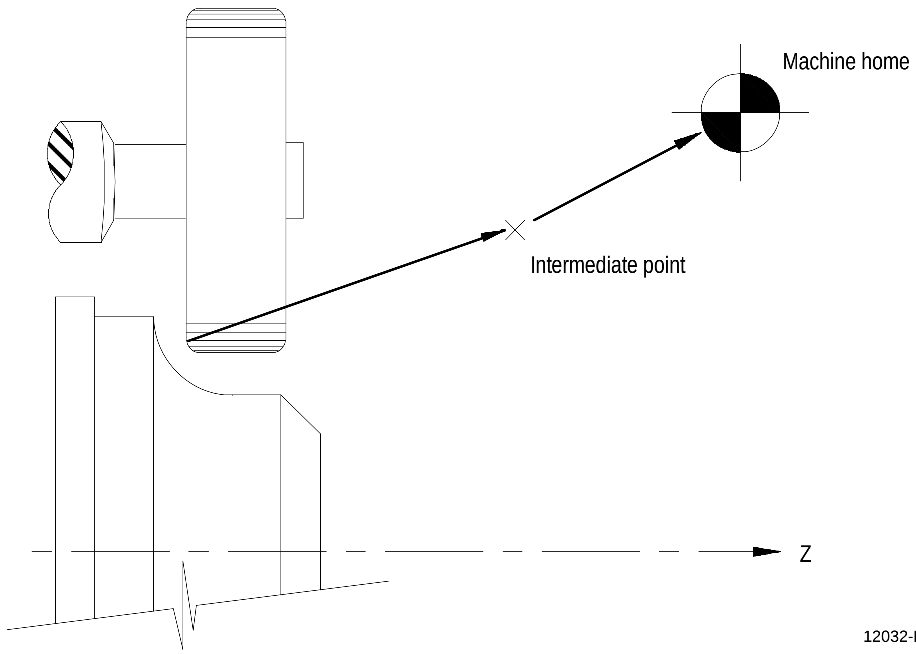
The return operation generates two axis moves both executed at the rapid feedrate. The first move is to the intermediate point and the second is to the axis home position.

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 are only temporarily canceled for the return to home moves.

Only the axes specified in the G28 block are returned to home. For example:

`N1 G28 X4.0;`	(X-axis is moved to home after moving to 4.0)
`N2 G28 X4.0 Z2.0;`	(X and Z axes are moved to home after moving to (4.0, 2.0))

Figure 12.18
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 stores the intermediate point in memory for 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 remains at the old value. For example:

`N1 G28 X4.0 Z3.0;`	Intermediate point X=4 Z=3
`N2 G28 Z2.0;`	New intermediate point X=4, Z=2

Important: When the control executes a G28 or G30 block, it temporarily removes any offsets and compensation during the axis move to the intermediate point. The offsets and/or 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 compensation remain deactivated on the way to the intermediate point and are re-activated when the axis moves from the intermediate point back to the point indicated in the G29 block.

12.4.3 Automatic Return from Machine Home (G29)

When a G29 is executed in a part program (or through MDI), the axis or axes move first to the intermediate point, and then to the position indicated in the G29 block. If a G28 was just executed, this has the effect of returning the axis from machine home. For example, executing the block:

```
G29 X7.0 Z1.5;
```

in absolute mode would move the axes to (7.0, 1.5) after passing through the intermediate point stored in control memory. In incremental mode, this block would move the axes to a position that is X7.0 and Z1.5 units away from the home point.

The intermediate point is stored in control memory after a G28 return to machine home or a G30 move to alternate home is executed. A G29 block is usually executed after a G28 or G30 block, typically to return the grinding wheel to the part.

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 are temporarily canceled for the return from home moves.

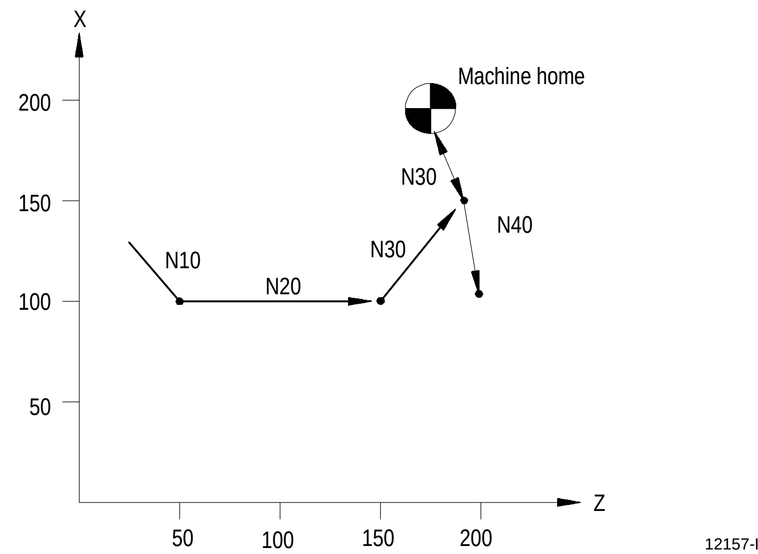
Only the axes specified in the G29 block are moved. For example:

N1 G28 X5.0 Z1.0;	(X and Z axes are moved to home after moving to X=5.0 Z=1.0)
N2 G29 X3.;	(X moves to X=5.0 then to X=3.0 --- Z does not move)

Example 12.17
Automatic Return From Machine Home

```
N00010  X100.Z50.;
N00020  Z150.;
N00030  G28X180.Z180.;
N00040  G29X200.Z100.;
```

Figure 12.19
Automatic Return From Machine Home, Results of Example 12.17



Important: When a G29 is executed, offsets and/or compensation is deactivated on the way to the intermediate point and are re-activated when the axis moves from the intermediate point back to the point indicated in the G29 block.

12.4.4 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 are moved.

```
G27 X__ 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 takes place and the control issues 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 are reactivated immediately after the G27 block is completed.

G27 block commands are usually given after wheel offset modes have been cancelled.

If an attempt is made to execute a G27 before the axes have been homed, the control enters cycle stop and the following error message occurs:

“MACHINE HOME REQUIRED OR G28”

12.4.5 Move To Alternate Home (G30)

The G30 command is similar to the G28 command, 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 are defined for each axis in AMP by your system installer.

To use the G30 command, follow this format:

```
G30 X__ Z__;
    or           (second alternate home position)
G30 P2 X__ Z__;

G30 P3 X__ Z__;  (third alternate home position)
G30 P4 X__ Z__;  (fourth alternate home position)
```

The axis words in the above block establish the intermediate point in the same manner as the G28 code described on page 12-29. The axes move to their intermediate points defined in the G30 block prior to moving to their alternate home positions. This intermediate point is the same intermediate point as the one described with the G28 code. When intermediate values are programmed in a G28 block, they replace G30 intermediate point values and visa-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.

If an axis included in the G30 block has not been homed, block execution stops and the following error message appears:

“MACHINE HOME REQUIRED OR G28”

Important: When the control executes a G28 or G30 block, it temporarily removes any offsets and compensation during the axis move to the intermediate point. The offsets and/or 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 compensation remain deactivated on the way to the intermediate point and are re-activated when the axis moves from the intermediate point back to the point indicated in the G29 block.

12.5 Spindle Speed Control

This section covers the following topics:

Grinding speeds are established by setting spindle speeds. In the case of cylindrical grinders, there are two separate spindles, the grinding wheel spindle and the part spindle. The control uses the S-word to control the speed of one of the spindles. If a machine requires more than one spindle, the second must be controlled manually or through the PAL program. Refer to the documentation prepared by your system installer for details.

The S-word controlled spindle operates in one of the following modes:

- Constant Surface Speed Mode (G96) - maintains the speed of the grinding wheel surface relative to the part surface at a constant rate. This relative speed is measured as a surface speed with units of m/min, m/sec, in/min, or in/sec.

  To maintain a constant surface speed when grinding a cylindrical part at different diameters, the grinding wheel or part spindle speed should be decreased as diameters are increased. This maintains a constant speed of the part surface relative to the grinding wheel surface. The same is true when dressing a wheel. In that case constant surface speed applies to the wheel speed relative to the dressing tool.

- RPM Spindle Speed Mode (G97) - maintains a constant spindle speed equal to the programmed S-word. This speed is independent of the working diameter of the part or the grinding wheel. G97 is described in detail on page 12-51 of this chapter.

There are a number of different grinder configurations possible using the control. The constant surface speed mode (G96) and the RPM spindle speed mode (G97) can be applied to certain grinder configurations under certain conditions. To adequately deal with the possibilities, this section is divided as follows:

If you have a:	that:	see page:
Surface Grinder	does not use the S-word for spindle speed control	12-35
Surface Grinder	uses the S-word to control the wheel spindle speed	12-36
Cylindrical Grinder	uses the S-word to control the part spindle speed	12-37
Cylindrical Grinder	uses the S-word to control the wheel spindle speed	12-40

In addition to the previously listed topics, there are other sections of this manual with information relative to spindle speed control:

Topic:	See page:
Notes on Constant Surface Speed mode	12-42
CSS axis selection	12-45
CSS examples	12-46
RPM spindle speed mode (G97)	12-51

12.5.1 Surface Grinder, No S-word

If your surface grinding machine does not use the S–word, then CSS (programmed with a G96) is not available. RPM spindle speed mode (programmed with a G97) also does not have any effect since it too acts on the S–word.

Machines that fall into this category have manual or PAL controlled spindles. Typically speeds are changed through a manual gear change, a speed selection switch or special M codes. Much of this depends on your control's AMP configuration and PAL program. Refer to the documentation prepared by your system installer for details.

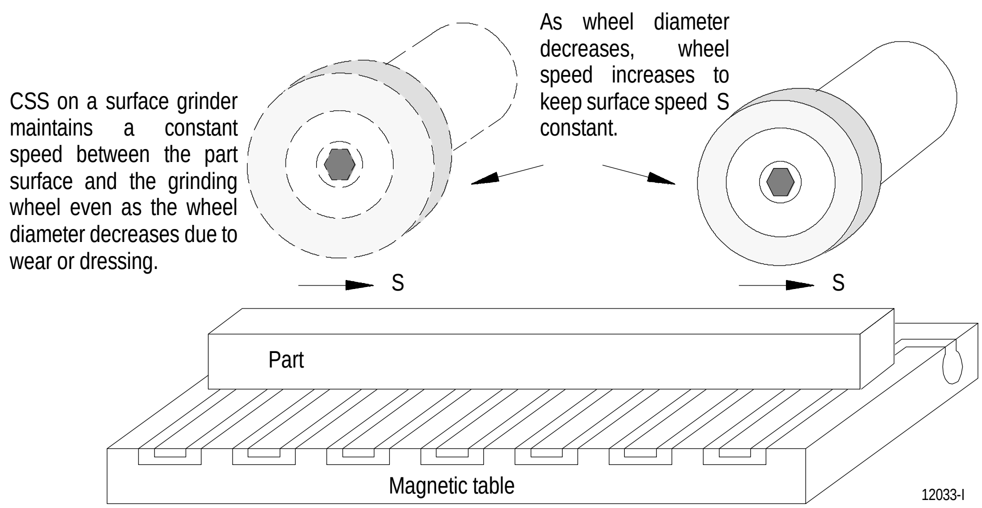
12.5.2 Surface Grinder, S-word for Wheel Speed

If your surface grinding machine uses the S–word to control wheel spindle speed, then CSS (programmed with a G96) and its counterpart RPM spindle speed mode (programmed with a G97) are available.

When CSS is active during surface grinding, it maintains the surface speed at the diameter of the grinding wheel at a constant rate. For example, dressing the grinding wheel typically reduces its diameter. With CSS active, the control increases the wheel speed proportional to the decrease in diameter to maintain the surface speed of the wheel relative to the part.

When using CSS during surface grinding and wheel dressing, the wheel spindle typically is selected as the CSS axis. CSS axis selection is described on page 12-45.

Figure 12.20
CSS on a Surface Grinder



This is especially important on machines that perform in-process wheel dressing or lose substantial amounts of wheel material with each pass. In those cases, the change in diameter during a long grinding operation can be significant. If the wheel speed is not adjusted (through CSS), then there could be a noticeable variation in surface finish.

The format for the G96 block is:

```
G96 L__S__;
```

Where :	Is :
L	specifies whether CSS is in per minute or per second mode. L1 specifies per second mode. L0 specifies per minute mode. If L is not programmed the control uses the default mode defined in AMP by your system installer.
S	specifies the constant surface speed in meters or feet per second (L1) or per minute (L0).

Use the following equation to determine grinding wheel speed in RPM when CSS is active. This equation is for reference only. The value obtained here is calculated automatically by the control and you do not need to program it.

$$N_w = \left(\frac{(S_{program})(K)}{D_w}\right)$$

Where :	Is :
N_w	Speed of grinding wheel (rpm) [1]
K	Constant 318.31 (1000/π) for metric system or 3.8197 (12/π) for inch system
$S_{program}$	Surface speed programmed in the G96 block (m/min or feet/min) [1]
D_w	Dressed or worn diameter of grinding wheel (mm or inch)

[1] if using units of seconds, make sure all S and N values are entered in seconds

12.5.3 Cylindrical Grinder, S-word for Part Speed

If your cylindrical grinding machine uses the S–word to control part spindle speed, then CSS (programmed with a G96) is available when grinding the part. RPM spindle speed mode (programmed with a G97) is also available for the part spindle.

Machines that fall into this category typically have manual or PAL controlled wheel spindles. Wheel spindle speeds are changed through a manual gear change, a speed selection switch or special M codes. Much of this depends on your control's AMP configuration and PAL program. Refer to the documentation prepared by your system installer for details.

Important: Constant Surface Speed (CSS) is applied only to the spindle that is controlled with the programmed S–word.

The format for the G96 block is:

```
G96 L__S__;
```

Where :	Is :
L	specifies whether CSS is in per minute or per second mode. L1 specifies per second mode. L0 specifies per minute mode. If L is not programmed the control uses the default mode defined in AMP by your system installer.
S	specifies the constant surface speed in meters or feet per second (L1) or per minute (L0).

When using CSS during part grinding, the axis perpendicular to the part axis typically is selected as the CSS axis. CSS axis selection is described on page 12-45.

Grinding wheel surface speeds relative to the surface of a rotating part should be based on the specifications of the wheel and the material being ground. Enter this surface speed in the G96 block using the S parameter as described below:

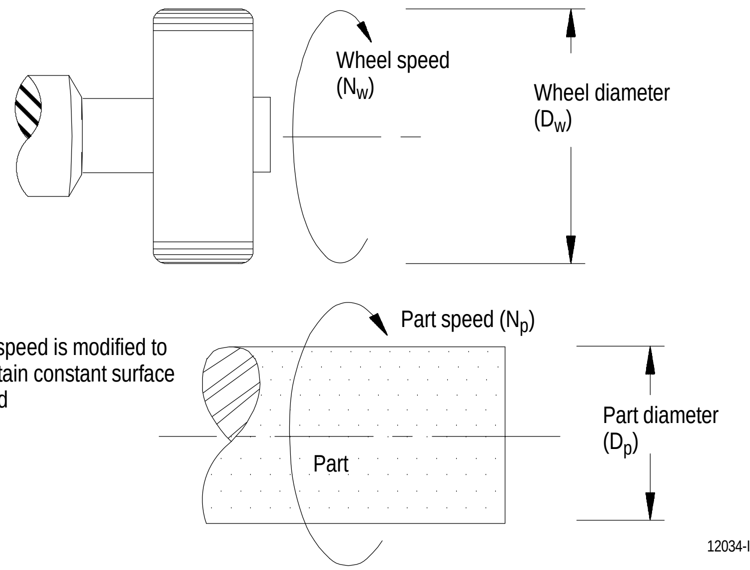
- If you want to program the part surface speed relative to a stationary wheel, enter that value directly with the S parameter in the G96 block.
- If you want to program actual surface speed (part surface speed relative to a rotating wheel), you must use the following equation to determine the correct value to enter with the S parameter in the G96 block. This is necessary to take into account the wheel rotation speed. This equation assumes that the wheel and part are rotating into each other (both spinning in the same direction).

$$S_{program} = S_{desired} - \left(\frac{(D_w)(N_w)}{K}\right)$$

Where :	Is :
$S_{program}$	Surface speed to program in the G96 block (m/min or feet/min) [1]
$S_{desired}$	Desired actual surface speed (m/min or feet/min) [1]
D_w	Diameter of grinding wheel (mm or inch)
N_w	Speed of grinding wheel (rpm) [1]
K	Constant 318.31 (1000/π) for metric system or 3.8197 (12/π) for inch system

[1] if using units of seconds, make sure all S and N values are entered in seconds

Figure 12.21
Part Spindle Speed Modified for CSS



Use the following equation to determine part speed in RPM when CSS is active. This equation is for reference only. The value obtained here is calculated automatically by the control and there is no need to program it.

$$N_p = \left(\frac{(S_{program})(K)}{D_p}\right)$$

Where :	Is :
N_p	Speed of part wheel (rpm) [1]
$S_{program}$	Surface speed programmed in the G96 block (m/min or feet/min) [1]
K	Constant 318.31 (1000/π) for metric system or 3.8197 (12/π) for inch system
D_p	Diameter at which part is being ground (mm or inch)

[1] if using units of seconds, make sure all S and N values are entered in seconds

Since the wheel spindle is not controlled by the S–word, grinding wheel surface speeds relative to a stationary wheel dresser cannot be held constant using CSS. Instead the wheel spindle speed should be a compromise speed (in RPM) that best accommodates the variations in surface speed as the wheel is dressed at different diameters.

12.5.4 Cylindrical Grinder, S-word for Wheel Speed

If your cylindrical grinding machine uses the S-word to control wheel spindle speed, then CSS (programmed with a G96) is available when dressing the grinding wheel and when grinding the part. RPM spindle speed mode (programmed with a G97) is also available for the wheel spindle.

Machines that fall into this category typically have manual or PAL controlled part spindles. Part spindle speeds are changed through a manual gear change, a speed selection switch or special M codes. Much of this is dependent on your control's AMP configuration and PAL program. Refer to the documentation prepared by your system installer for details.

Important: Constant Surface Speed (CSS) is applied only to the spindle that is controlled with the programmed S-word.

The format for the G96 command is:

```
G96 L__S__;
```

Where :	Is :
L	specifies whether CSS is in per minute or per second mode. L1 specifies per second mode. L0 specifies per minute mode. If L is not programmed the control uses the default mode defined in AMP by your system installer.
S	specifies the constant surface speed in meters or feet per second (L1) or per minute (L0).

When using CSS during wheel dressing, the wheel spindle typically is selected as the CSS axis. CSS axis selection is described on page 12-45.

Grinding wheel surface speeds relative to a stationary wheel dresser should be based on the specifications of the wheel and the wheel dresser. Enter this surface speed in the G96 block using the S parameter as described below.

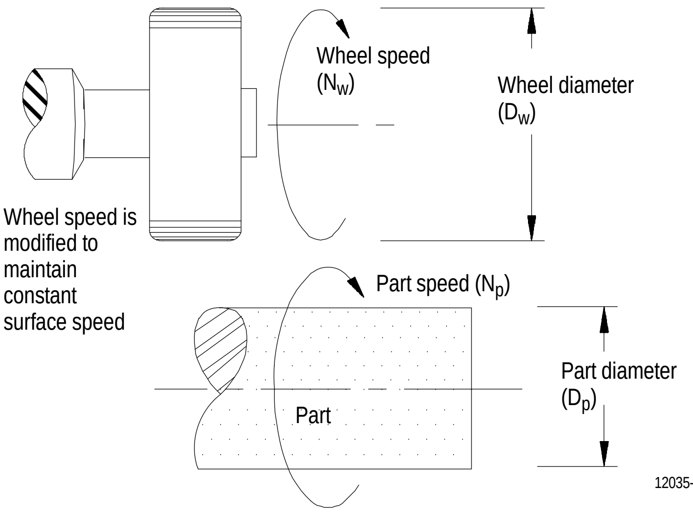
Use the following equation to determine grinding wheel speed in RPM when CSS is active. This equation is for reference only. The value obtained here is calculated automatically by the control and there is no need to program it.

$$N_w = \left(\frac{(S_{program})(K)}{D_w}\right)$$

Where :	Is :
N_w	Speed of grinding wheel (rpm) [1]
$S_{program}$	Surface speed programmed in the G96 block (m/min or feet/min) [1]
K	Constant 318.31 (1000/π) for metric system or 3.8197 (12/π) for inch system
D_w	Diameter at which wheel is being dressed (mm or inch)

[1] if using units of seconds, make sure all S and N values are entered in seconds

Figure 12.22
Wheel Spindle Speed Modified for CSS



Grinding wheel surface speeds relative to the surface of a rotating part should be based on the specifications of the wheel and the material being ground. Enter this surface speed in the G96 block using the S parameter.

When using CSS during part grinding, the axis perpendicular to the part axis typically is selected as the CSS axis. CSS axis selection is described on page 12-45.

- If you want to program the wheel surface speed relative to a stationary part, enter that value directly with the S parameter in the G96 block.
- If you want to program actual surface speed (wheel surface speed relative to a rotating part), you must use the following equation to determine the correct value to enter with the S parameter in the G96 block. This is necessary to take into account the part rotation speed. This equation assumes that the wheel and part are rotating into each other (both spinning in the same direction).

$$S_{program} = S_{desired} - \left(\frac{(D_p)(N_p)}{K}\right)$$

Where :	Is :
$S_{program}$	Surface speed to program in the G96 block (m/min or feet/min) [1]
$S_{desired}$	Desired actual surface speed (m/min or feet/min) [1]
D_p	Diameter at which part is being ground (mm or inch)
N_p	Speed of part wheel (rpm) [1]
K	Constant 318.31 (1000/π) for metric system or 3.8197 (12/π) for inch system

[1] if using units of seconds, make sure all S and N values are entered in seconds

12.5.5 Notes on Constant Surface Speed Mode (G96)

The previous sections described the basic function of CSS with different grinding machine configurations. This section includes some specific considerations for CSS operation.

Enabling (M58) and Disabling (M59) CSS Mode

The G96 mode must first be enabled by programming an M58 code. G96 is modal and remains active until it is cancelled by the G97 code or disabled with an M59 code.

Use M58 or M59 to turn constant surface speed mode on or off. When M59 is programmed, the control ignores G96 mode and the spindle revolves at whatever speed it was at when the M59 was executed. If an M58 is executed to restore the G96 mode, the spindle immediately changes speed (if necessary) to maintain the correct surface speed.

If an S–word is programmed in the same block as an M59, G96 mode is ignored but the S value is registered in memory as the constant surface surface speed to be used if CSS is reactivated. The spindle speed does not change relative to diameter.

When programming M58, the M59 code is cancelled and the G96 mode becomes active again. The spindle returns to the previous G96 surface speed unless an S–word was programmed in a previous M59 block.

ATTENTION: Restoring the constant surface speed mode might cause the spindle speed to change rapidly depending on the grinding wheel position relative to the active CSS axis.

Programming G96 with no S-word

When changing from G97 to G96 mode, if an S–word is programmed in the G96 block the spindle changes speed to produce the surface speed value indicated by the S–word. If no S–word is programmed in the G96 block, the control uses the current surface speed as the CSS value to be maintained.

In Example 12.18, modes are switched from G97 to G96, but there is no S–word programmed in the G96 block. When the G96 is executed, initially the spindle does not change from the 5,000 rpm programmed in the G97 block. When a change in diameter is made, the control changes spindle speed to maintain the surface speed that was effective when the spindle was running at 5,000 rpm at the previous diameter.

Example 12.18
Initiating G96 mode with no S-word

Program	Comments
`G97 S5000;`	RPM spindle speed mode, 5,000 rpm
`X25. Z5.;`	X diameter move to 25, Z move to 5
`X20.;`	X move to 20, spindle remains at 5,000 rpm
`G96;`	CSS mode, no S-word, spindle remains at 5,000 rpm
`X15.;`	X move to 15, spindle speed increases to maintain the surface speed effective when the G96 was executed

Important: In most cases, spindle motion is started and stopped by executing an M code (typically M03 or M04 to start and M05 to stop). Refer to the documentation prepared by your system installer.

Setting a Maximum CSS Spindle Speed with G92

In G96 mode, spindle speeds increase as diameters decrease. If a maximum spindle speed is reached, it is held (not exceeded) until the G96 required speed drops below the spindle maximum.

Important: The maximum spindle RPM can be affected by gear selection (usually controlled by PAL) and maximum gear speeds (defined in AMP). Refer to the documentation prepared by your system installer.

You have the option of lowering the maximum spindle speed. A temporary maximum spindle speed can be programmed by a G92 block with an S value in RPM. This upper limit is valid only for the G96 mode and is ignored in G97 mode. The value for this upper limit cannot exceed the value set in AMP for the maximum spindle speed.

To enter 3,500 rpm as the upper limit, program:

```
G92S3500;
```

Important: The G92 S command to set a new maximum spindle RPM in CSS cannot be programmed while CSS is active.

The G92 maximum spindle RPM is canceled only after power is shut down, a control reset is performed, or when the control goes into E-Stop. M02 or M30 do not cancel this value. When canceled, the maximum speed defined by your system installer becomes effective again.

Important: A maximum spindle RPM for CSS can also be entered into the in-process dresser status screen. This maximum RPM is in affect only while the in-process dresser is active. While the in-process dresser is active the control uses the slowest maximum CSS RPM defined between AMP, the G92 command and the in-process dresser parameter.

Spindle Speed during Rapid Traverse

During rapid moves while in G96 mode, spindle speed changes in one of the following ways:

- the spindle speed changes constantly as the relative position of the CSS axis is changed

  or

- the control calculates the total change in spindle speed (if any) that would be made from beginning to end of the rapid move and uses that spindle speed for the entire rapid move.

Your system installer selects the spindle speed control type in AMP.

Displayed Spindle Speed during CSS

Regardless of whether you are in G97 or G96 mode, the CRT display shows the current spindle speed in RPM following the S–word. This is also true during CSS. The display shows the actual spindle RPM, not the surface speed.

CSS R-word Programming

The optional R-word lets you specify a surface footage value to be applied at a point other than the current axis position. The R-word defines the incremental distance from the current wheel edge to the spindle centerline. The sign of the R-word defines on which side of the spindle centerline the tool tip is positioned.

The R-word is recognized only in a G96 block; its value is valid as long as CSS mode is active and may only be changed when a G96 is programmed in the block.

If you change the S-word (surface footage) while in G96 mode and if an R-value was previously programmed in a G96 block, the R-value is not cancelled.

Important: An R0 and no R-word do not mean the same thing. An R-word of zero means that the spindle centerline is the tool tip position. No R-word means to use the current position.

12.5.6 CSS Axis Selection

CSS axis selection depends on what type of machine you have, the axis configuration for that machine and the specific grinding application.

- Surface grinding – the spindle is typically selected as the CSS axis. This results in CSS calculations being based on the distance from the spindle centerline to the diameter at which the grinding wheel is being used or dressed
- Cylindrical grinding, S–word controls part spindle – the CSS axis is typically the axis perpendicular to the part centerline. This results in CSS calculations being based on the distance from the part centerline to the diameter being ground on the part
- Cylindrical grinding, S–word controls wheel spindle – the CSS axis is typically the axis perpendicular to the part centerline both when grinding the part and when dressing the wheel. This allows implementation of CSS during part grinding and wheel dressing

CSS axis assignment is determined by your system installer in AMP. The CSS axis can be changed by programming a P–word (P0 through P9) in the G96 block when activating CSS.

Each P–word corresponds to a specific axis assigned to it in AMP. Any CSS axis changes made by programming a P–word in the G96 block remain in effect regardless of what mode the control is in. The default CSS axis is assigned to P0 and is active on power-up and after a control reset.

Your system installer can assign the spindle to be the CSS axis by entering S as the CSS axis in AMP. If S is assigned to a P–word, it can also be activated through the G96 block. When active (typically for surface grinders), it determines surface speed based on the current wheel diameter value described on page 21-9.

12.5.7 CSS Examples

This section provides examples using CSS in typical surface and cylindrical grinding applications. Before programming any of these examples, verify that the conditions listed apply to your machine.

CSS While Grinding Part (surface grinding application)

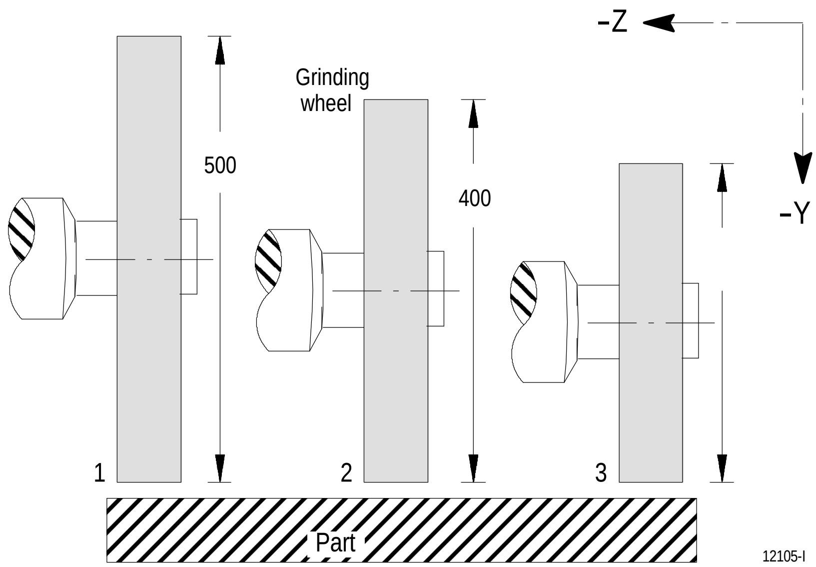
Example 12.19 shows how the wheel spindle speed changes with CSS active to compensate for a changed wheel diameter. The following conditions are assumed for this example:

- the S-word for this machine applies to the grinding wheel spindle
- the grinding wheel spindle has been selected as the CSS axis
- constant K is 318.31 (metric constant)
- the maximum wheel spindle speed is 9,000 rpm

Example 12.19
CSS while Surface Grinding

Program	Comments
`G90 M58;`	absolute mode, enable CSS
`G96 L0 S9000;`	CSS mode, surface speed of 9,000 m/min
`G00 X5 Y200 Z1;`	rapid move to position 1 above part
`G82 X95 Y190 J2 E750 F420;`	plunge grind cycle, wheel surface speed 9,000
`M98 P55555;`	dressing subroutine, wheel speed increases
`G00 X5 Y150 Z2;`	rapid move to position 2 above part
`G82 X95 Y140 J2 E750 F420;`	plunge grind cycle, wheel surface speed 9,000
`M98 P55555;`	dressing subroutine, wheel speed increases
`G00 X5 Y100 Z3;`	rapid move to position 3 above part
`G82 X95 Y90 J2 E750 F420;`	plunge grind cycle, wheel surface speed 9,000
`G97 S8000;`	cancel CSS mode, set wheel speed to 8,000 rpm

Figure 12.23
Results of Example 12.19, CSS while Surface Grinding



Grinding Wheel Position	Wheel Diameter after dressing (mm)	Wheel Spindle Speed (rpm)
1	500	5,730
2	400	7,162
3	300	9,549

CSS While Dressing Wheel (surface/cylindrical grinding application)

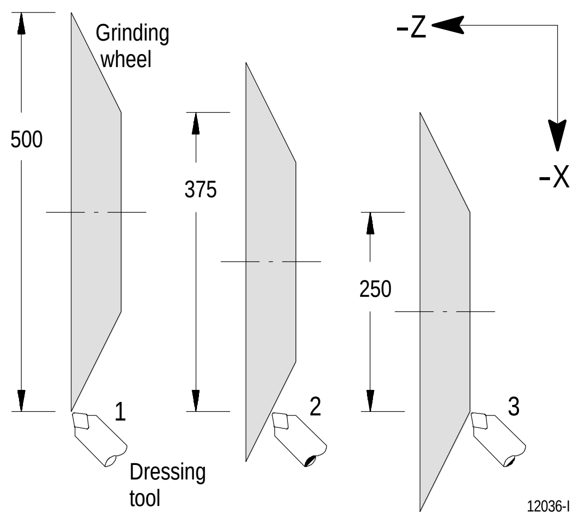
Example 12.20 shows how the wheel spindle speed changes with CSS active as the wheel is dressed to different diameters. The following conditions are assumed for this example:

- the S-word for this machine applies to the grinding wheel spindle
- the X axis has been selected as the CSS axis
- constant K is 318.31 (metric constant)
- the wheel spindle speed is to be limited to 9,000 rpm

Example 12.20
CSS while Dressing a Grinding Wheel

Program	Comments
`G92 S9000;`	limit spindle speed to 9,000 rpm maximum
`G90M58;`	activate absolute prog. mode, enable CSS
`G96 L0 S10000;`	CSS mode, surface speed of 10,000 m/min
`G00 X260. Z165.;`	rapid move to position wheel near dressing tool
`G01 X125. Z80.;`	feed move along X and Z, wheel speed increases
`G00 X260.;`	rapid move from dresser, wheel speed decreases
`G97 S8000.;`	cancel CSS mode, set wheel speed to 8,000 rpm

Figure 12.24
Results of Example 12.20, CSS while Dressing a Grinding Wheel



Grinding Wheel Position	Wheel Diameter (mm)	Wheel Spindle Speed (rpm)
1	500	6,366
2	375	8,488
3	250	9,000 [1]

[1] The calculated spindle speed would be 12,732 rpm at a wheel diameter of 250mm at position 3. When the maximum wheel spindle speed limit of 9,000 rpm is reached, it is not exceeded.

Important: To prevent the grinding wheel from exceeding 9,000 RPM, a limit is placed on the grinding wheel spindle speed (described on page 12-42) with the G92 S9000; block.

CSS While Grinding Part (cylindrical grinding application)

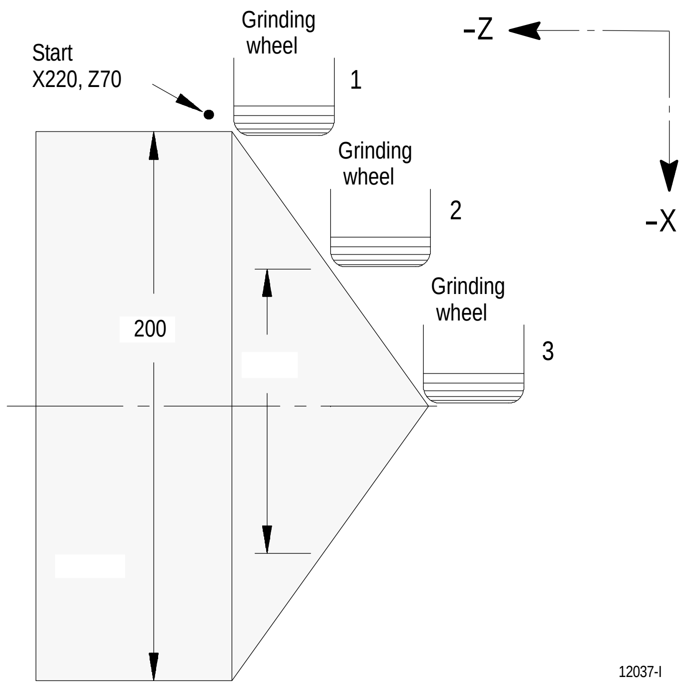
Example 12.21 applies strictly to a cylindrical grinder. It shows how the part spindle speed changes as the diameter of the part being ground changes. The following conditions are assumed for this example:

- the S-word for this machine applies to the part spindle
- the X axis has been selected as the CSS axis
- constant K is 318.31 (metric constant)
- the maximum part spindle speed is 3,500 rpm

Example 12.21
CSS while Grinding a Rotating Part

Program	Comments
`G90 G08;`	activate absolute and diameter prog. modes
`M58;`	enable CSS
`G96 L0 S800;`	CSS mode, surface speed of 800 m/min
`G00 X220. Z70.;`	rapid move to position wheel above part
`G01 X0. Z160.;`	feed move along X and Z, part speed increases
`G00 X220.;`	rapid pull wheel out, part speed decreases
`G97 S400.;`	cancel CSS mode, set part speed to 400 rpm

Figure 12.25
Results of Example 12.21, CSS while Grinding a Rotating Part



Grinding Wheel Position	Part Diameter (mm)	Part Spindle Speed (rpm)
1	200	1,273
2	100	2,546
3	0	3,500 [1]

[1] The calculated spindle speed would be infinite because the diameter of the workpiece at wheel position 3 is zero (0). When the maximum part spindle speed of 3,500 rpm (set in AMP) is reached, it is not exceeded.

Important: To prevent the spindle speed from reaching the maximum RPM, you can set a limit on the spindle speed using a G92 block with an S-word as described on page 12-42.

The surface speed used in this example (800 m/min) is the surface speed relative to a stationary grinding wheel. If an actual surface speed of 800 m/min were desired, the grinding wheel speed and diameter would have to be entered into the equation on page 12-38 and an $S_{program}$ value determined.

For example, if the grinding wheel speed was 1,200 rpm and its diameter was 180 mm, to maintain an actual surface speed of 800 m/min, you would program a surface speed of 121 m/min with the S-word in the G96 block.

12.5.8 RPM Spindle Speed Mode (G97)

In the G97 mode, the spindle revolves at the programmed RPM regardless of the position of the grinding wheel.

For example, to revolve the spindle at 500 rpm, program:

```
G97 S500;
```

Important: In most cases, spindle motion is started and stopped by executing an M code (typically M03 or M04 to start and M05 to stop). Refer to the documentation prepared by your system installer.

The G97 code is modal and remains active until it is cancelled by the G96 code.

When changing from G96 to G97 mode, if an S-word is programmed in the G97 block the spindle changes speed to the RPM value indicated by the S-word. If no S-word is programmed in the G97 block, the control maintains the control uses the current speed of the wheel as the spindle RPM to be maintained.

12.6 Part or Wheel Spindle Orientation (M19)

The control is equipped to perform a part or wheel spindle orient operation. Use this operation to rotate the part or grinding wheel to a given angle. Typically this feature is used to orient the part or grinding wheel for load/unload/removal operations.

Important: M19 Spindle Orientation is applied only to the spindle that is controlled with the programmed S-word.

Two types of part or wheel spindle orients are available:

- Open loop orient - The part or wheel does not use a feedback device for this type of orient. The final destination of the part or wheel when performing an open loop orient is determined by PAL. Typically there is some form of hardware switch used to determine the part or wheel is at the proper position. When the open loop orient is performed, the part or wheel is turned at an AMP defined RPM and in an AMP defined direction.

- Closed loop orient - The part or wheel must be equipped with a feedback device. The final destination of the part or wheel when performing a closed loop orient can be determined in AMP, or entered in a program block requesting an orient. When the closed loop orient is performed, the part or wheel is positioned at an AMP defined RPM. The direction of the orient is as follows:

  - if the part or wheel is turning, the orient completes in the same direction as the part or wheel is currently revolving. If the part or wheel is turning faster than the orient speed defined in AMP, it first slows to that orient speed before performing the orient.

  - if the part or wheel is not turning, the orient is performed in whichever direction that results in the part or wheel reaching the required position using the shortest angular distance.

Refer to the documentation prepared by your system installer to determine which orient your system is equipped to perform. This manual assumes that a closed loop type orient is available.

Both open and closed loop part or wheel orients can be requested either by programming an M19 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 your system installer in AMP.

If a closed loop orient is requested by programming an M19, the option exists to orient the part or wheel 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 part or wheel is positioned relative to an angle of zero that is fixed for a specific machine (refer to documentation prepared by your 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. However, each time you want to orient to a specific angle, you must program an M19 with an S–word. Programming an S–word alone replaces the current modal part or wheel speed used later when the M19 mode is canceled. Cancel the M19 part or wheel orient by programming one of the other part or wheel mode M codes:

M03 - part or wheel rotation clockwise
M04 - part or wheel rotation counter clockwise
M05 - part or wheel stop

12.7 Feedrates

This section covers the following topics:

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 and apply to all G01, G02 and G03 axis motion. If the block requires rapid traverse motion (G00), the programmed feedrate is ignored for that block, but is stored in control memory as the active feedrate.

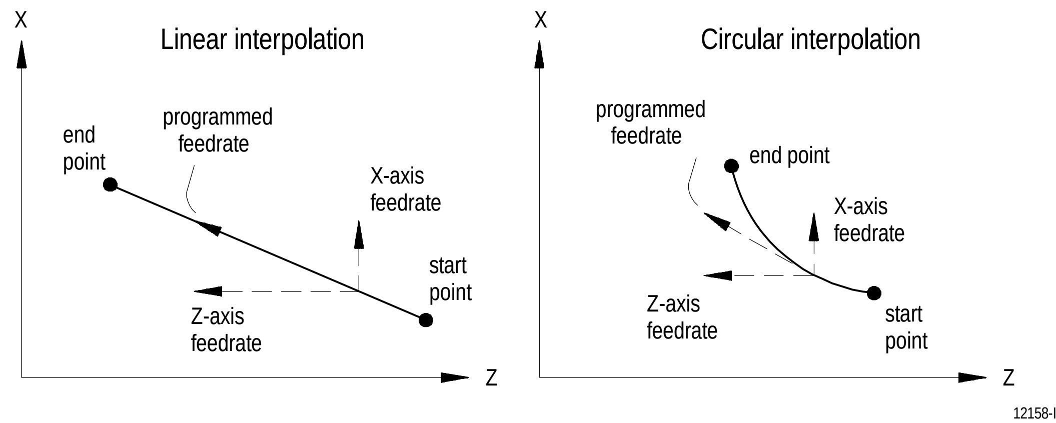
Feedrates are modal, meaning that they remain active in control memory unless replaced with a different feedrate programmed with an F-word.

Feedrate modes are either G95 (grinding wheel distance per workpiece revolution) or G94 (grinding wheel distance per minute). The following table shows the possible feedrate units depending on axis type.

Active G-code	Linear Axis Feed	Rotary Axis Feed
`G71 and G94`	millimeters/min.	degrees/min.
`G71 and G95`	millimeters/rev.	degrees/rev.
`G70 and G94`	inches/min.	degrees/min.
`G70 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 12.26).

Figure 12.26
Programmed Feedrate Executed along the Effective Axis Path



For example, if a feedrate is programmed as F100.0 millimeters per minute and a linear move is made from X0, Z0 to X10., Z10. the feedrate along that 45 degree angular path would be 100.0 mmpm. The actual feed rate of each axis is approximately 70.7 millimeters per minute.

12.7.1 Feedrates Applied During Dresser/Wheel Radius Compensation

When the grinding wheel is offset from a programmed path (as in the case of dresser/wheel radius compensation), the programmed feedrate is applied to the center of the wheel radius for all linear and outside arc paths.

For linear paths, the result is not significant because the speed of the wheel surface relative to the part surface remains the same as the programmed feedrate.

For outside arc paths, the resulting speed of the wheel surface 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.

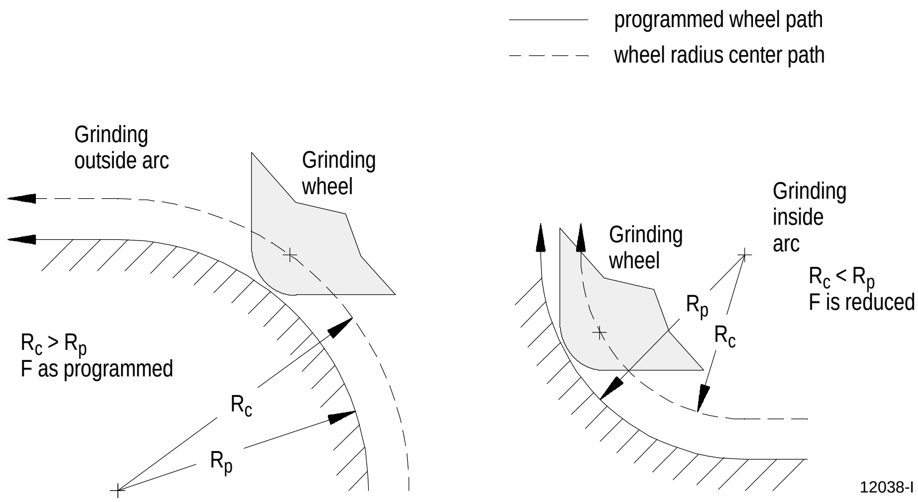
For inside arc paths, the resulting speed of the wheel surface relative to the part surface would be greater than the programmed feedrate. Since this could cause excessive wheel loading and poor grinding performance, the control automatically takes corrective action.

For outside arc paths, the speed of the wheel surface relative to the part surface can be determined using the following formula:

$$\text{Wheel surface speed} = F \times \frac{R_p}{R_c}$$

Where :	Is :
F	programmed feedrate
R_c	radius of the arc measured to the center of the wheel radius
R_p	programmed radius of the arc

Figure 12.27
Arc Feedrates during Dresser/Wheel Radius Compensation



For <u>inside</u> arc paths, the control automatically maintains the <u>programmed feedrate at the wheel surface</u>. The actual wheel radius center feedrate is reduced as needed through the arc path and then return to the programmed feedrate after the arc is completed.

During inside arc paths, the control decreases the wheel 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 wheel with a large radius, the value of R_c/R_p is nearly zero, and the wheel radius center feedrate becomes excessively small.

To avoid this problem, your system installer must set a minimum feed reduction percentage (MFR) in AMP. This sets a minimum feedrate to be used whenever the value of R_c/R_p is very small. If R_c/R_p < MFR, the control reduces the wheel radius center feedrate no more than the MFR percentage.

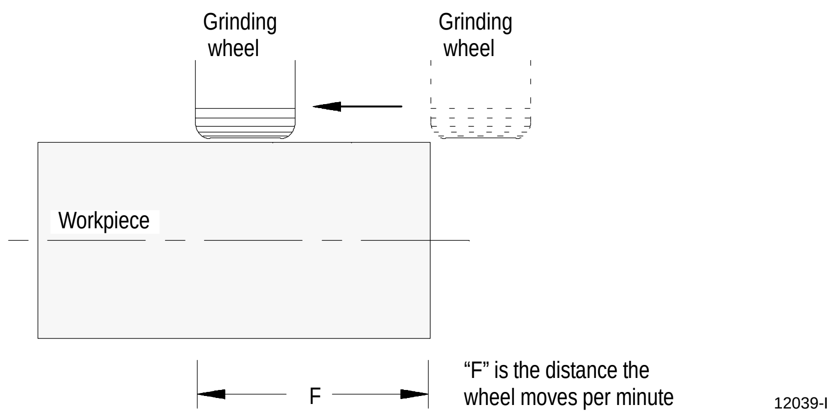
12.7.2 Feed Per Minute Mode (G94)

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 program a feedrate of 55 mm of axis motion per minute program:

```
G94 F55.;
```

Figure 12.28
Feed Per Minute Mode (G94)



When changing from G95 to G94 modes, a feedrate must be programmed in the first G94 block.

Since the G94 code is modal, any F–word programmed in any block after the G94 is considered a feed distance per minute until a G95 is executed.

12.7.3 Feed Per Revolution Mode (G95)

In the G95 mode (feed per revolution), the numeric value following address F represents the distance the axis or axes move (in inches or millimeters) per revolution of the spindle. If the axis is a rotary axis, the F–word value represents the number of degrees the axis rotates per revolution of the spindle.

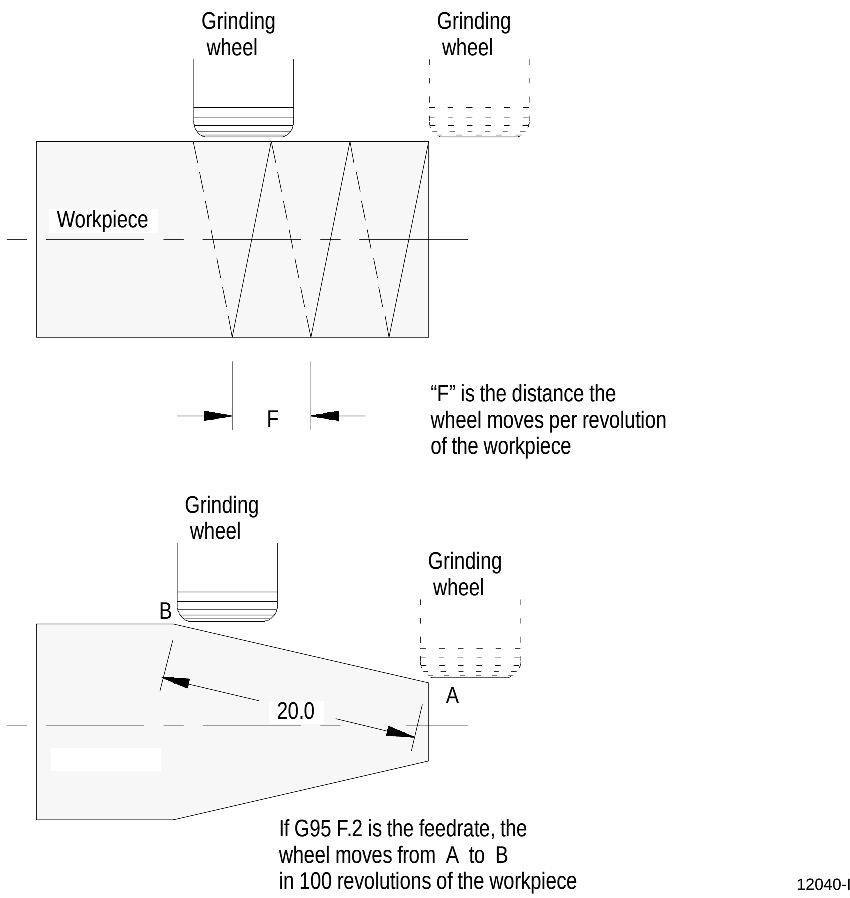
To program a feedrate of 1.5 mm per revolution of workpiece program:

```
G95 F1.5;
```

When changing from G94 to G95 modes, a feedrate must be programmed in the first G95 block.

Since the G95 code is modal, any F–word programmed in any block after the G95 is considered a feed distance per spindle revolution until a G94 is executed.

Figure 12.29
Feed Per Revolution Mode (G95)



12.7.4 Rapid Feedrate

Rapid feedrate is used to position axes to a specified point at a high speed. It is called by executing a G00 followed by an axis motion command. It is also called automatically for some of the motions made by the fixed cycles (see the fixed cycle specifications in chapters 15 and 16).

When more than one axis is commanded to move at rapid feedrate, they are driven together to produce a linear move. The control drives one of the axes at its rapid feedrate and reduces the feedrate of the others as required to insure that all axes start and stop at the same time.

When rapid positioning (G00) is canceled, the control restores the previously commanded F–word feedrate.

Axes can also be driven at their maximum allowable speeds during the jogging operations by holding down the <TRVRS> button while executing a jog move (for details on jogging an axis, see chapter 4).

12.7.5 Feedrate Overrides

<FEEDRATE OVERRIDE> Switch

The <FEEDRATE OVERRIDE> switch on the MTB panel can override:

- the feedrate programmed with an F–word in any of the feedrate modes (G93/94/95)
- the reciprocation feedrate programmed with an E–word during any of the surface or cylindrical grinding fixed cycles (G82 - G88.1)

The <FEEDRATE OVERRIDE> switch has a range of 0%-150% of the programmed feedrate, and alters the programmed feedrate in 10% increments. When set to 0%, the control is effectively in feedhold.

The control checks whether the feedrate resulting from the <FEEDRATE OVERRIDE> switch setting exceeds the maximum 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 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%. They can be enabled by programming an M48 (overrides enabled). See chapter 10 for details.

Important: Ultimately, all switches on the MTB panel are controlled by PAL. Refer to the documentation prepared by your system installer.

<RAPID FEEDRATE OVERRIDE> Switch

The <RAPID FEEDRATE OVERRIDE> switch on the MTB panel can override the rapid feedrate for G00 mode in four increments:

- F1 – % value set in AMP by your system installer
- 25%
- 50%
- 100%

Important: Normally this override switch is not active for any dry run motions (see chapter 7) unless otherwise specified in PAL by your system installer.

Important: This override is also effective for jog moves that use the rapid feedrate (see jogging using the <TRVRS> button in chapter 4).

Important: Ultimately, all switches on the MTB panel are controlled by PAL. See the documentation prepared by your system installer.

Feedrate Override Switches 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%. They can be enabled by programming an M48 (overrides enabled). See chapter 10 for details.

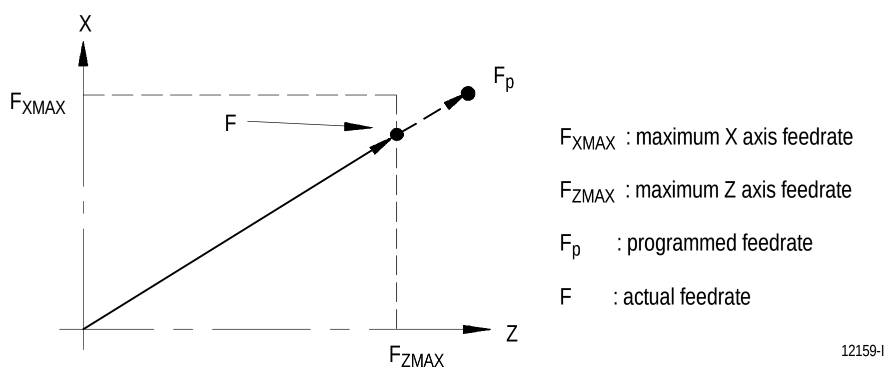
Feedhold

Your system installer can write PAL to allow the activation of a feedhold state through the use of a button or switch. When activated, the control decelerates 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 your system installer.

12.7.6 Feedrate Limits (Clamp)

The maximum allowable speed for each axis is set in AMP. If any axis feedrate exceeds its maximum allowable speed, the control automatically adjusts the feedrate to a value that does not cause axis speed to exceed its set limit.

Figure 12.30
Feedrate Clamp



In Figure 12.30, when the commanded feedrate is F_P, it causes the Z-axis feedrate to exceed the maximum feedrate (F_{ZMAX}). The control then adjusts the feedrate for both axes so that F becomes the actual feedrate.

When the feedrate is “clamped” to a value below the programmed feedrate, the control displays a flashing C next to the current axis feedrate. The displayed axis feedrate is the actual feedrate of the wheel, not necessarily the programmed feedrate.

12.7.7 Rotary Axis Feedrates

The feedrate for a rotary axis is determined in much the same way as for a linear axis.

When programming in G94 feed per minute mode, the rotary axis rotates the programmed number of degrees at the programmed feedrate. Rotary feedrate units are in degrees per minute.

When programming in G95 feed per revolution mode, the rotary axis rotates the programmed number of degrees at the programmed feedrate. Rotary feedrate units are in degrees per revolution of the spindle.

When G01, G02, or G03 are active, the control uses the programmed feedrate to calculate the angular velocity of the rotary axis. This feedrate is still limited to the maximum grinding feedrate (feedrate clamp) as determined in AMP.

If you program a rotary axis move in a block with other axis feedrate moves, all axes move simultaneously at independent feedrates so that the rate along the effective path is equal to the programmed feedrate (see Figure 12.26). All axes must start and stop at the same time, so the control may reduce the feedrate along the effective path to avoid exceeding the maximum rotary axis feedrate

When the control is in rapid mode (G00), the feedrate for the rotary axis is the rapid feedrate for that axis as defined in AMP by your system installer. If other axes are moving in the same block, the control drives one of the axes at its rapid feedrate and reduces the feedrate of the others as required to insure that all axes start and stop at the same time (see page 12-57 for details).

ATTENTION: Remember when programming a rotary axis that the programmed feedrate is in units of angular velocity. This means that the actual feedrate depends on the distance from the center of rotation of the rotary axis.

12.8 Special AMP Assigned Feedrates

This section covers the following topics:

You can select special feedrates that are assigned in AMP. The feedrate for rapid moves described on page 12-62 is such a feedrate as is the feedrate for dry run described in chapter 7. This section covers the feedrates assigned in AMP for the single-digit F–word and the external feedrate switch.

12.8.1 Single-Digit F-words

Program a one-digit numeric value (1-9) following the F–word to select various pre-set feedrates. The digits 1-9 have corresponding numeric feedrate values. These feedrates are defined in AMP by your system installer. They also can be entered using the single-digit feedrate table described on page 3-27.

When programming a single-digit F–word it must be programmed with <u>no decimal point</u> and a single digit. For example, if you program F1.0 instead of F1, a feedrate of 1.0 is assigned in the current feedrate mode.

Important: Single-digit feedrates are always entered as per minute feedrates (IPM or MMPM) regardless of the current feedrate mode. When a single-digit feedrate is programmed, the control automatically switches to the IPM or MMPM 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 a single-digit feedrate (F1-F9) is programmed but has a value of zero in the single-digit feedrate table, an error results indicating that no feedrate has been programmed.

Single-digit F–word feedrates are affected by the setting of the <FEEDRATE OVERRIDE> switch. 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 grinding feedrate (not normally the rapid feedrate) set in AMP by your system installer. F0 generates an error if the feature is not used.

12.8.2 External Deceleration Feedrate Switch

Your system installer can 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 your 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 grinding feedrate (G01, G02, G03, etc.) use a special feedrate assigned in AMP. When this feature is active, any axis moves that are to take place at a rapid feedrate (G00, etc.) 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 grinding moves does not exceed the maximum grinding feedrate.

If you use this feature simultaneously with the dry run feature the feedrates that are assigned to the external deceleration 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 FEEDRATE OVERRIDE> switch settings and operates as if the switches are set at 100%. Blocks that are programmed to move at the rapid feedrate are still executed in the rapid mode.

Typically this feature protects 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 can result if a sudden stop is performed at the overtravel or the machine can 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 is much less.

If the current feedrate is less than the feedrate set for the external deceleration feature, it is accelerated to the external deceleration feedrate. This can cause problems with part finish or can damage the wheel.

ATTENTION: Your system installer can 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 deceleration and replace all feedrates in the program with the external deceleration feedrates. This can result in damage to the machine, part, or injury to the operator.

12.9 Automatic Acceleration/Deceleration

This section covers these topics:

Three types of axis acceleration/deceleration are available:

- Exponential Acc/Dec
- Uniform or Linear Acc/Dec
- S–Curve Acc/Dec

Use these to produce smooth starting and stopping of axes and prevent damage to the machine resulting from jerking movements.

Your system installer determines the acc/dec parameter type (exponential or linear) for some manual motion types. To determine which motion types are configuratble, refer to the following table. Refer to your system installer's documentation for more information about how your system is configured.

Refer to Table 12.A to determine the type of acceleration/deceleration performed for manual motion and programmed moves.

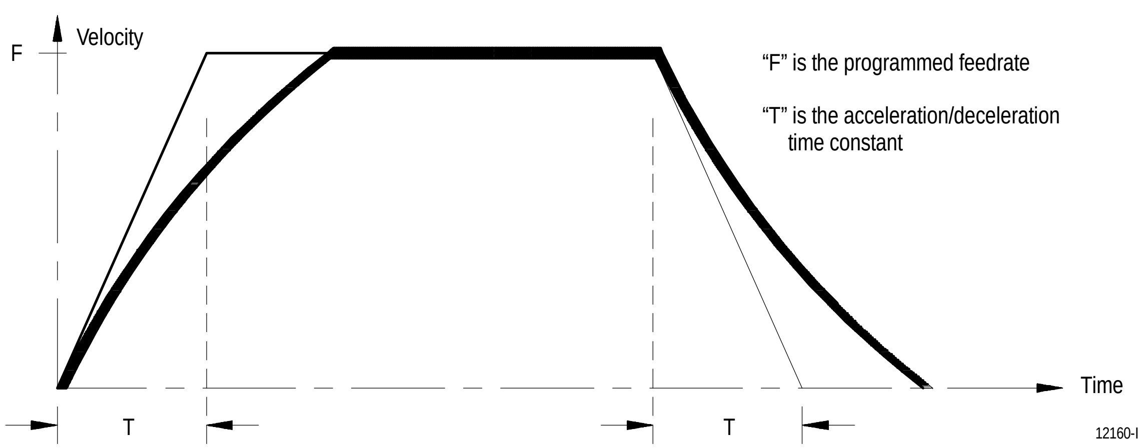
Table 12.A
Acc/Dec Type Performed with Manual Motion and Programmed Moves

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	✓			
Arbitrary angle moves (i.e., hand-pulse generator and continuous and incremental motion)	✓			
Homing	✓			
All programmed moves except for G00 and exact stop			✓	
Manual continuous motion		✓		
Manual incremental motion		✓		
PAL axis mover		✓		
All moves programmed in G00 (positioning) mode				✓

12.9.1 Exponential Acc/Dec

To begin and complete a smooth axis motion, the control uses an exponential function curve to automatically accelerate/decelerate an axis. Your system installer sets the acceleration/deceleration time constant “T” for each axis in AMP. Figure 12.31 shows axis motion using exponential acc/dec.

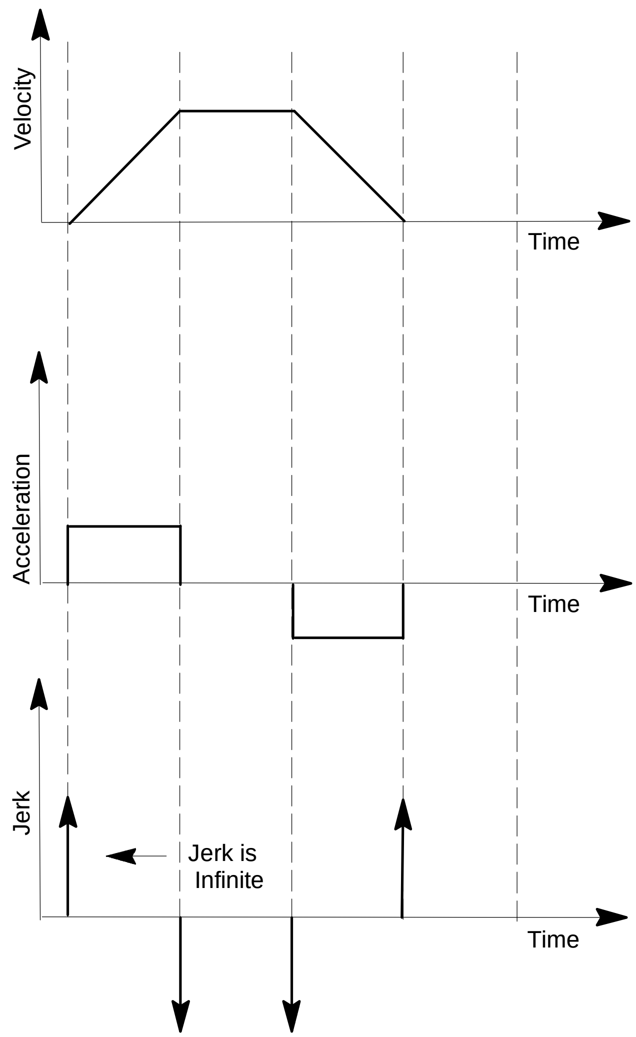
Figure 12.31
Exponential Acceleration/Deceleration



12.9.2 Linear Acc/Dec

Axis motion response lag can be minimized by using Linear Acc/Dec for the commanded feedrates. Your system installer sets Linear Acc/Dec values for interpolation for each axis in AMP. Figure 12.32 shows axis motion using Linear Acc/Dec.

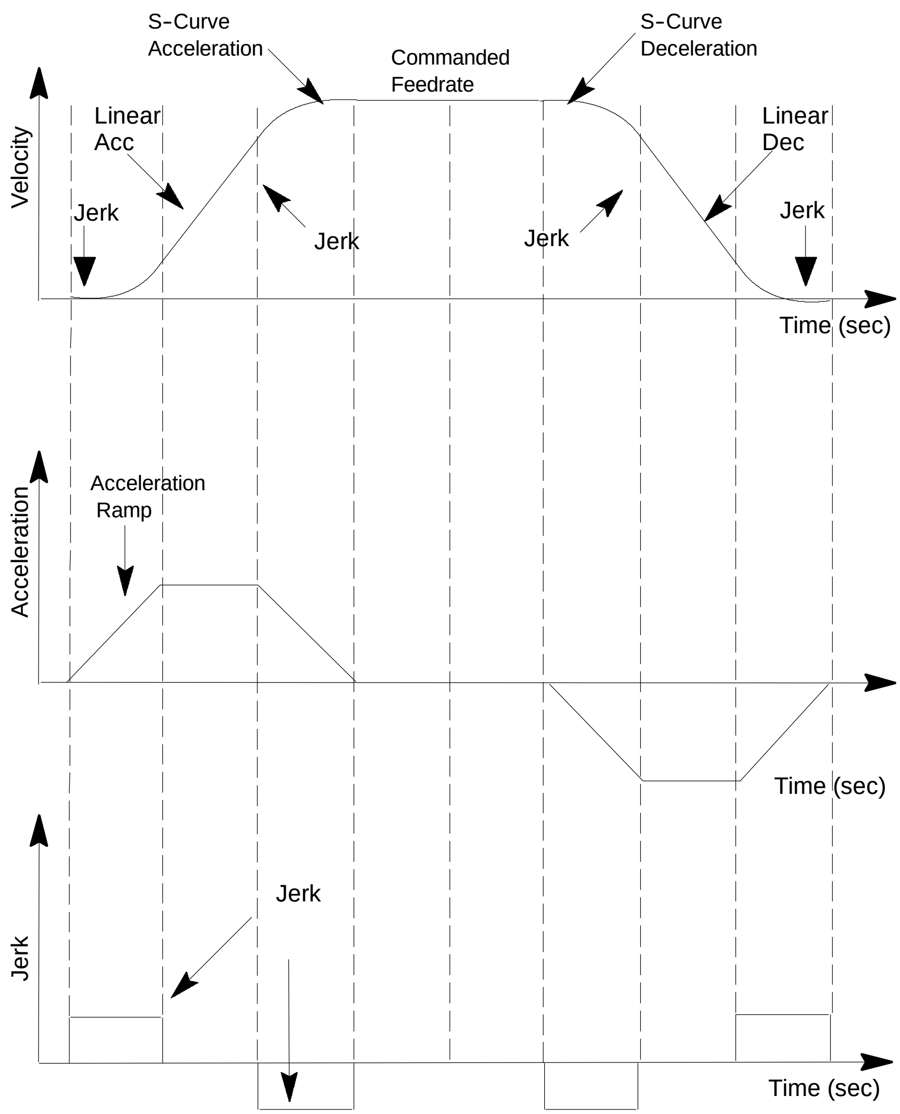
Figure 12.32
Linear Acc/Dec



12.9.3 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 12.33 shows axis motion using S–Curve Acc/Dec.

Figure 12.33
S-Curve Acc/Dec



12.9.4 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 12.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

[1]Linear/Exponential is a function of Positioning Acc/Dec. If Exponential is AMPed, this is the acc/dec type, otherwise, the type is Linear.
[2]S-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./sec3 or mm/sec3	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/sec^3, 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 12.22
Allowable Programmed Range

```
#1 = 100000;
G48.5 X #1;
```

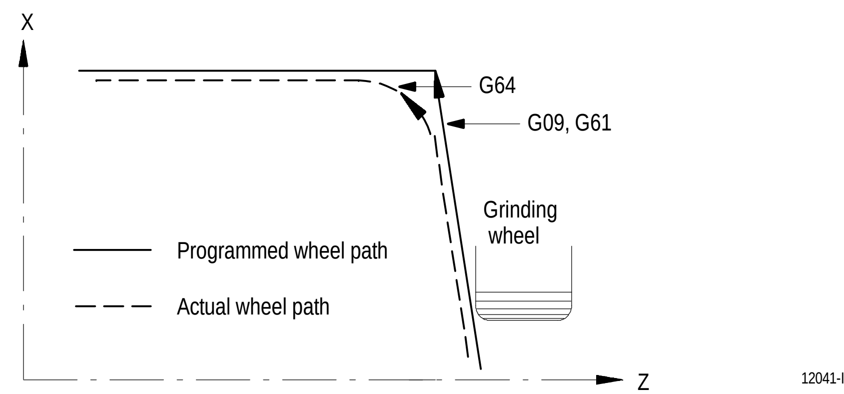
Important: The part program G48.n adjustments to Acc/Dec Ramps are not applied to jog moves. The AMPed Linear Acc/Dec mode rates are used when Manual Acc/Dec mode is linear.

12.9.5 Precautions on Corner Grinding

When exponential acc/dec is active, the control automatically performs acc/dec to give a smooth acceleration/deceleration for grinding wheel motion. However, there are cases in which exponential acc/dec can result in rounded corners on a part during grinding.

As illustrated in Figure 12.34, this problem is most obvious when the direction of grinding changes from the X-axis to the Z-axis. In this case, the X-axis is moving at nearly the full grinding feedrate while the Z-axis is moving very slowly. The X-axis decelerates as it completes its move and as soon as it reaches the AMP defined in-position band, the Z-axis begins accelerating to make its commanded move. Since the Z-axis begins motion before the X-axis finishes, a slight rounding results.

Figure 12.34
Rounding of Corners



Use the following G–codes to eliminate corner rounding:

Exact Stop (G09 - - non-modal)

If a programmed motion block includes a G09, the axes move to the commanded position and decelerate to zero speed; the control waits until the following error for each of the programmed axes is within the AMPed in-position band before completing the current block.

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. It is the modal equivalent of G09.

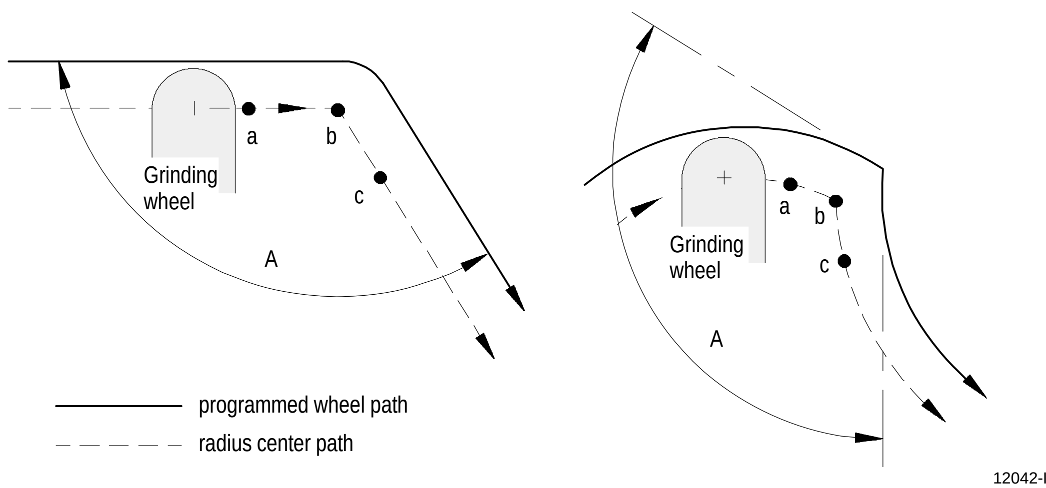
Cutting Mode (G64 - - modal)

G64 establishes the cutting mode. This is the normal mode for axis motion and generally is selected by your system installer as the default mode active on power-up. When active, motion commands begin as soon as the motion command of the previous block has been completely issued. G64 does not wait for in-position band.

Automatic Corner Override (G62 - - modal)

In compensation mode (G41/G42), the load on the grinding wheel increases while moving inside a corner. If the G62 automatic corner override mode is active, the control automatically overrides the programmed feedrate to reduce the load on the wheel.

Figure 12.35
Automatic Corner Override (G62)



When the corner angle (A) is larger than the value set for “min. angle for corner override” in AMP, the programmed feedrate is overridden from point “a” to point “b,” and from point “b” to point “c.”

Your system installer sets the following values in AMP:

- Min angle for corner override –– minimum angle (A) between programmed paths before corner override is activated
- Corner override distance (DTC) – vector distance from and of current move (b) to point on programmed path (a) where corner override is activated

- corner override distance (DFC) – vector distance from end of current move (b) to point on programmed path (c) where corner override is deactivated
- corner override percent – amount that feedrate is to be reduced once corner override is activated

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 cancel each other if programmed. Be aware that G09 is non-modal.

12.9.6 Spindle Acceleration (Ramp)

Your system installer can 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 your system installer can also, in effect, generate a spindle "ramp" for even smoother spindle acceleration. Refer to documentation prepared by your system installer.

12.9.7 Controlling Spindles (G12.1, G12.2, G12.3)

Use the G12 code to program the active controlling spindle for all programmed axes motions for features and modes requiring spindle operation. The G12 code is modal as only one spindle can be the controlling spindle.

- G12.1 — Spindle 1 Controlling
- G12.2 — Spindle 2 Controlling
- G12.3 — Spindle 3 Controlling

Table 12.C lists the allowed spindle capabilities.

Table 12.C
Spindle Capabilities

Control Type	Number of Spindles	Spindle Type
9/240	1	Spindle 1
9/260	2	Spindle 1, Spindle 2
9/290	3	Spindle 1, Spindle 2, Spindle 3

Spindle 1, as well as spindles 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 primary 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 count-per-rev device.

Important: On the 9/260 and 9/290 controls, if spindles 2 and 3 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"

12.9.8 Spindle Orientation (M19, M19.2, M19.3)

For each possible spindle configured, 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 load/unload operations, to position a chuck for automatic chuck wrench operation, etc. This orient operation is not the same as using a spindle as an axis for positioning. See virtual C axis, section 18.5. An orient operation is performed separately from axis motions and cannot be interpolated with 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 RPM.

If the spindle is:	the orient will:
turning	complete in the same direction as the spindle is currently revolving. If the spindle is turning faster than the orient speed defined in AMP, it first slows to that orient speed before performing the orient.
not turning	be performed in whichever direction that results in the spindle reaching the required position by using the shortest angular distance.

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 your system installer's documentation to determine which orient your system is equipped to perform. This manual assumes that a closed-loop type orient is available. If an open-loop orient is used, refer to the system installer's 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. However, each time it is necessary to orient to a specific angle, an M19 with an S-word must be programmed. Programming an S-word alone causes a decode error, "ILLEGAL ANGLE VALUE." Cancel the M19 spindle orient 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

12.9.9 Spindle Direction (M03, M04, M05)

Use the spindle directional M-codes to program each configured spindle program controlled spindle rotation.

Table 12.D lists the spindle direction codes.

Table 12.D
Spindle Directional Codes

Spindle Type	Directional Code	This means:
Spindle 1	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 12.23
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 M03.3 S10`	Spindle 2 clockwise 10 rpm Spindle 3 counterclockwise 10 rpm

Important: On the 9/260 and 9/290 controls, if directional M-codes for spindles 2 and 3 are programmed but spindles 2 and 3 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

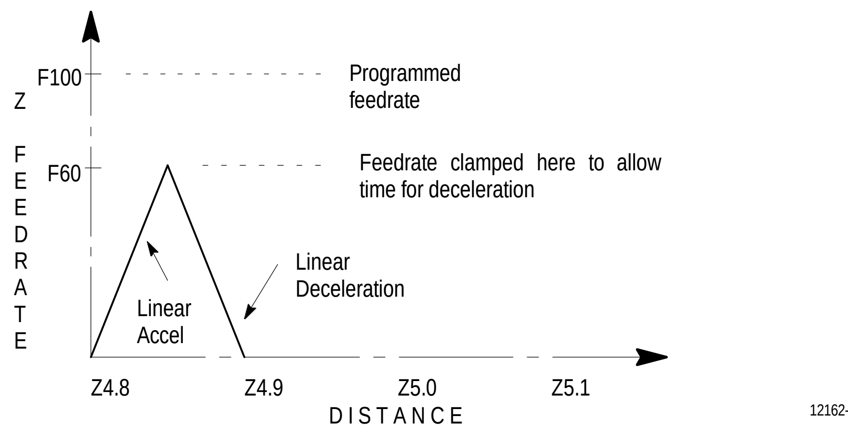
“SPINDLE 3 NOT CONFIGURED”

12.9.10
Short Block Acc/Dec Check G36, G36.1

In the control's default mode (G36), the Acc/Dec feature sometimes limits axis feedrates far below the programmed feedrate. This occurs when the length of axis motion in a block is short relative to the length of time necessary to accelerate and decelerate the axis.

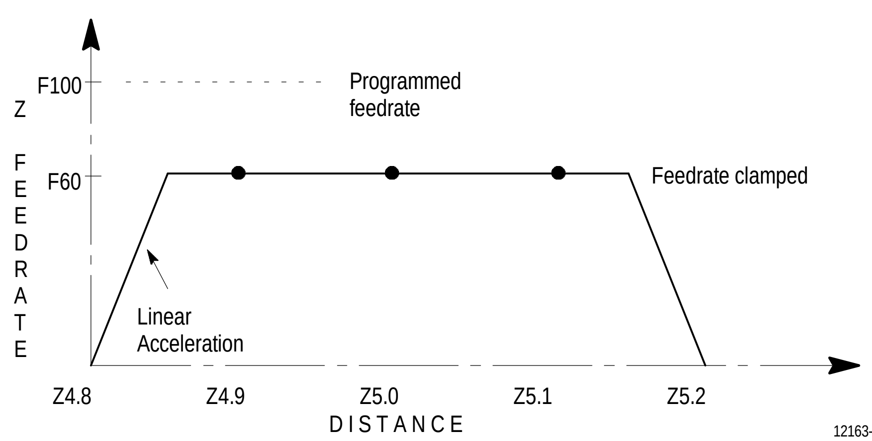
In the default mode (G36), the control limits the axis feedrate in any block to the maximum speed from which it can properly decelerate to a stop before that block ends. For example consider the following velocity profile of an axis moving from Z4.8 to Z4.9.

Figure 12.36
Programmed Feedrate Not Reached



Normally this causes no problem. However, in cases where a series of very short axis moves in separate blocks exist, this limitation to the feedrate can cause finish problems as well as increased cycle time. Figure 12.37 shows the velocity profile that would result from a series of short Z axis moves from 4.8 to 4.9 to 5.0 to 5.1 to 5.2.

Figure 12.37
Feedrate Limited Because of a Series of Short Moves

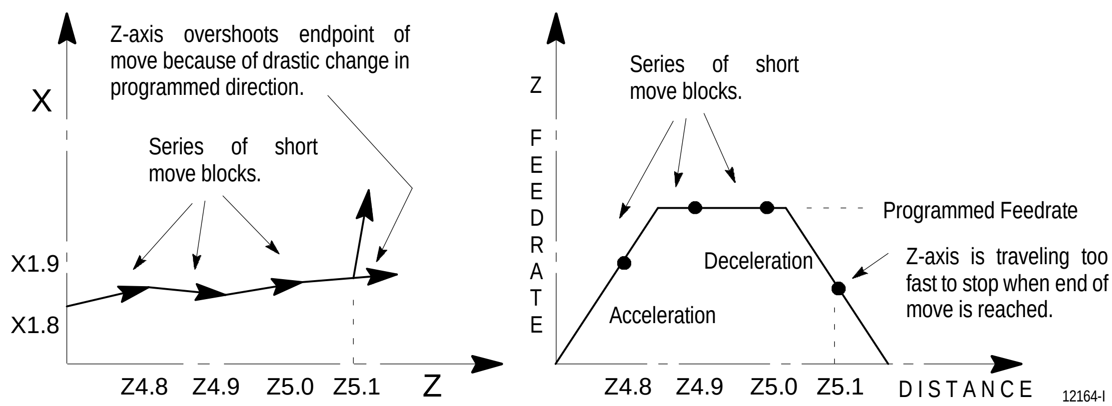


To avoid this feedrate limitation, the short block Acc/Dec clamp can be disabled by programming a G36.1. In this mode, the control assumes that no rapid decelerations are required and allows axis velocities to go higher than they otherwise would. Activate G36.1 mode only when:

- no sudden changes in programmed feedrate within consecutive short motion blocks exists (this includes requesting a feedhold or cycle stop)
- no drastic change in programmed direction is present within the short blocks

If any of the above conditions are not met during the G36.1 mode, the control can overshoot positions since the axes do not have time to decelerate. For example, consider the following position and velocity plots if a drastic change in direction is requested after the move from Z5.0 to Z5.1 when in G36.1 mode (see Figure 12.38) . The position Z5.1 is overshot and the axis would have to reverse direction to reach proper position.

Figure 12.38
Drastic Change in Direction while in Short Block Mode (G36.1)



ATTENTION: The programmer must consider the direction and feedrate transitions from block to block when the short block acc/dec check is disabled (G36.1 mode). 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 only be in short block check disable mode (G36.1) when executing a series of fast short blocks that contain only slight changes in direction and 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 clamp Enable
- G36.1 – Short Block Acc/Dec clamp Disabled

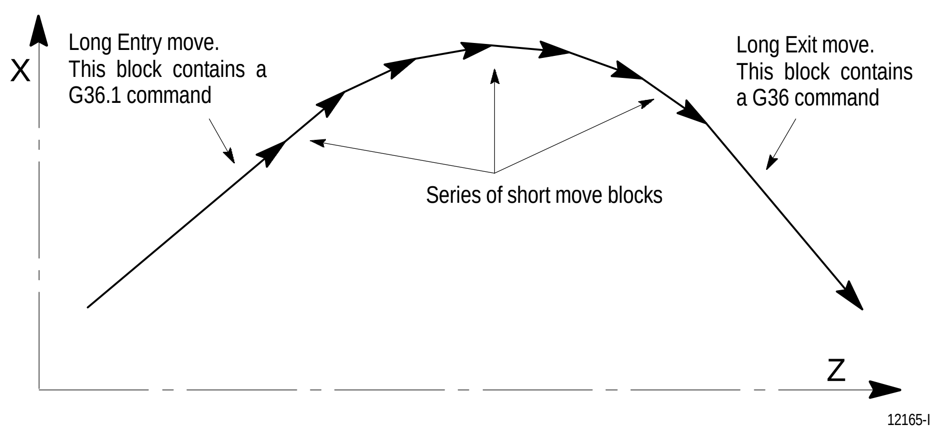
G36 is the default mode and is established at power-up, E-Stop reset, and end of program (M02, M30, or M99). The recommended method of programming G36 and G36.1 is to program a relatively long entry and exit move into and out of the mode.

- The entry move should be a long move, in the general direction of the first short move, and at the same feedrate as the first short move. 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 move, and at the same feedrate as the last short move. This exit move should be long enough for the axes to decelerate properly without overshooting their end points. Program the G36 code in this exit block.

Figure 12.39 shows the recommended entry and exit moves for short block acc/dec clamp disable mode.

Figure 12.39
Entry and Exit Move to/from Short Block



12.10 Dwell (G04)

This section covers the following topics:

The G04 command delays the execution of the next data block. Dwell period is specified in either of two types.

- Seconds
- Number of spindle revolutions

The type used is normally dependant on the feedrate mode (G94 or G95) active at the time. The type can also be permanently fixed to "seconds" regardless of G94 or G95 mode, by setting the proper AMP parameter.

A dwell cannot be executed in the G93 inverse time feed mode.

12.10.1 Dwell - Seconds

In the G94 mode (feed per minute) G04 suspends execution of the commands in the next block for a specified length of time in seconds.

```
G94G04  P__;    X__;    U__;
```

Program 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 block. The allowable dwell time is 0.001 - 99999.999 seconds.

When programming a dwell in seconds your 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 automatically skips 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).

12.10.2 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 spindle to turn a specified number of revolutions.

```
G95G04  P__;    X__;    U__;
```

Program 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 block. The allowable range is 0.001 - 99999.999 revolutions.

12.11 Mirror Image (G50.1, G51.1)

There are two types of mirroring:

- programmable mirror image

  This is activated through programming a G50.1 and G51.1

- manual mirror image

  This is activated through PAL or the {FRONT PANEL} softkey

Programmable Mirror Image (G50.1, G51.1)

Use the programmable mirror image feature to mirror (duplicate yet reversed) axis motion commands about some defined plane. Activate this feature using the G51.1 code and cancel it using the G50.1 code. Mirroring takes place about the axis position specified in the G51.1 code.

The format for the G51.1 code is:

```
G51.1X__ Z__ ;
```

The axis motion commands in any following blocks are executed with the motion direction reversed (including incremental moves) as if a mirror were placed on the programmed point perpendicular to the axis. The G51.1 code is modal and remains in effect until cancelled by a G50.1.

If only one axis is programmed in the G51.1 block, the mirroring plane is perpendicular to that axis. If more than one axis is programmed, a mirror plane is established perpendicular to each. Axis moves is mirrored relative to the mirror point defined for each axis.

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.

```
G50.1X__ Z__ ;
```

The control cancels the mirror feature only for those axes that are programmed in the G50.1 block. Axes not programmed in the G50.1 block remain mirrored.

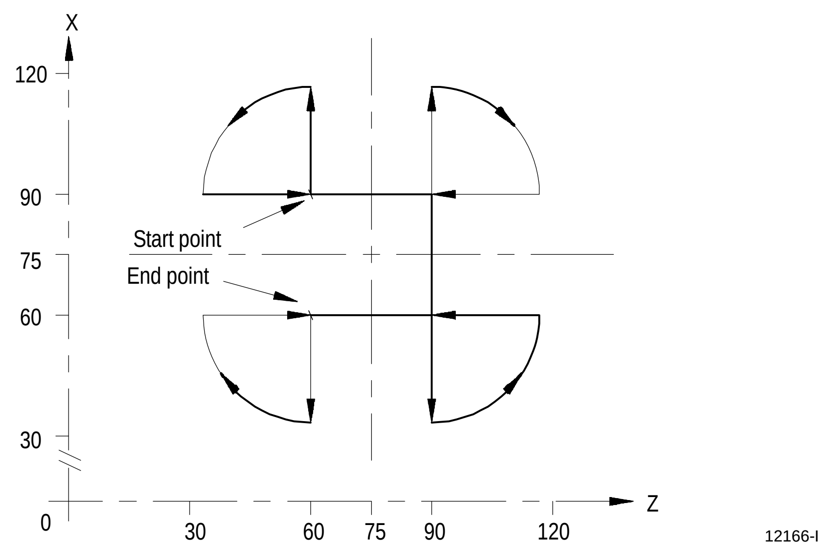
It may be necessary to program a numeric value with the axis words in a G50.1 block to avoid a syntax error (depending on how AMP was configured by your system installer). In any case, there is no significance to any numeric values programmed with the axis words. These values are ignored by the control.

Example 12.24
Programmable Mirror Image

Main Program	Comment
`(Mirror);`	comment block, main program
`G00G90;`	rapid positioning, absolute mode
`M98P8500;`	call subprogram 8500
`G51.1Z75.;`	mirror active on Z
`M98P8500;`	call subprogram 8500
`G51.1X75.;`	mirror active on X (and Z)
`M98P8500;`	call subprogram 8500
`G50.1Z0;`	cancel mirror on Z (active on X only)
`M98P8500;`	call subprogram 8500
`G50.1X0;`	cancel mirror on X (no mirroring)
`M30;`	

Subprogram	Comment
`O8500;`	program number
`G00G90Z60.X90.;`	rapid to start point
`G01X120.F.1;`	move 1
`G03Z30.X90.R30;`	move 2
`G01Z60.;`	move 3
`M99;`	return from subprogram

Figure 12.40
Programmable Mirror Image, Results of Example 12.24



When the mirror image function is active on only one of a pair of axes, the control:

- executes a reverse of programmed G02/G03 arcs. G02 becomes counter-clockwise and G03 becomes clockwise.
- activates a reverse of programmed G41/G42 compensation. G41 becomes compensation right and G42 becomes compensation left.

Manual Mirror Image

In addition to the programmable mirror image feature, the control can also be equipped with an optional mirror image switch, installed by your system installer, that activates the manual mirror image feature. This feature can also be activated using the {FRONT PANEL} softkey with the appropriate PAL programming.

The manual mirror image feature differs from the programmable mirror image feature in that when using manual mirror image 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 also passes through the zero point of the currently active work coordinate system.

The mirrored plane is fixed and cannot be moved from along the selected axis. This mirrored plane is the equivalent of programming a programmable mirror image and using all zero values for the axis words.

Your system installer has the option to install a switch for each of the available axes. What axes are mirrored with what switches depend on the PAL program for your system. You can mirror about more than one axis using more than one manual mirror image switch at the same time or one switch can control more than one axis. Refer to documentation prepared by your system installer for details.

Important: You can use programmable mirror image at the same time as manual mirror image. When this is done the coordinates for the programmable mirror image are calculated first followed by the manual mirror image.

12.12 Axis Clamp

This feature disables the axis position display and lets an axis 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. Your system installer determines how the axis clamp feature is enabled in PAL. Refer to the documentation prepared by your system installer for details.

When an axis is clamped, the control freezes the axis position display values. Any axis drift or movement caused by some external force does not generate any corrective response from the axis servo. This prevents the servo from trying to move an axis that may have been mechanically clamped to prevent movement.

Any axis movement when that axis is clamped is added to the current value of the following error and can be viewed on the screen displaying following error (refer to the integration/maintenance manual). If the axis following error exceeds the allowable maximum following error (set in AMP), an error is generated and the control goes into E-Stop.

When the axis is unclamped, the control position display is reactivated and the servo returns the axis to the necessary position for zero following error.

12.13 Dual Axis Operation

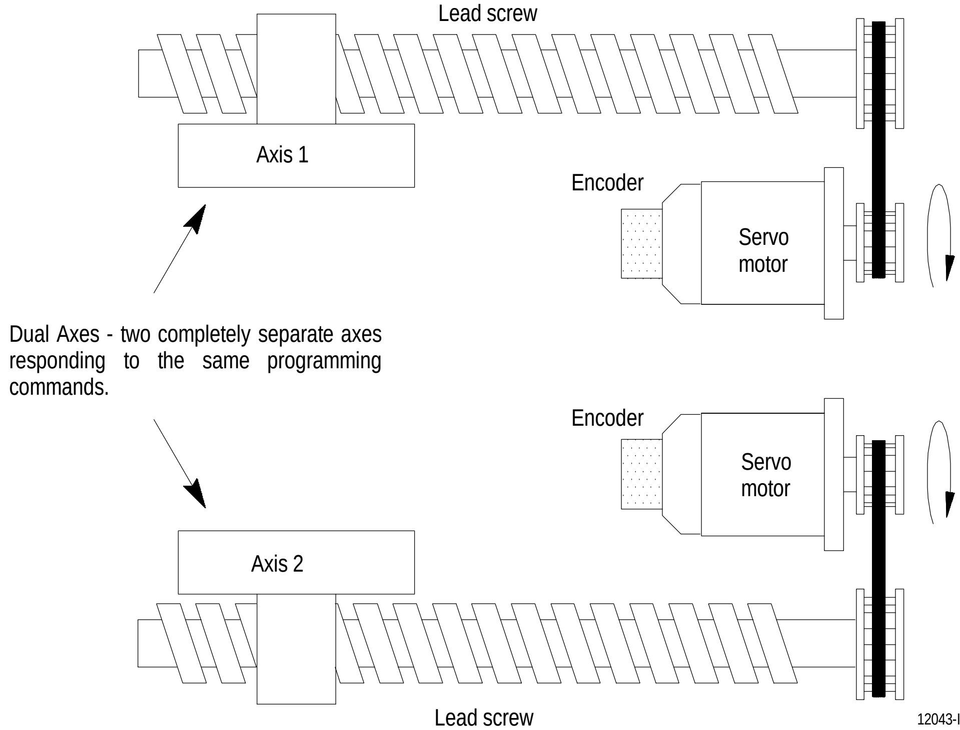
This section covers these topics:

The dual axis feature lets you simultaneously control multiple axes while commanding only one. The command, either for a manual jog or a programmed motion, is automatically directed to the other axes in the dual group. It differs from the split axis feature of the control in that the split axis feature controls a <u>single axis</u> positioned by two servo motors.

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 the documentation prepared by your system installer to see if the dual axis option has been purchased and set up for your machine.

Figure 12.41 shows a typical configuration for dual axes.

Figure 12.41
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:

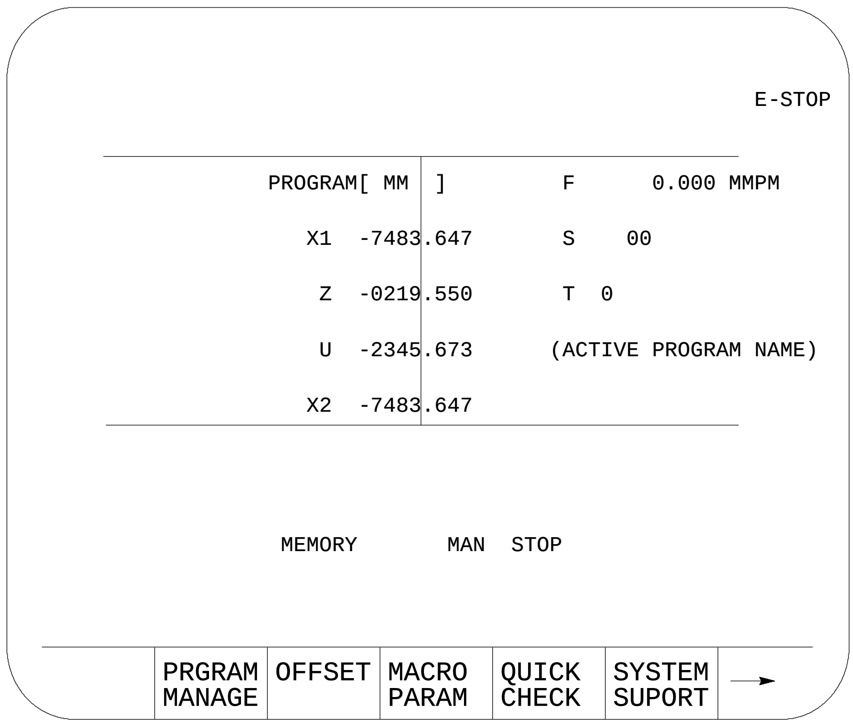
- Master Axis – A master axis is the name used to command the axes in a dual group

- 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 12.42 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 your system installer.

Important: A dual axis cannot be programmed as a reciprocating axis. Any attempt to program a G81 or any cycle that generates reciprocation on a dual axis generates an error.

Figure 12.42
Axis Position Display for Dual X-Axis



12.13.1 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 can 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 can only be parked or unparked when the control is in cycle stop and end-of-block state. Also the control cannot 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 if any one of the above requirements is not true, the control ignores the request to park/unpark the axis.

ATTENTION: 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 be 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 documentation prepared by your system installer for details on how axes are parked.

Important: Some systems have special parking requirements when homing axes in a dual group. Refer to the next section for details on homing dual axes.

12.13.2 Homing a Dual Axis

There are two different methods to home axes in a dual axis group. Your system installer determines through PAL which method is available to you. 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.

Your 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 documentation prepared by your system installer 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 on an individual method. Refer to chapter 4 for details on how to request a manual home operation.

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 your system installer can define different feedrates and different home positions for each axis in the dual group.

With proper PAL programming, your system installer can configure all axes in the dual axis group to home when the request is made to the master axis. If this homing method is used, all unparked axes are homed together. Refer to chapter 4 for details on how to request a manual home operation and page 12-27 on how to request an automatic home operation (G28).

12.13.3 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 individual axis feedrates can be modified, depending on the distance each axis must travel from start to end of the requested move.

Your system installer can assign different maximum positioning, external deceleration, 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 these 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 documentation prepared by your system installer and page 12-79 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 11 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: You can use the PAL axis mover feature if you want to position dual axis group members separately without requiring any parking. Refer to your PAL reference manual and the documentation prepared by your system installer for details.

Invalid Operations on a Dual Axis

The following table contains a list of the features that are not compatible with dual axes. You must execute one of these features on a dual axis, only the AMP defined master axis can be used. <u>All other axes in the dual group must be parked</u>. See the documentation prepared by your system installer to determine master axis and slave axis names.

Table 12.E
Features NOT Compatible with Dual Axes

G-code	Feature
G20, G24	Single Pass Rough Grinding
G33, G34	Thread Grinding Modes
G31-G31.4	External Skip Functions
G37-G37.4	Automatic Wheel Gauging Skip Functions
G82-G86	Surface Grinding Fixed Cycles
G82-G88	Cylindrical Grinding Fixed Cycles

12.13.4 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 describes the difference in operation of a dual axis when it concerns offsets. How to activate/deactivate and enter these offset values is not described here unless some change specific to a dual axis occurs. Refer to the section in chapter 3 of this manual describing 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 want 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 of whether they are parked. 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 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.

You can create different G92 offset values for each axis. 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

You can perform a set zero operation 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.

Wheel Compensation

Only one wheel diameter can be active at any one time. Any offset created by compensation affects all axes in the dual group.

Wheel Length Offsets

Functionality is the same as for any other axis; independent values can be entered for each axis in the dual group. A T-word selects the length offset number, and the wheel length offset is activated for each axis. Refer to chapter 13 for details on activating wheel length offsets.

Additional programming for the G10L10 codes is necessary when a dual axis is the wheel length axis. Since each axis in the dual group can have separate offset values, the G10 block must contain an individual name for each axis being assigned a value. Just programming the master axis name assigns values only to the master axis. Your system installer assigns names to all axes in the dual group in AMP. Refer to the documentation prepared by your system installer for details.

END OF CHAPTER

Chapter 13

Wheel Length Offsets

13.0 Chapter Overview

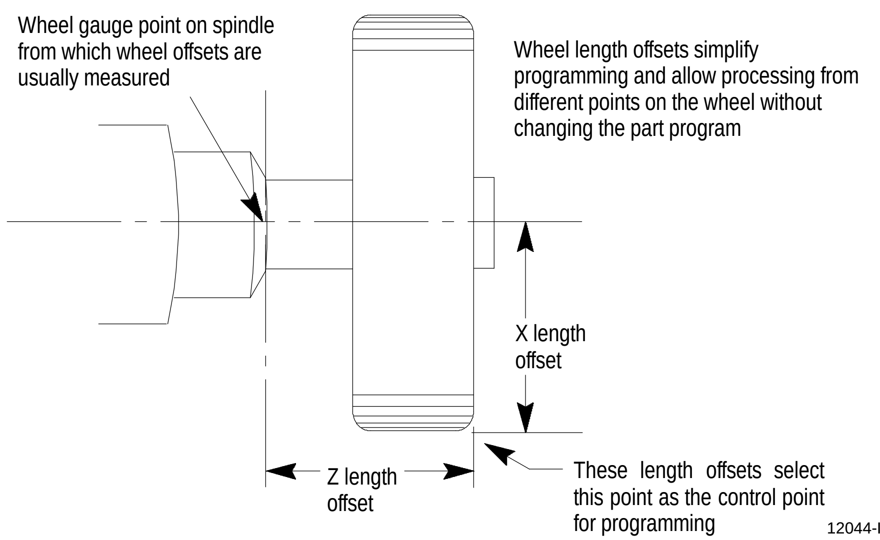
This chapter describes how to select and activate wheel length offsets. Some grinding applications require the use of wheel length offsets in conjunction with dresser/wheel radius compensation. For details on selecting and activating dresser/wheel radius compensation, see chapter 15.

This chapter covers these topics:

13.1 T Words and Wheel Length Offsets

Modern grinders typically require the selection of multiple control points on the wheel for grinding different contours and dressing different profiles. Use the wheel length offset feature to shift the control point to the various control locations on the grinding wheel.

Figure 13.1
Typical Grinding Wheel Length Offset to Select Control Point



The control can store up to 32 wheel length offsets for each axis. Your system installer configures the actual number of available offsets on your system in AMP. Each offset can select a different control point on the wheel. Typically, each time you dress a different profile into the wheel, you need to activate different length offsets.

13.1.1 Selecting Wheel Length Offsets (T Word)

Program a T word to select and activate a wheel length offset. You can program a T word in any block in a part program.

Important: When you activate the miscellaneous function lock feature, the control ignores M, B, S, and T words in the part program with the exception of M00, M01, M02, M30, M98, and M99. You can activate this miscellaneous function lock feature through the front panel screen (as described on page 2-11) or by an optional switch installed by your system installer.

The format for a T word is as follows:

```
Trrll;
```

Where:	Is:
`rr`	the dresser/wheel radius and orientation offset number. The radius/orientation offset number is the first two digits of the T word. See chapter 3 for details on radius/orientation data.
`ll`	the wheel length offset number. The wheel length offset number must be the last two digits of the T word. See chapter 3 for information on wheel length offset values.

An offset number selected as zero (for either length or radius/orientation) selects an offset value of zero and effectively cancels wheel length offsets and dresser/wheel radius compensation. Dresser/wheel radius compensation is not canceled when a radius offset number of zero is programmed (G41 or G42 remain active). In this case, the control uses a radius value of zero. You must cancel dresser/wheel radius compensation by programming a G40.

Important: T words are always right-justified. This means that programming a two-digit T word only enters a wheel length offset value and does not specify a radius/orientation offset number. Table 13.A gives some example T words and the resulting offset numbers activated.

Your system installer can also write PAL to automatically select and activate a wheel length and radius/orientation offset number. See your system installer's documentation and the PAL reference manual for details.

Table 13.A
T Words and Resulting Offsets

Program this T word:	to activate this radius/orientation offset:	and this length offset:
T0102;	#01	#02
T1223;	#12	#23
T0;	cancels offset	cancels offset
T123;	#01	#23
T12;	cancels offset	#12
T1;	cancels offset	#01
T1200;	#12	cancels offset
T0012;	cancels offset	#12

From Table 13.A, you can see that you cannot program a T word without inadvertently programming both a length offset and a radius/orientation offset. By not programming one of the offsets, the control assumes an offset of 00 is programmed and cancels any active offset value. We recommend that you always program a T word followed by all 4 digits. This is to make sure that you are not inadvertently canceling one offset while trying to activate another.

Your system installer determines in AMP if the control cancels all wheel length or radius/orientation offsets when you reset the control or when you execute an M02, M30, or M99 end of program block.

13.1.2 Activation of Wheel Length Offsets

Your system installer has the option in AMP to determine exactly when wheel length offsets take effect and when the wheel position updates on the screen to the new shifted location. This manual assumes that your system is configured to immediately shift the coordinate system by the geometry offset amounts, and delay the actual physical move that re-positions the wheel in the current work coordinate system. See documentation prepared by your system installer to determine the method used in your specific system.

The control activates a wheel length offset when it executes any block that contains a T word. Your system installer selects the activation type from the following:

System installer selects this activation type:	Resulting in this display change:	and this axis motion:
immediate shift/ immediate move	position display changes to reflect the grinding wheel's new location	the control generates a move equal to and opposite the offset amount on each axis.
immediate shift/ delay move	position display changes to reflect the grinding wheel's new location	the control does not generate the move activating the offset until a move on the axis is made. Then the offset amount is applied to the endpoint of the move. Each axis that has a length offset must be moved to activate the offset on that axis.
delay shift/ delay move	position display does not change to reflect the offset until axis motion on the offset axis occurs	the control does not generate the move activating the offset until a move on the axis is made. Then the offset amount is applied to the endpoint of the move. Each axis that has a length offset must be moved to activate the offset on that axis.

Important: If your system installer selects delay move type as the activation type, the control cannot activate wheel offsets in a circular block (G02 or G03). In this case, the control generates an error if you program a circular move in the same block, or as the first motion block immediately following the T word.

If the activation type is immediate shift /immediate move, the control generates its own linear blocks to activate the offset. In this case, you can program the T word in a circular or linear block. The control activates the offset before the moves in that block take place.

13.2 Programming Changes to Wheel Geometry and Radius Offset Tables (G10L10)

You can enter data in the wheel geometry table and radius/orientation offset table through programming. This section describes the use of the G10 command for loading these offset tables.

Important: Only the value in the table changes when a G10 modifies a table value. If the value just changed with a G10 is currently being used by the control, the value of the active offset does not changed until it is called again from the offset table using a T word.

When the control is in:	any values entered in an offset table using the G10 command are:
incremental mode (G91)	added to the currently existing offset values.
absolute mode (G90)	replace the currently existing offset values.

Use this format to modify the offset tables with a G10:

```
G10 L10 P__ X__ Z__ R__ Q__ T__ O__
```

Where :	Is :
L10	determines the table being modified. L10-Modifies the wheel geometry and radius offset tables
P	the offset number that is having its values changed is specified with the P word. The P word must be a 2-digit number specifying the radius/orientation offset number and/or the wheel geometry offset number being modified.
X*	the value to add to or replace the wheel length offset for the X-axis. This value can be a diameter or radius value if using a cylindrical grinder as determined with the O word. This value is entered into the wheel geometry offset table.
Z*	the value to add to or replace the tool length offset for the Z-axis. This value is entered into the wheel geometry offset table.
R	the value to add to or replace the dresser/wheel radius compensation amount. This value is entered into the radius offset table.
Q	the value to add to or replace the dresser/wheel radius compensation orientation value. This value is entered into the radius offset table.
O	determines if the value being entered into the offset table is a radius or diameter value. This applies only when setting data for the control's diameter axis (typically the axis perpendicular to the spindle) on a cylindrical grinder. If no O word is programmed, the control uses the current radius/diameter mode active on the control. O1-indicates a radius value O2-indicates a diameter value Surface grinder applications should not program an O word.

* indicates the axis name configured on your machine. This manual assumes a (ZX) 2 axis machine configuration.

Any values not specified in the G10 block remain in the table unchanged.

Example 13.1
Using G10 to Change Offset Table Values

`G90;`	Selects absolute mode causes values in G10L10 block to replace existing table values.
`G10 L10 P01 Z2.1 X3.0 R.3 Q1 O1;`	Wheel geometry offset number 1 has a new length value of 2.1 for Z-axis, 3.0 for X-axis. Radius offset number 1 has a new radius value of .3 and new orientation value of 1.
`G10 L10 P02 Z4.0 X2.1 O1;`	Wheel geometry offset number 2 has a new value of 4.0 for Z, 2.1 for X.
`G91;`	Selects incremental mode causes values in G10L10 block to add to existing table values.
`G10 L10 P01 Z-1 R-.5;`	Wheel geometry offset number 1 has a new length value of 1.1 (-1 added to previous value set above). Radius/orientation offset number 1 has a new radius value of 2.5 (-.5 added to previous value set above). The current value for any axis not specified and for the dresser/wheel radius remains unchanged.

Important: You cannot program G10 blocks when dresser/wheel radius compensation is active.

END OF CHAPTER

Chapter 14

Angled-Wheel Grinding

14.0 Chapter Overview

This chapter covers angled wheel grinder applications. The following topics on angled-wheel grinding are covered in this chapter:

- Determining the wheel-angle on an angled-wheel grinder
- Selecting an Angled-Wheel mode (G16.3, G16.4 or G15)
- Position Displays on an Angled-Wheel Grinder
- Manual Motion on an Angled-Wheel Grinder
- Homing an Angled-Wheel Grinder
- Plane Selection on an Angled-Wheel Grinder (G17, G18, or G19)
- Offset on an Angled-Wheel Grinder
- Overtravels and Programmable Zones on an Angled-Wheel Grinder

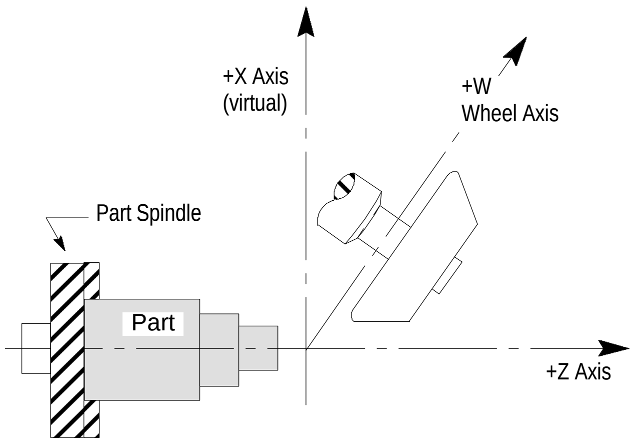
Important: The Angled-Wheel Grinding feature is not available on 9/240 Grinder packages.

14.1 Angled-Wheel Grinder Configuration Assumptions

This manual uses the following configuration for examples and discussions throughout this manual:

- You have a cylindrical grinder with one linear non-orthogonal axis (all axes are not perpendicular)
- The rear non-orthogonal axis is named W and is the wheel axis
- The rear axis (axial) parallel to the part spindle center line is named Z
- There is a virtual axis named X perpendicular to the part spindle center line (there is no actual mechanical hardware for a virtual axis, it is created by the control).
- This manual assumes G16.3 is the default mode (configured in AMP)

Figure 14.1
Angled-Wheel Grinder typical Axis Configuration



Angled-wheel grinders have the same integrand letter for the wheel axis (W) and the virtual axis (X). Refer to your system installers documentation to determine the integrand name for the X and W axes.

14.2 Determining the Wheel Angle

It is necessary to tell the control the angle of the wheel axis. This angle is required to perform calculations that determine the wheel axis position (W) based on the programmed control positions for the virtual axis X. There are three methods to specify this angle. They are:

This entry method:	Has this configuration:
Fixed Angle	the system installer has set the angle of the wheel axis in AMP. The angle of the wheel axis is rigid and can not be adjusted by the operator.
Rotary Axis	the angle of the wheel axis is determined by the position of a rotary axis. Typically this rotary axis changes the wheel angle. When angle wheel mode is entered, the position of the rotary axis is recorded and used as the wheel angle. The system installer determines this rotary axis in AMP.
PAL Selected	the system installers PAL program enters the angle of the wheel axis. This method can require you to enter the angle of the wheel axis, or some external input to PAL can determine the angle. Refer to your system installers documentation for details.

Refer to your system installers documentation for details on which of these methods are used on your system.

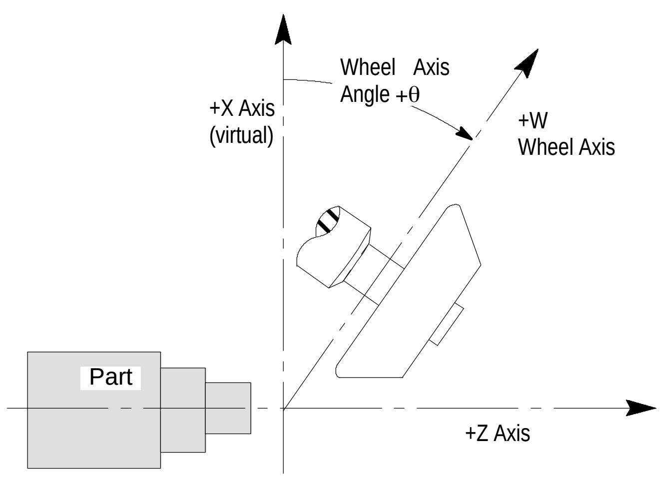
Regardless of the method used, the angle of the wheel axis must remain constant once angle wheel mode is entered (G16.3 or G16.4). You can not change the entered value of the angled wheel axis in angled wheel mode. Any rotary axis or PAL interface that determines the wheel axis angle must be disabled in angled wheel mode (except for homing).

You can home a rotary axis that determines the wheel axis angle while in one of the angled wheel modes. This homing results in angled wheel mode being re-initialized using the angle of the wheel immediatly after it has been homed.

Manually Measuring your Wheel Axis Angle

In some cases the system installer may have written PAL to require that the wheel axis angle be entered into PAL manually. This is usually done through a customized PAL display page on the control or through a switch on the MTB panel. When this is the case you must measure the angle of the wheel axis. Figure 14.2 shows how the wheel axis angle is measured. θ in the figure below is a positive angle.

Figure 14.2
Manually Measuring Wheel Axis Angle



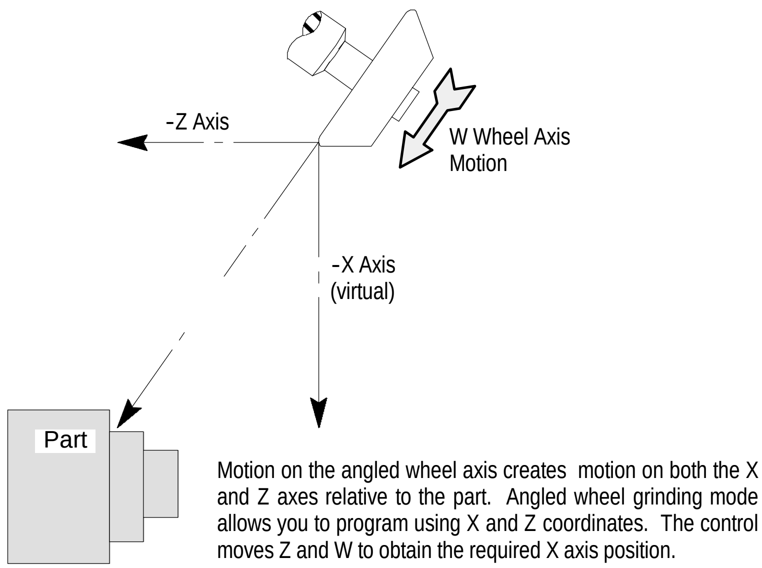
Valid wheel axis angles are ultimately determined by the system installer under the following constraints:

- The wheel axis angle can not be larger then 180°
- Wheel angles that approach 90° are invalid
- Wheel angles are always positive angles
- Angles can be measured to the resolution of the axis if a rotary axis determines the angle
- Angles can be measured to a maximum resolution of two decimal places if the angle is determined through PAL.

14.3 Angled-Wheel Mode (G15, G16.3 and G16.4)

Programming a part contour (or any wheel path) relative to the part on an angled-wheel grinder while not in one of the angled-wheel grinding modes is difficult. Because of the angle of the grinding wheel, the part programmer must consider that any W axis motion generates motion along both the Z and X axes.

Figure 14.3
W Axis Motion generates Wheel Movement on both X and Z Axes



When in one of the angled-wheel grinding modes, the part programmer can program using the X and Z axes. No W axis can be programmed. The control will automatically determine how much actual Z and W motion is necessary to create the programmed X axis move.

The following sections discuss the different modes of angled-wheel grinding available on your 9/SERIES control. There are three angled-wheel modes of operation. They are:

- G16.3 – Angled-wheel normal motion
- G16.4 – Angled-wheel two step motion
- G15 – Angled-Wheel transformation off

These G codes are modal (only one of these G codes can be active at any given time) and belong to modal G code group 15. The system installer determines in AMP which group 15 G code is the default G code (i.e. active at power up or at control reset). This manual assumes G16.3 is the default mode.

The angle of the wheel axis should already have been established before attempting to enter angled wheel mode. You can not change the value of the angled wheel axis in angled wheel mode. Any rotary axis or PAL interface that determines the wheel axis angle must be disabled in angled wheel mode (unless the rotary axis is being homed, see page 14-3).

The following features should be considered when in one of the angled-wheel grinding modes:

Probing

If a probing operation is performed while in one of the angled-wheel grinding modes (G16.3 or G16.4), only the real axes coordinates are saved. Virtual X axis coordinates are not recorded and are not used for any probing calculations.

CSS (Constant Surface Speed)

The virtual X axis is used for CSS when in the angled-wheel mode and the W axis has been selected in AMP as the diameter axis. The system installer could also allow the selection of S (wheel spindle) as the CSS axis. In this case the wheel size as specified on the in process dresser screen is used for CSS calculations.

If not in angled-wheel mode, CSS is still valid however the W axis is used to calculate CSS. This will cause some inaccuracy in part spindle speed because the angle of the W axis is not used in CSS calculations. CSS is not recommended in G15 mode because of the angle of the W axis.

In Process Dresser

This feature is not available on an angled-wheel grinder. The in process dresser screens and softkeys however are available to allow the entry of CSS information.

Mirror Imaging

Mirroring can be performed as follows:

In this mode:	These axes can be mirrored:
G16.3	X and Z (can not mirror W)
G16.4	X and Z (can not mirror W)
G15	Z and W (can not mirror X)

Refer to your system installers documentation for details on how axis mirroring is performed.

Radius and Diameter Modes

Both radius and diameter modes (G07 and G08) are available on an angled-wheel grinder. The control will not allow the X and W axes to be in different radius/diameter modes. Both the X and W axes are forced by the control to be in the same mode (G07 or G08).

Programming Restrictions

With the exception of G86, G86.1, G87, G87.1, G88, G88.1, G89 and G89.1, the following operations should be performed only in G16.3 mode:

- circular interpolation
- cylindrical grinding cycles
- threading operations
- turning cycles
- reciprocation operations
- wheel radius compensation
- ,C or ,R chamfer or radius programming

These features are available in G15 mode, however it is not recommended because of the increased complexity of programming. These features are not available in G16.4 mode. The G89 and G89.1 cycles where designed to operate in both the G16.3 and G16.4 modes.

QuickView

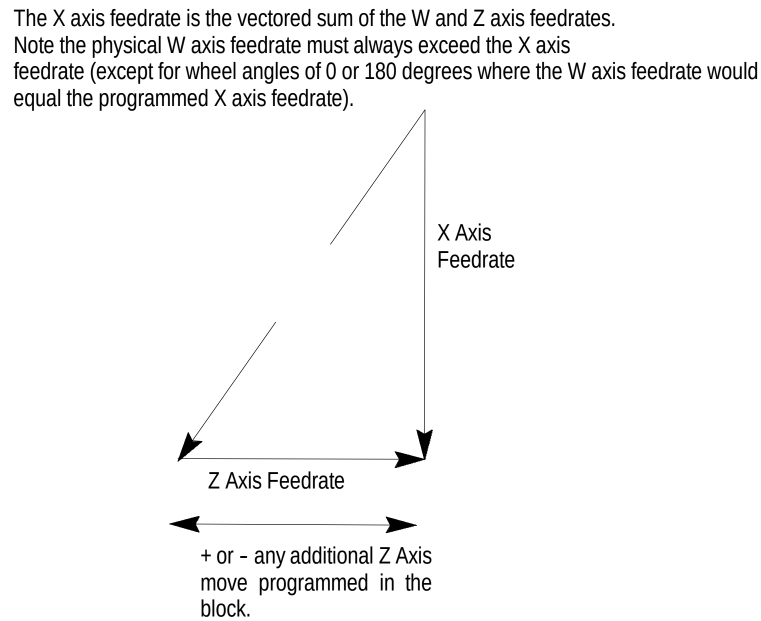
On angled-wheel grinders an additional plane select softkey is available that allows you to select the angled-wheel plane (XZ). When this softkey is pressed, all plane axis references on the QuickView screens change to the angled-wheel plane. Refer to page 5-27 for details on selecting a QuickView plane.

14.3.1 Normal Angled-Wheel Grinding Mode (G16.3)

Normal angled-wheel grinding mode (G16.3) forces the control to position the X and Z axes simultaneously. The control will calculate how much W axis motion must occur to reach the programmed endpoint. Z and W axis moves are interpolated together to reach the desired endpoint.

The feedrate programmed for the block is the actual feedrate of the wheel relative to the part (times any feedrate override). The control calculates the necessary W and Z axis speeds required to generate a combined feed equal to the programmed feedrate on X. The control will clamp the axes speeds when the move would exceed the allowable rapid traverse for any axis. The feedrate at which the axes clamp is dependant on the wheel axis angle and the maximum cutting speed configured for the W and Z axes.

Figure 14.4
Feedrate Clamp Reached on W Axis



If one of the axes that make up the X move reaches its maximum rapid feedrate, that axis speed is clamped and the X axis speed is never reached. The speed of the second axis that makes up the X axis move is also adjusted so to reach endpoint simultaneously with the clamped axis. Keep in mind that if a Z axis move is also programmed with an X axis move that feedrate is also added (or subtracted depending on the direction of the move) from the Z axis speed but this sum is not allowed to exceed the axis rapid.

The maximum cutting feedrate for the virtual axis (X) is a function of the angle of the wheel axis, the angle of the move, and the maximum cutting feedrates for the wheel axis (W) and the axial axis (Z). The physical speed of any given axis can exceed the cutting feedrate however, the wheel feed relative to the part will not exceed the cutting feed.

The G16.3 command is modal and belongs to modal group 15. Programming a G16.3 in a block cancels G16.4 mode.

The G16.3 command must be programmed in a block that contains no motion. If any axis words are in a G16.3 block, the control will generate an error.

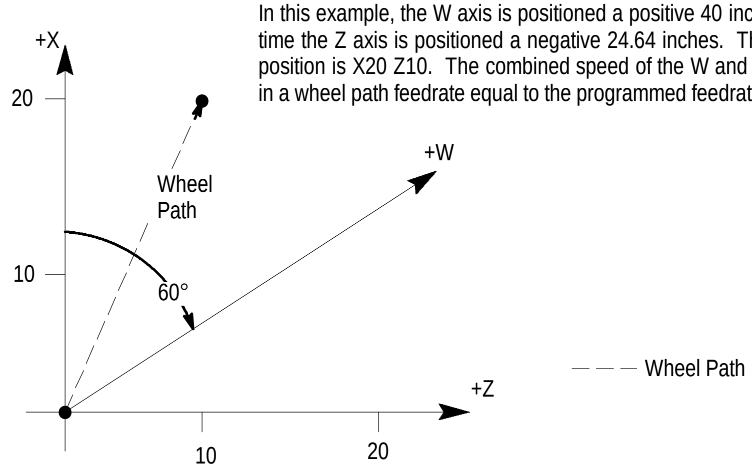
Program using the X and Z axis words in this mode. In the G16.3 mode, the control will no longer allow you to program W axis words. Instead the virtual axis named X is activated. X axis moves are actually a combined move on both the W and Z axes. Both linear and circular interpolation can be programmed for the X and Z axis pair in this mode.

Upon entry into one of the angled wheel modes the control cancels all active offsets. Offsets are not canceled when you change between G16.3 and G16.4 mode as long as angled wheel mode is not canceled with a G15 between modes.

Example 14.1
Linear Interpolation in G16.3 Normal Angled-Wheel Grinding Mode

This example assumes a 60° wheel axis angle.

```
G15;
G08 G90 G00 W0 Z0;
G16.3;
G90G01X20Z10F.1;
```



In this example, the W axis is positioned a positive 40 inches at the same time the Z axis is positioned a negative 24.64 inches. The resulting final position is X20 Z10. The combined speed of the W and Z axis will result in a wheel path feedrate equal to the programmed feedrate of .1.

The W and Z axes must be homed before any programmed motion can occur on the X axis. If a rotary axis is used to determine the angle of the wheel axis, that rotary axis must also be homed before positioning on the X axis can occur.

Canceling G16.3 mode

The G16.3 mode is modal with other modal group 15 G codes. The following G codes cancel the G16.3 mode:

- G16.4 – Angled-Wheel Two step Motion
- G15 – Angled-Wheel Transformation Off

14.3.2 Two Step Angled-Wheel Grinding Mode (G16.4)

Two step angled-wheel grinding mode (G16.4) positions the X and Z axes separately. The control will calculate how much W axis motion must occur to reach the programmed X and Z endpoint. Z and W axis moves are positioned to their respective endpoints in two independent moves.

The Z axis move is performed at the feedrate programmed for that block times any feedrate override. The W axis feedrate is modified so that the feedrate of the X axis motion generated by the W move (feedrate of the wheel towards or away from the parts centerline) is equal to the programmed feedrate times any feedrate override. This means that feedrate of the W axis is dependant on the wheel axis angle. The feedrate of the W axis move will always be greater than or equal to the programmed feedrate (not taking into account any feedrate override that is currently active).

The system installer determines in PAL which of the two step moves take place first. The system default is as follows:

	X axis direction is programmed away from the part spindle centerline:	X axis direction is programmed towards the part spindle centerline:
STEP 1	all W (wheel axis) motion takes place	all axis motion generated from the block takes place except W axis motion (Z axis and any other axes programmed)
STEP 2	all other axis motion generated from the block takes place (Z axis and any other axes programmed)	all W (wheel axis) motion takes place

The G16.4 command is modal and belongs to modal group 15. Programming a G16.4 in a block cancels G16.3 mode.

The G16.4 command must be programmed in a block that contains no motion. If any axis words are in a G16.4 block, the control will generate an error.

Program using the X and Z axis words in this mode. Once the G16.4 mode is entered, the control will no longer allow the programming of W axis words. Instead the virtual axis named X is activated. X axis moves are actually a combined move on both the W and Z axes.

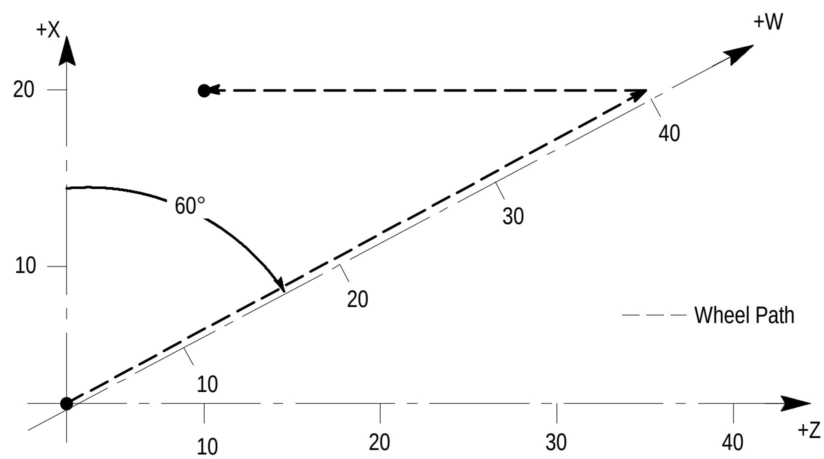
Upon entry into one of the angled wheel modes the control cancels all active offsets. Offsets are not canceled when you change between G16.3 and G16.4 mode as long as angled wheel mode is not canceled with a G15 between modes.

Example 14.2
Linear Interpolation in G16.4 Two Step Angled-Wheel Grinding Mode
(motion is away from part)

This example assumes a 60° wheel axis angle.

```
G15;
G08 G90 G00 W0 Z0;
G16.4;
G90G01X20Z10F.1;
```

In this example, the W axis is positioned a positive 40 inches. Once the W axis reaches position, step 2 begins which positions the Z axis a negative 24.64 inches. The resulting final position is X20 Z10.



The following features can not be used while the control is in the G16.4 mode:

- Wheel/Dresser radius compensation (G41/G42)
- Circular Interpolation (G02/G03)
- Reciprocation and cycles that require reciprocation or dither (G81, G81.1, G82, G82.1, G83, G83.1, G84, G84.1, G85, G85.1)
- Turning cycles (G20, G24)
- Threading blocks (G33, G34)
- Chamfer and Corner Radius blocks (,C and ,R)

When executing blocks in single block mode, the control will stop between steps one and two.

The W axis must be homed before any programmed motion can occur on the X axis. If a rotary axis is used to determine the angle of the wheel axis, that rotary axis must also be homed before positioning on the X axis can occur.

Acceleration/Deceleration Considerations

While in the G16.4 two step angled wheel mode, there is a complete deceleration of the axis to zero at the end of each step. When in positioning mode (G00) or in exact stop mode (G61), the control will perform an in-position check after each step before continuing execution.

Canceling G16.4 mode

The G16.4 mode is modal with other group 15 G codes. The following G codes cancel the G16.4 mode:

- G16.3 – Normal Angled-Wheel Grinding Mode
- G15 – Angled-Wheel Transform Off

14.3.3 Angled-Wheel Transform Off (G15)

Programming a G15 cancels both angled-wheel grinding modes. The G15 command must be programmed in a non-motion block (no axis commands allowed in the block). In the G15 mode the virtual X axis can not be programmed. Any part program blocks that contain an X axis command will generate an error when executed in G15 mode.

Programming a part contour (or any wheel path) relative to the part on an angled-wheel grinder while not in one of the angled-wheel grinding modes is difficult. Because of the angle of the grinding wheel, the part programmer must consider that any W axis motion generates motion along both the Z and X axes. In G15 mode W axis moves must be programmed directly.

14.4 Position Displays for Angled-Wheel Grinders

This section covers how axis position registers are presented on the operator panel. Some screens will show a combination of the following axis position registers:

- Z (real) – This is the physical position of the grinding wheel along the Z axis slide relative to a fixed position. This does not change with W axis motion.
- Z (virtual) – This is the position of the grinding wheel along the Z axis relative to a fixed position on the machine which is not on the Z axis slide. This position is changed by W axis motion. Note that in the G16.4 mode the Z (real) position will not change during the W move because the Z axis attains its real endpoint in a separate move. The Z (virtual) position changes during both steps one and step two in G16.4 mode.
- W (real) – This is the physical position of the grinding wheel along the W axis slide relative to a fixed position. There is no W (virtual) display.
- X (virtual) – this is a virtual diameter axis. It is orthogonal to the Z axis of the machine. There is no physical hardware that makes up an X axis so there is no X (real) position display.
- Other Axes (real) – All other axes selected to appear on position displays will be the actual physical position along their respective axis slide.

The Z and W axis display remains on all screens regardless of whether in angled-wheel mode or not. The X axis position display only appears when in one of the angled wheel modes. The position displays when in angled wheel mode are as shown in the following table:

This Position Display:	Absolute	Programmed	Target	DTG	Following Error
Shows these Axis Positions:	Z (real)	Z (virtual)	Z (virtual)	Z (virtual)	Z (real)
	X (virtual)	X (virtual)	X (virtual)	X (virtual)	-
	W (real)	W (real)	W (real)	W (real)	W (real)
	Other Axes	Other Axes	Other Axes	Other Axes	Other Axes

Note: The word (real) and (virtual) do not physically appear with the axis position displays. They are shown here for clarity only.

When not in angled-wheel mode (G15) all position displays show (real) values. There is no X (virtual) or Z (virtual) in G15 mode.

The following table shows the position displays as a program executes under the following conditions:

- in G16.3 normal angled wheel mode
- in single block mode.
  wheel axis angle of 60 degrees

Program Block:	Program Display:			Absolute Display:		
	W	Z	X	W	Z	X
G07G00W0Z0;	0.0	0.0	N/D	0.0	0.0	N/D
G01W10Z1F100;	10.0	1.0	N/D	10.0	1.0	N/D
G16.3;	10.0	9.66025	5.0	10.0	1.0	5.0
X6;	12.0	9.66025	6.0	12.0	-.73205	6.0
X4;	8.0	9.66025	4.0	8.0	2.73205	4.0
G15;	8.0	2.73205	N/D	8.0	2.73205	N/D

N/D - no display is shown for this axis.

The following table shows the position displays as a program executes under the following conditions:

- in G16.4 two step angled wheel mode
- in single block mode.
- wheel axis angle of 60 degrees
- PAL is configured to perform W moves first when moving away from spindle centerline

Program Block:	Program Display:			Absolute Display:		
	W	Z	X	W	Z	X
G07G01W10Z1F0;	10.0000	1.0000	N/D	10.0000	1.0000	N/D
G16.4;	10.0000	9.6602	5.0000	10.0000	1.0000	5.0000
X6F100;						
STEP1	12.0000	11.3923	6.0000	12.0000	1.0000	6.0000
STEP2	12.0000	9.6602	6.0000	12.0000	-.7320	6.0000
X4;						
STEP1	12.0000	13.1243	6.0000	12.0000	2.7320	6.0000
STEP2	8.0000	9.6602	6.0000	8.0000	2.7320	4.0000
G15;	8.0000	2.7320	N/D	8.0000	2.7320	N/D

Graphics

Graphics can be configured to plot the X (virtual) as well as the W (wheel axis) in any angled-wheel grinding mode (G15, G16.3 or G16.4). No motion will appear on the graph for the X axis in the G15 mode. Plots of the wheel axis are orthogonal regardless of your current wheel axis angle. This can cause graphics to show an part contour not representative of the actual wheel path relative to the part.

14.5 Manual Motion on an Angled-Wheel Grinder

This section covers features or considerations that must be taken into account when jogging an angled-wheel grinder. For details on using the jogging features refer to the manual motion sections starting on page 4-1. All manual motions use normal angled-wheel positioning method (G16.3) regardless of the current angled-wheel mode. Two step positioning is not available in manual mode. G16.4 mode uses the G16.3 type of positioning for manual motions.

Important: The PAL axis mover feature is not available to move the X virtual axis in any grinding mode. Refer to your system installers documentation for details on features that use the PAL axis mover on your control.

Normal single axis jogs

In angled-wheel mode (G16.3 or G16.4 active) the operator can jog either the real W or Z axes, or the virtual X axis. Restrictions apply to jogging multiple axes (see below).

When not in angled-wheel mode, the virtual X axis does not exist. This means manual X axis moves can not be performed while in G15 mode.

If a rotary axis is used to specify the wheel axis angle, you can not jog this rotary axis while in angled-wheel mode. You can however home the rotary axis while in angled-wheel mode.

Multiple axis jogs including Arbitrary Angle jogs

In angled-wheel mode (G16.3 or G16.4 active) jogs that request multiple axes are not permitted. You can only request a jog on X, Z, or W one axis at a time. Requests to perform the following multiple axis jogs are not permitted in angled-wheel mode:

- X and Z together
- X and W together
- W and Z together

X axis jogs are not considered a multiple axis jog even though X axis moves position W and Z together.

Note that X axis jogs will position both Z and W axes simultaneous in G16.3 mode and G16.4 mode.

Multiple axis jogs or arbitrary angle jogs are permitted with axes other than X, Z, and W. For example a UZ or UX jog would be possible assuming U was an axis not in the angled-wheel plane and not the angled-wheel rotary axis.

While not in angled-wheel mode (G15 active) the W and Z multiple axis or arbitrary angle jog combination is permitted. Again as discussed above since the X axis does not exist in G15 mode, no jogs requesting the X axis are permitted.

14.6 Homing an Angled-Wheel Grinder

This section covers features or considerations that must be taken into account when homing an angled-wheel grinder. For details on homing an axis refer to the homing section starting on page 4-11.

Only real axes can be homed. The X axis is a virtual axis on an angled-wheel grinder and can not be homed. Since X axis motion is comprised of a combination of W and Z axis positioning, both the W and Z axes must be homed before the X axis can be programmed.

If the angle of your wheel axis (W) is determined by a rotary axis, the rotary axis must be homed before angled wheel grinding mode (G16.3 or G16.4) is allowed. While in G16.3 or G16.4 mode you must home the rotary axis before the (Z) axial axis or wheel axis (W) is homed.

Automatic homing (G28) and all of the programmable to/from homing operations (G27, G28, G29, and G30) are available when in angled wheel mode. Since however the W axis can not be programmed in angled-wheel grinding mode, the W axis can not be automatically homed nor can any of the return to/from home operations be performed on the W axis.

14.7 Plane Selection on Angled-Wheel Grinders

Plane selection (G17, G18, or G19) is not allowed in either angled-wheel grinding mode (G16.3 or G16.4). When angled-wheel grinding mode is entered the control automatically selects the ZX plane (assuming the Z axis has been configured as the axis parallel to the part spindle center line and the X axis is the virtual axis perpendicular to Z).

Other plane dependant features can not be performed on other axes while in angled-wheel mode. For example you could not change planes to perform circular interpolation on a ZU axis while in angled-wheel mode.

Plane select G codes programmed in angled-wheel grinding mode generate an error. When angled-wheel mode is exited the modal plane (G17, G18, or G19) that was active before angled-wheel mode was entered is re-established.

When angled-wheel mode is exited either:

- the plane that was active prior to entering angled-wheel mode is re-established

  or

- if a plane select G code is programmed in the G15 block, that plane becomes active.

14.8 Offsets on an Angled-Wheel Grinder

Read this section if you are using wheel length and radius offsets, and work coordinate system offsets on an angled-wheel grinder.

Upon entry into one of the angled wheel modes the control cancels all active offsets. Offsets are not canceled when you change between G16.3 and G16.4 mode as long as angled wheel mode is not exited with a G15.

When using the automatic offset table backup features the control inserts a G16.3 block as the first block in the G10 program. This block is always inserted when an offset table is backed up on an angled wheel grinder regardless of the current angled wheel mode. This allows the control to restore X axis offsets when the backup program is executed later. This can change the angled-wheel mode of your machine if the control is not configured to reset to default G codes when an M30 end of program is executed or if the default G codes is not the mode you were in when you began executing the G10 program.

Offset table values can be manually changed (using the softkeys) for all axes in and out of angled wheel mode (G15, G16.3, and G16.4). When programming a G10 block in your own part program:

- in G15 mode any virtual axis (X) values programmed are not entered into the offset table.

- in angled wheel mode (G16.3 and G16.4) all axis values (including the virtual axis X and wheel axis W) programmed in the G10 block are entered into the offset table.

Wheel Length Offsets

When wheel length offsets are entered into the offset table both the X(virtual) and W(real) axes allow entry. When a wheel length offset is activated the control selects the offset value out of the offset table as follows:

in Angled-Wheel Mode: (G16.3 or G16.4)	X and Z offsets are used
not in Angled-Wheel Mode: (G15)	W and Z offsets are used

If a wheel length offset is activated manually (through the use of the {ACTIVE OFFSET} softkey), the wheel length offset will be activated in the normal angled wheel mode (G16.3).

Important: If you are using either the G89 or G89.1 cycles the active tool offset when a micro-feed is aborted has special constraints. Refer to page 17-40 for details.

Work Coordinate System Offsets

When work coordinate system offsets are entered into the offset table both the X(virtual) and W(real) axes allow entry. When the control activates a work coordinate system, it selects the coordinate offset values out of the offset tables as follows:

In Angled-Wheel Mode: (G16.3 or G16.4)	X and Z offsets are used
Not in Angled-Wheel Mode: (G15)	W and Z offsets are used

14.9 Overtravels and Programmable Zones on an Angled-Wheel Grinder

This section covers features or considerations that must be taken into account when using software overtravels, and programmable zones with an angled-wheel grinder.

Software Overtravels

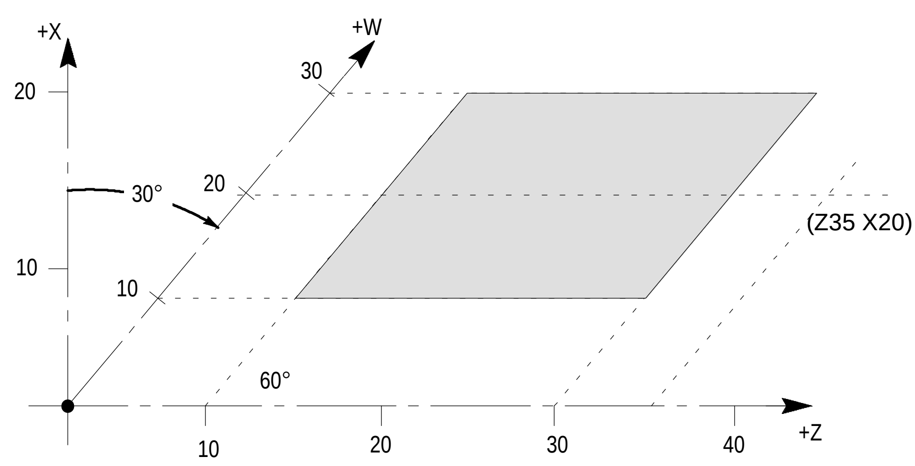
For details on what software overtravels are and how they work refer to page 11-34. The system installer determines the location and operation of the software overtravels. Since the X axis is a virtual axis, no software overtravels are available to monitor the X axis. Software overtravel are available that monitor the W and Z axis positions.

Programmable Zones

For details on what programmable zones are and how they work refer to page 11-34. Programmable zones can be configured by the system installer, programmer, or operator. The programmable zones you set up on an angled wheel grinder are significantly different zones when in angled wheel mode versus non-angled wheel mode. The virtual axis (X) is used to perform zone checks when in angled wheel mode. The wheel axis (W) is checked when not in angled wheel mode. Zones are always checked as absolute values on their given axes.

Zone values entered either through AMP, the programmable zone table discussed in chapter 3, or through programming (for zone 3 only), are always entered for the W axis when in non-angled wheel mode (G15 active). These W values are used at their entered value for the W wheel axis in non-angled wheel mode. Figure 14.5 illustrates the zone in non-angled wheel mode. Note that the Z axis values in the figure are constant and independent of any Z motion created in physical space by the W axis.

Figure 14.5
Programmable Zone on an Angled-Wheel Grinder
Not in Angled-Wheel Mode



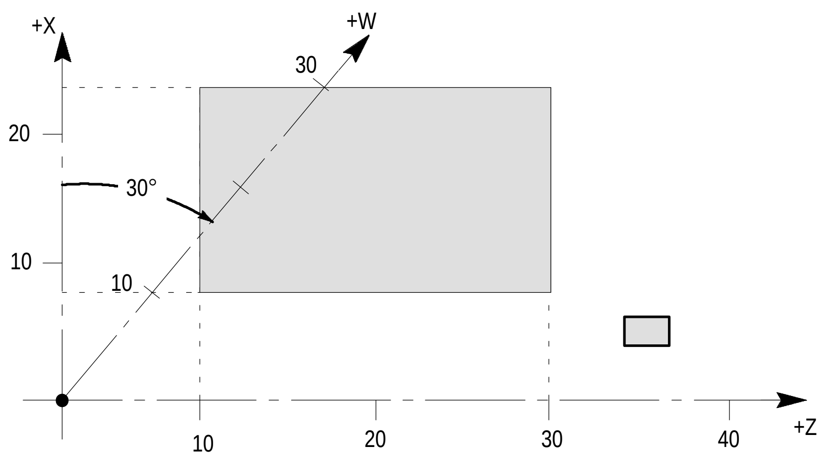
The minimum and maximum Z and W axis zone values in this picture are both configured at 10, 30. Though it appears that in real space if Z is at 35 and W is at 20 the wheel would be inside of this zone, that is not the case. All Z axis values must be calculated parrallel to the W axis to negate any Z component created by the W axis position.

When you make the transition into angled-wheel mode, zone values entered for the W axis are transformed over to the X axis based on your current wheel axis angle. The equation used to transform values from the W to the X axis is as follows:

```
X zone value = (COS A)(W zone value)
```

Where:
A = wheel axis angle
W = min or max W axis zone value

Figure 14.6
Programmable Zone on an Angled-Wheel Grinder
in Angled-Wheel Mode



While in angled wheel mode if you decide to change any of the zone values, either through the programmable zone tables, or through programming (for zone 3 only), you assign values to the X axis. These values are used as you entered them for the X axis and no calculation is made. When you make the transition out of angled wheel mode (G15), the zone values entered for the X axis are transformed over to the W axis based on your current wheel axis angle. The equation used to transform values from the X to the W axis is as follows:

$$\frac{\text{X zone value}}{(\text{COS A})} = \text{W zone value}$$

Where:
A = wheel axis angle
W = min or max W axis zone value

While in angled wheel mode you can not enter W axis zone values. Also X axis values can not be entered in non-angled wheel mode (G15).

If you last entered W axis values, those values are transformed over to the X axis for angled wheel mode. If you last entered X axis values those values are transformed over to the W axis for non-angled wheel mode.

ATTENTION: Changing the wheel axis angle results in a change to the programmable zone area. As the wheel axis approaches 90 degrees, the protected area of the programmable zone approaches zero. Areas once protected by the programmable zone can become unprotected when the wheel angle changes. Additionally areas once accessible can become unaccessible as the wheel axis angle changes. You must recalculate and reenter your programmable zones limits any time the wheel axis angle is changed.

END OF CHAPTER

Chapter 15

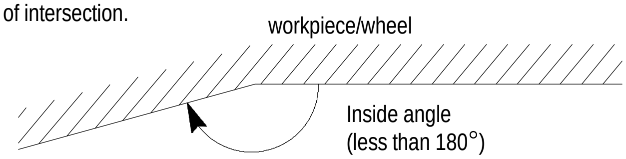
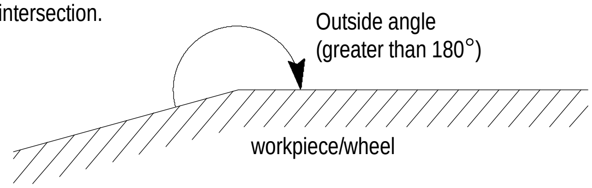
Dresser/Wheel Radius Compensation

15.0 Chapter Overview

This chapter contains this information:

Terms Used

We use the following terms in this chapter:

If you see:	It means:
inside	an angle between two intersecting programmed paths is referred to as inside if, in the direction of travel, the angle measured clockwise from the second 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 path at their point of intersection. 
outside	an angle between two intersecting programmed paths is referred to as outside if, in the direction of travel, the angle measured clockwise from the second 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 path(s) at their point of intersection. 
r	dresser or wheel radius
CR	cross-point between two programmed paths after the dresser/wheel radius compensation is activated

15.1 Introduction to Dresser/Wheel Radius Compensation

When programming a CNC, the part program is actually the reference position of a control point on the grinding wheel relative to the active coordinate system. Your system installer establishes this control point by establishing a machine gauge point combined with your active wheel length offsets.

Generally diamond-point dressers and the corners of the grinding wheel have rounded edges that often result in a difference between the actual machining edge and the control point. This inevitably causes a difference between the programmed contour and the actual final contour of the part or dressed wheel (see Figure 15.1).

Use the dresser/wheel radius compensation feature to compensate for this difference. This feature allows you to use a wheel and diamond point dresser with different radii without modifying the part program.

Dresser/wheel radius compensation also uses dresser/wheel orientation data. You need orientation data:

- to compensate for inaccuracies that can occur from difficulties in measuring wheel corner and dresser radius because of mounting position

  and

- to tell the control which edge of the dresser/wheel you intend to use when dressing or grinding

If the above is not a factor, make sure all orientations use an orientation of 0 or 9.

Enter the radius and orientation data into the offset tables before attempting to activate any compensation. See chapter 3 for details on entering offset table data.

Types of Dresser/Wheel Radius Compensations

Two types of dresser/wheel radius compensation are available on the control:

- type A (as described on page 15-17)
- type B (as described on page 15-27)

The following table highlights the differences between the two types:

Type of Move	Type A	Type B
Entry Move Into compensation	The dresser/wheel takes the shortest possible path to its offset position.	The dresser/wheel 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 can occur in Type A.
Dresser/wheel Path	Same as Type B.	Same as Type A.
Exit Move From compensation	The dresser/wheel takes the shortest path to the end-point of the exit move for both inside and outside corners.	The dresser/wheel takes the shortest path to the end-point of exit move for inside corners only. For outside corners, the dresser/wheel stays at least one radius away from the end-point.

Your system installer determines in AMP whether your control uses type A or type B compensation.

Figure 15.1
Grinding Wheel Radius Compensation Taper and Arc Cutting

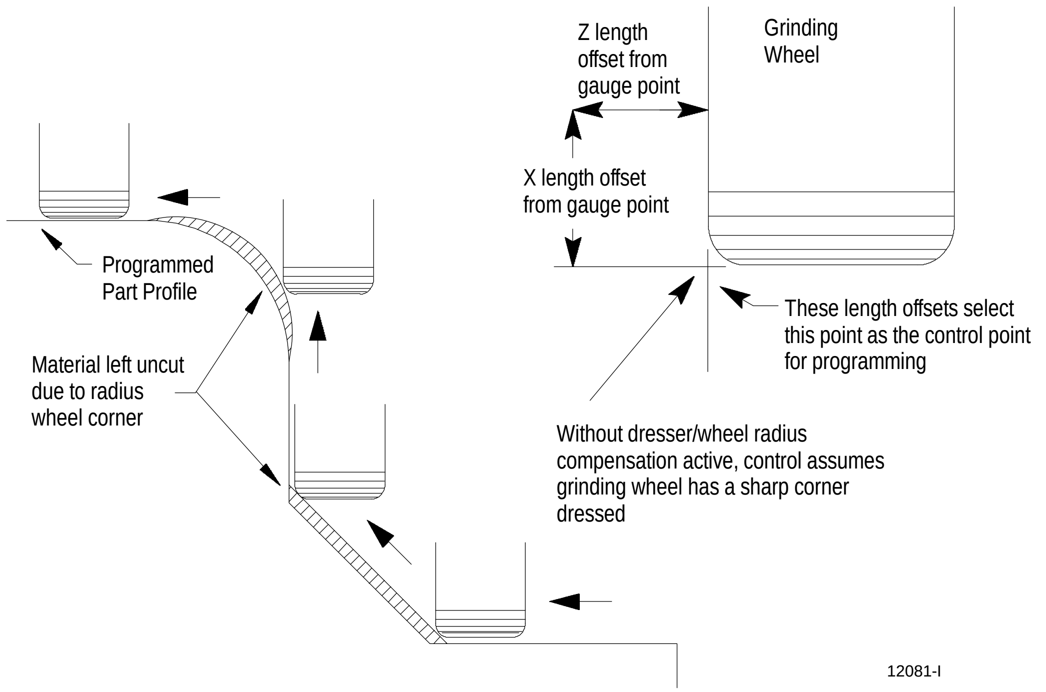
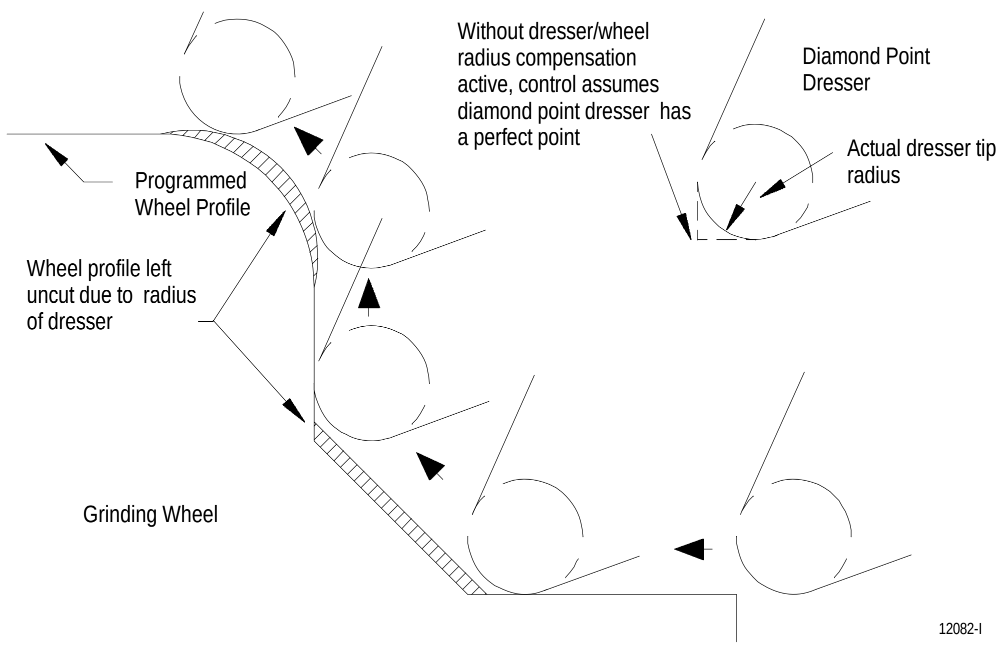


Figure 15.2
Diamond Dresser Radius Compensation Taper and Arc Cutting



15.2 Programming Compensation (G40, G41, G42)

Use the G-codes in Table 15.A for dresser/wheel radius compensation:

Table 15.A
G Code Compensation Direction

G Code [1]:	Dresser/Wheel Radius Compensation:
G40	cancel
G41	left of program path [2]
G42	right of program path [2]

[1] All of these G-codes are modal and belong to the same modal group

[2] Left or right is defined as offsetting the dresser/wheel to the left or right of the programmed path when facing the direction of axis motion

Important: The dresser/wheel radius compensation function is not available during thread cutting (G33 and G34). You must cancel dresser/wheel radius compensation before you perform threading.

Important: If a negative value is set in the offset tables as the dresser/wheel radius, compensation direction (dresser/wheel left or right) is reversed for the G41 and G42 codes. You can also reverse G41 and G42 during the mirroring operation. See page 12-79 for details on Mirror Image operation.

15.2.1 Application Schemes

The control supports 3 typical dresser/wheel radius compensation application schemes. Other schemes can be supported as variations of these 3 basic applications:

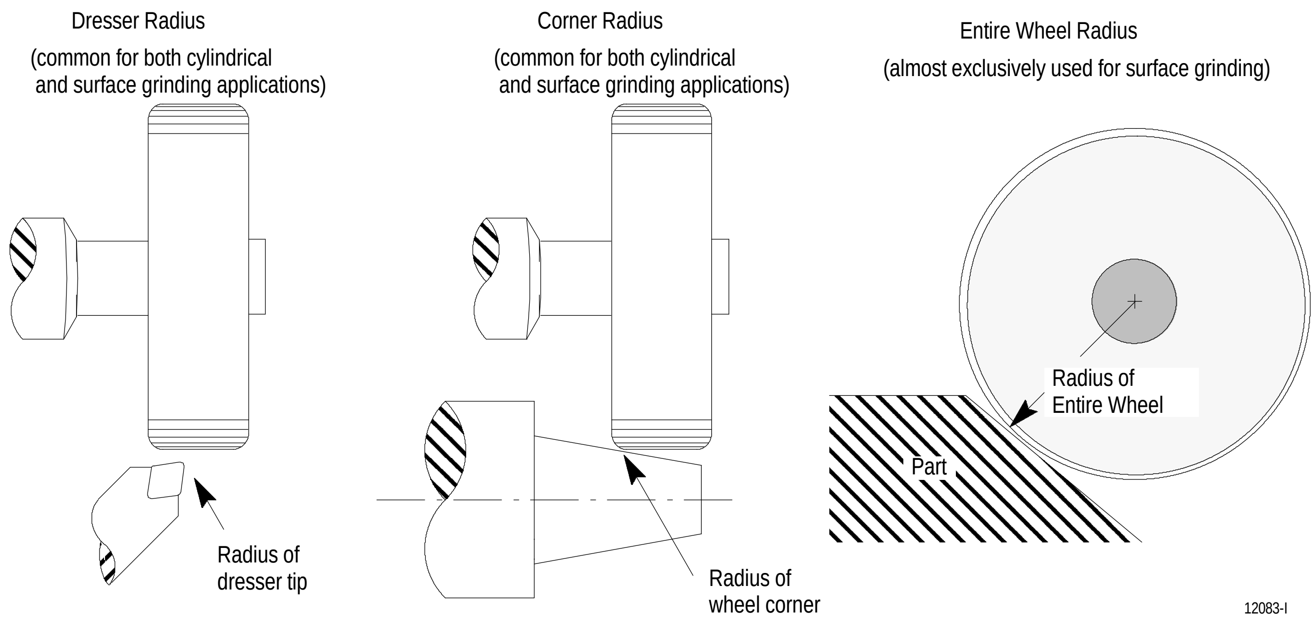
- Dresser radius compensation
- Corner radius compensation
- Entire wheel radius compensation

All of these schemes use the same radius table to store radius values. The dresser/wheel radius compensation scheme used on your system depends on the current application of your grinder.

We describe these 3 compensation schemes below:

Dresser/wheel Radius Compensation Scheme	Length Offsets	Coordinate System Offset (G54-G59.3)	Dresser/wheel Radius Compensation
Dresser Radius	Shifted on Z and X axis to wheel control point	Shifted to point on dresser tip	On radius of diamond dresser
Corner Radius	Shifted on Z and X axis to wheel control point	Shifted to point on part being machined	On radius of wheel corner where Z and X length offset is located
Entire Wheel Radius	Shifted on Z to control point of wheel, Y offset is taken into consideration with dresser/wheel radius compensation	Shifted to point on part being machined	On entire radius of wheel

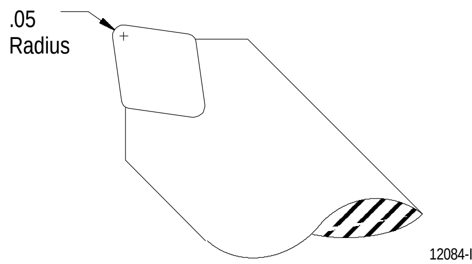
Figure 15.3
Standard Dresser/wheel Radius Compensation Schemes



Dresser Radius

The control can compensate for any dressing error resulting from slight or even large radius of the dresser tip. To do so, you must enter the radius of the dresser in the radius table for radius compensation to properly compensate.

Figure 15.4
Dresser Radius for a Typical Diamond Point Dresser

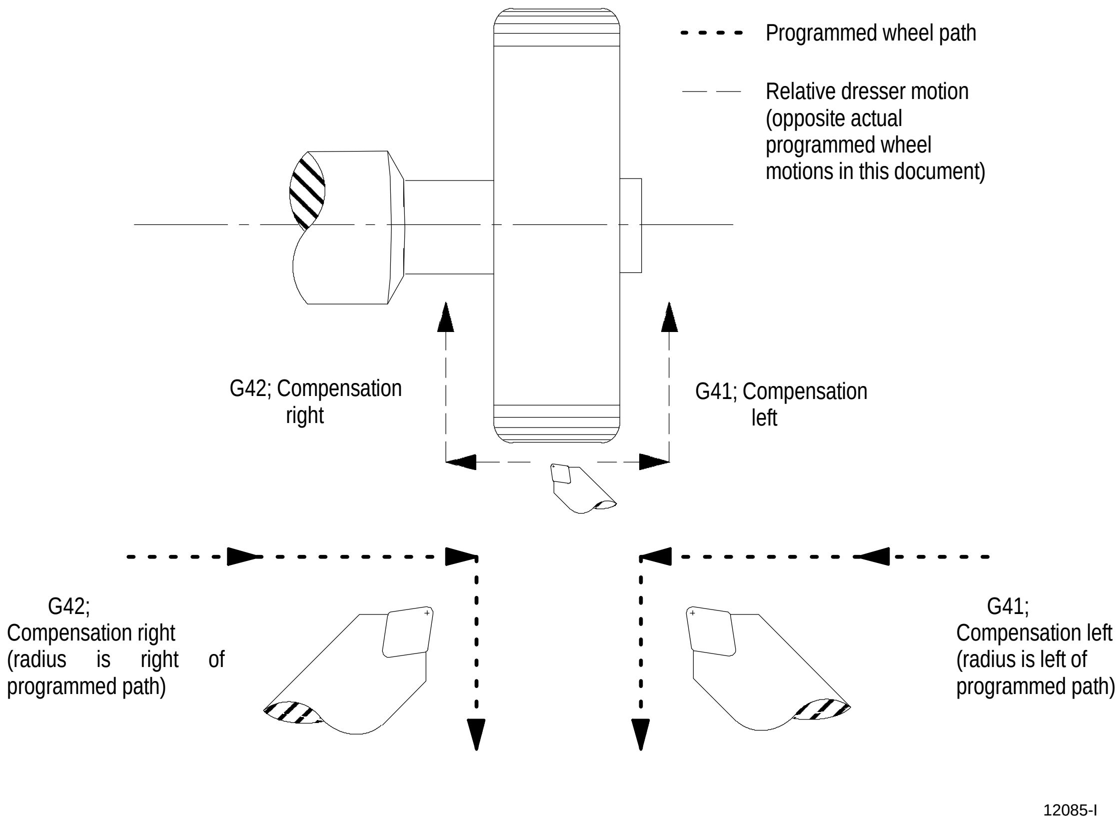


Typically during diamond point dressing, the wheel physically moves across the dresser (i.e. the dresser becomes the shaping tool and the wheel becomes the part). Actual axis motion of the wheel is typically opposite the relative dresser motion. See Table 15.A for compensation direction for G codes.

Figure 15.5 assumes that you configure your grinding wheel to move around a fixed dressing tool. On some machine applications, the dressing tool is mounted on independently moving axes. In this case, see your system installer's documentation for details on compensation direction.

By shifting the currently active coordinate system zero point to the tip of the diamond dresser, programming dressing is simplified. This is because you can program the control point of the wheel about the actual dressing edge of the dresser.

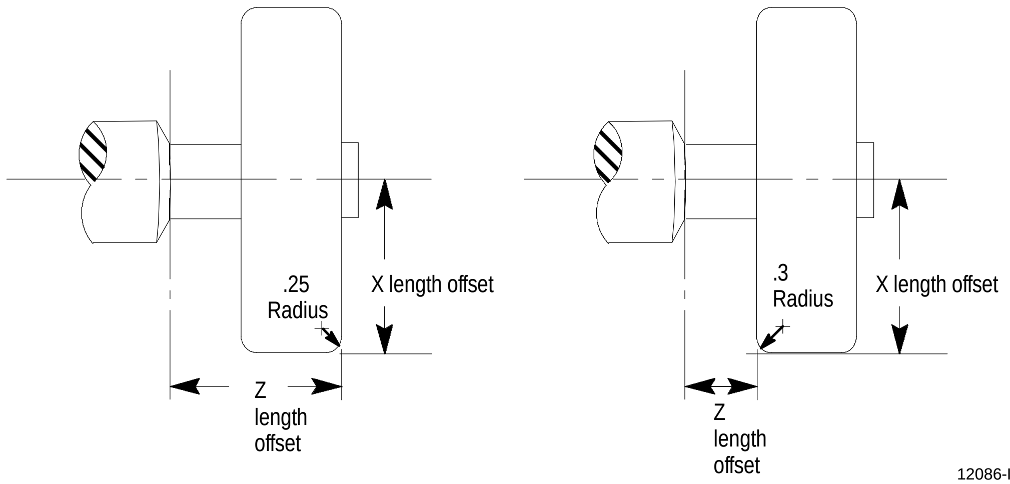
Figure 15.5
Diamond Dresser Relative Motion Across Grinding Wheel to Establish Compensation Direction



Corner Radius

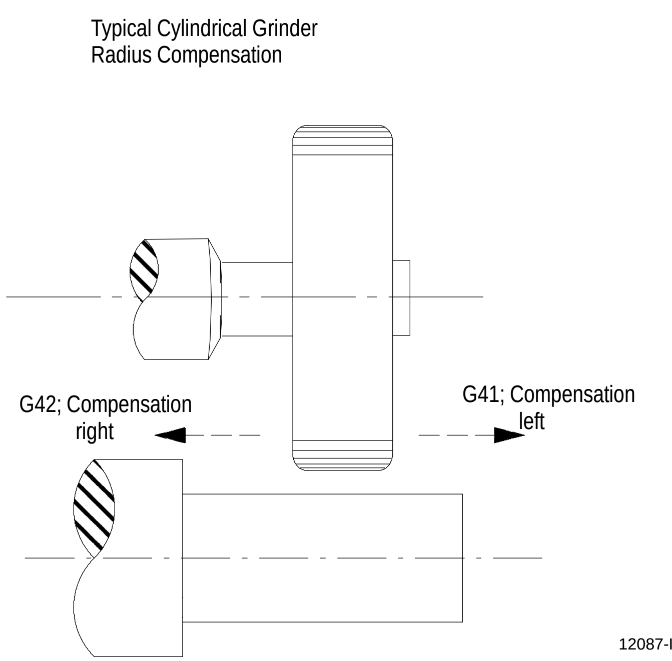
The control can compensate for grinding errors resulting from the rounded corner commonly dressed into the grinding wheel. This corner radius should be the corner of the wheel selected as the control point using wheel length offsets. In order for radius compensation to properly compensate for wheel corner roundness, you must enter the corner radius of the wheel corner currently selected as the control point into the offset table. The same wheel can have one, two, or more corners entered for its corner radius. It is the programmer’s responsibility to use the correct radius (called with a T word) for the correct control point of the grinding wheel (also selected with a T word).

Figure 15.6
Corner Radius for a Typical Grinding Wheel



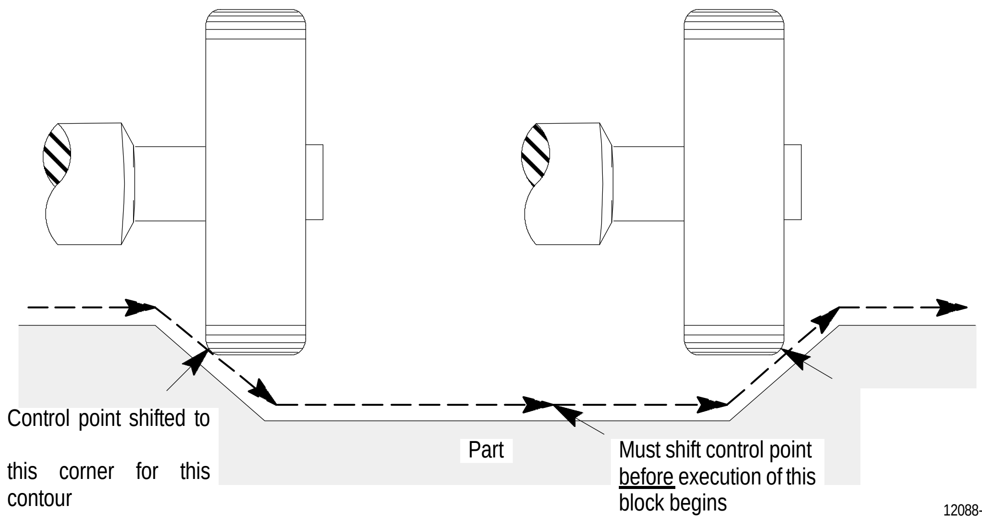
See Table 15.A for compensation direction for G codes.

Figure 15.7
Grinding Wheel Motion Across Part to Establish Compensation Direction



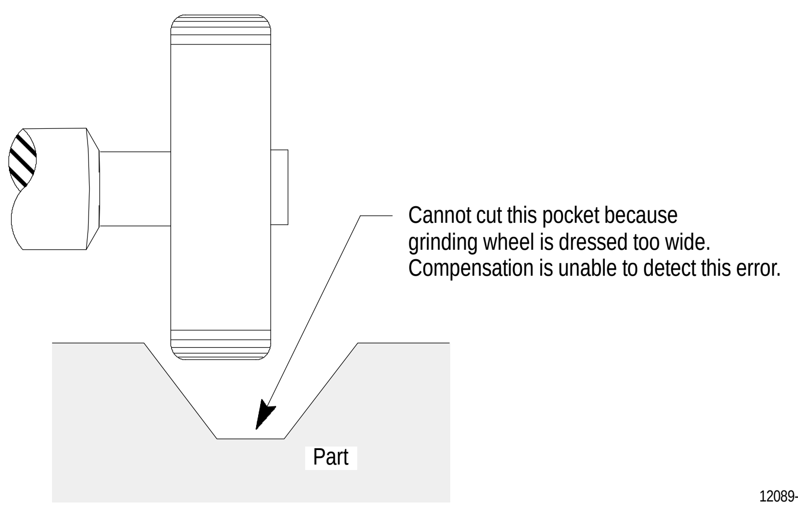
Use care when programming contours using this compensation scheme. You must consider the wheel width when programming and change to the proper control point using the appropriate wheel length offsets as the contour of the part dictates.

Figure 15.8
Programming Considerations For Wheel Width



Additional error conditions can occur from programming without consideration for wheel width. It is the programmer's responsibility to catch these programming errors.

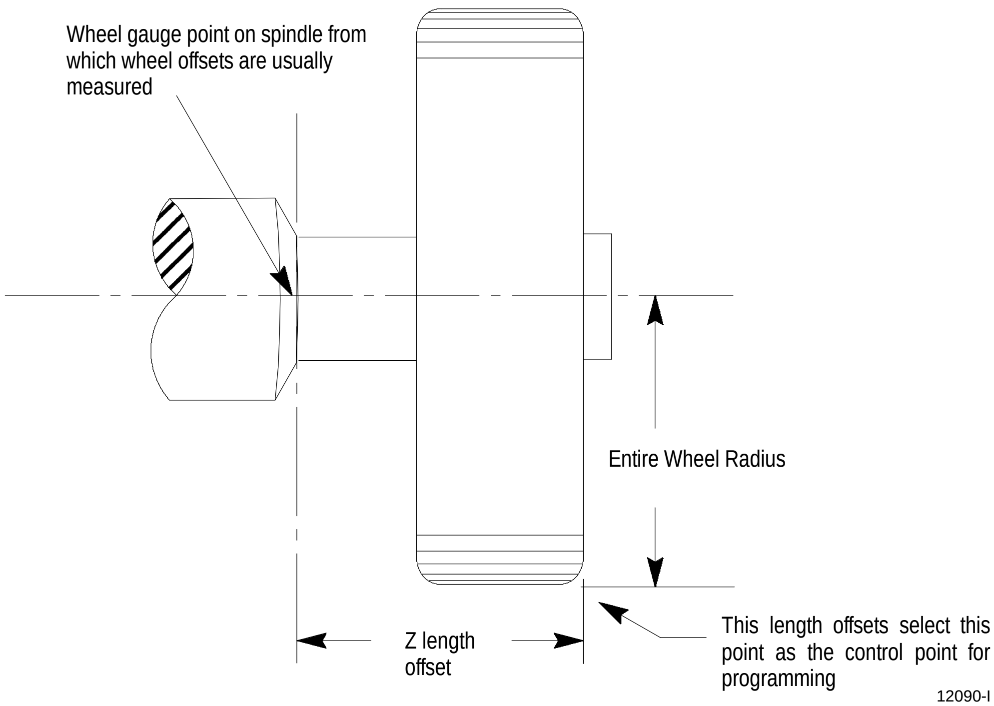
Figure 15.9
Programming Error From Not Considering Wheel Width



Entire Wheel Radius

The control can compensate for any grinding error resulting from the radius of the entire grinding wheel. To do so, you must enter the radius of the wheel in the radius table for radius compensation to properly compensate. This method of compensation does not require the control to activate any X-length offset. Activate only a Z-length offset with the X offset being compensated for by the wheel radius.

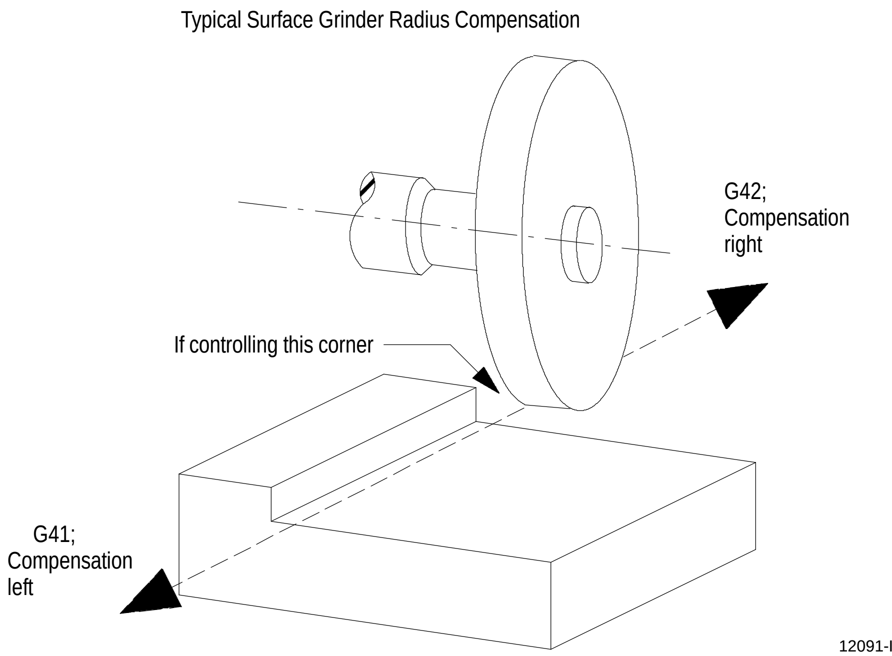
Figure 15.10
Entire Wheel Radius Compensation for a Typical Grinding Wheel



See Table 15.A for compensation direction for G codes.

If using an entire wheel compensation scheme in conjunction with the in-process dresser feature (see chapter 21), you can choose to use a radius offset number 33. This offset number cannot be accessed through the normal radius offset table. Radius offset 33 is created and maintained automatically by the in-process dresser feature and accounts for changes in wheel radius as the wheel is dressed.

Figure 15.11
Grinding Wheel Motion Across Part to Establish Compensation Direction



15.2.2 Compensation Block Format

Program the dresser/wheel radius compensation function with the following format:

```
G41(or G42)X ___ Z ___ T ___ ;
```

Where :	Is :
`G41(or G42)`	dresser/wheel radius compensation direction, G41=left, G42=right
`X, Z`	end-point of entry move into dresser/wheel radius compensation. Program an entry move only on axes in the currently active plane. Axis motion must take place on each axis in the currently active plane in order for dresser/wheel radius compensation to be active.
`T`	used to select the offset numbers and pull data from: • the radius table for the dresser/wheel radius and orientation (first two digits) • the wheel geometry table for the wheel length offsets (second two digits) See page 10-36 for information on programming a T word. The T word is optional in the G41 or G42 blocks. The T word can also be programmed in any previous or following program block.

You can activate dresser/wheel radius compensation in various ways. Example 15.1 illustrates a few examples of activating dresser/wheel radius compensation.

Example 15.1
Initializing Dresser/Wheel Radius Compensation

`Assume: G18 (ZX Plane Selection)`

Program Block	Comment
	One Block
`G42 T1600 X1 Z1;`	Sets compensation right, selects dresser/wheel radius offset number 16, and activates compensation in, move to X1 Z1
	Two Blocks
`T1600;`	Selects dresser/wheel radius number 16
`G42 X1 Z1;`	Sets compensation right and activates compensation in, move to X1 Z1
	Three Blocks
`T1600;`	Selects dresser/wheel radius number 16
`G42;`	Sets compensation right
`X1 Z1;`	Activates compensation in, move to X1 Z1
	Three Blocks
`G42;`	Sets compensation right
`T1600;`	Selects dresser/wheel radius number 16
`X1 Z1;`	Activates compensation in, move to X1 Z1

Important: The T words in the above examples select a length offset number 00. This cancels any active wheel length offsets. In your program, the T word should contain both a radius and length offset number.

Important: Any entry move (see page 15-17 for type A and page 15-27 for type B entry moves) into dresser/wheel radius compensation must be a linear move. You are not allowed to program G41 or G42 commands in a G02 or G03 circular mode in the initial activation of dresser/wheel radius compensation. If, however, dresser/wheel radius compensation is already active, then you can program the G41 or G42 commands in a circular block to change dresser/wheel radius compensation direction, either left (G41) or right (G42).

The T word calls these data from the offset tables:

- Initial dresser/wheel radius data (from radius table)
- Dresser/wheel orientation data (from radius table)
- Wheel length offset data (from geometry table, see chapter 13)

For details on programming a T word, see page 10-36. If you program a T word that contains a change in dresser/wheel radius after dresser/wheel radius compensation is activated, the next block that contains axis motion in the current plane must be a linear block.

Important: The dresser/wheel radius compensation feature is not available for any motion blocks that are programmed in MDI mode (see page 15-55). You can alter the dresser/wheel radius compensation mode by programming either G41, G42, or G40; or you can change the dresser/wheel radius in an MDI program. However, the control does not compensate any of the dresser/wheel paths executed in MDI mode. Any changes made to dresser/wheel radius compensation are not applied until the next block executed in automatic mode.

The orientation data called with a T word is used when determining dresser/wheel positioning information relative to the part. For details on orientation values, see page 3-8.

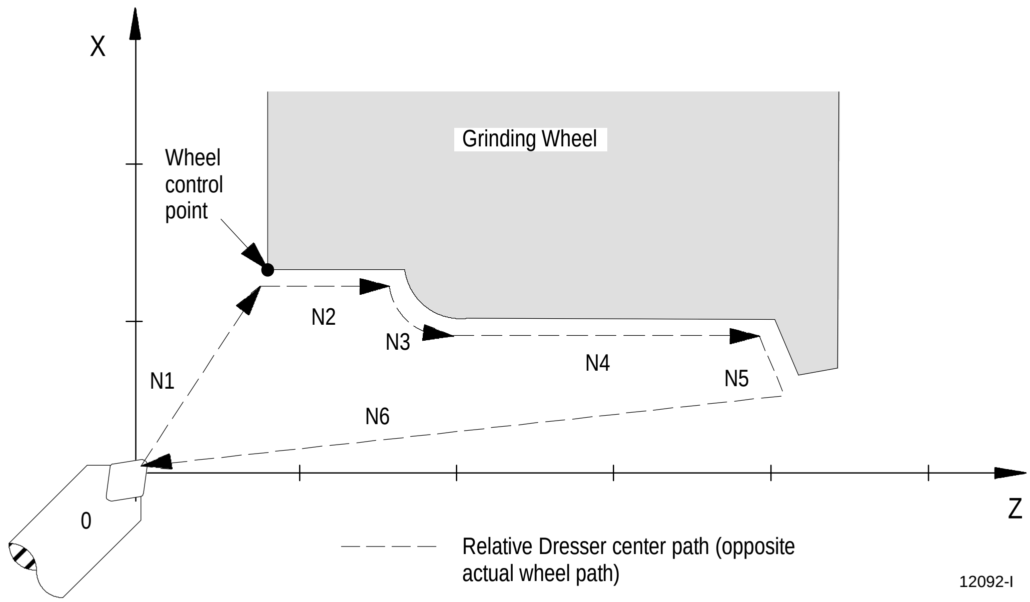
G40 (dresser/wheel radius compensation cancel) is active when power is turned on, E-STOP is reset, the control is reset, or the control executes an M02 or M30 end-of-program block.

Example 15.2
Dresser/Wheel Radius Compensation Sample Path

```
Assume: T0100 = 5mm (radius offset table)
                orientation = 0
```

Program Block	Comment
N1G00G41X0.Z0.T0102;	rapid to start and set dresser/wheel radius compensation left
N2G01Z-10.F.1.;	feed move to Z-10
N3G02Z-15.X5.R5.;	arc of radius 5
N4G01Z-35.;	feed move to Z-35
N5Z-38X10.;	feed move to Z-38 X10
N6G40X12.Z7.50;	exit dresser/wheel radius compensation
N7 M30	end of program

Figure 15.12
Results of Dresser/Wheel Radius Compensation Program Example



15.3 Generated Compensation Blocks G39, G39.1

In certain instances, dresser/wheel radius compensation creates a non-programmed move called a generated block. These blocks improve cycle time and corner-cutting quality.

Dresser/wheel radius 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
Dresser/Wheel Path	Block is generated	Block is generated
Exit Move	No block is generated	Block is generated

Important: These blocks are created only if:

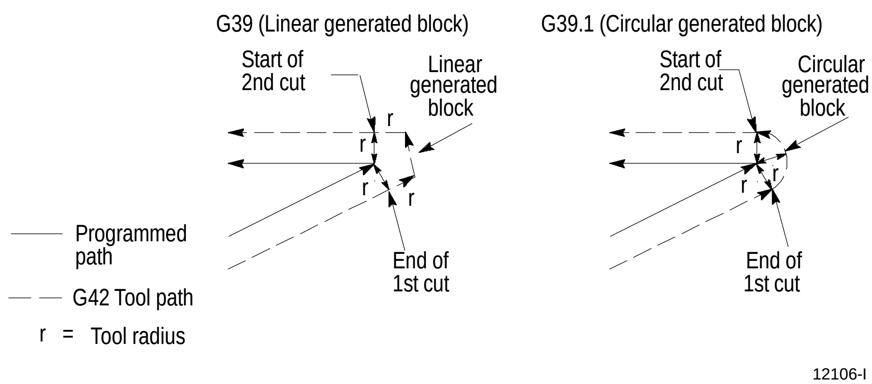
- G41 is active and you program an inside angle that is less than 90°
- G42 is active and you program an outside angle that is more than 270°

You can program the generated block between the two dresser/wheel paths as linear or circular with these G-codes:

```
G39(or G39.1);
```

Where :	Causes:
G39	linear transition blocks. If you program a G39 or G39.1, G39 is the default (established at control reset or end of program command). This command is modal.
G39.1	circular transition blocks. When programming straight line-to-arc or arc-to-straight line moves, the generated block is always linear, and the control ignores the G39.1. This command is modal.

Figure 15.13
Dresser/Wheel Radius Compensation Generated Blocks (G39 vs G39.1)



You can program a G39 or G39.1 in any block. However, you must program them in or before the block that causes a dresser/wheel radius compensation generated block.

Important: For linear generated blocks only, your 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 and the dresser/wheel 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.

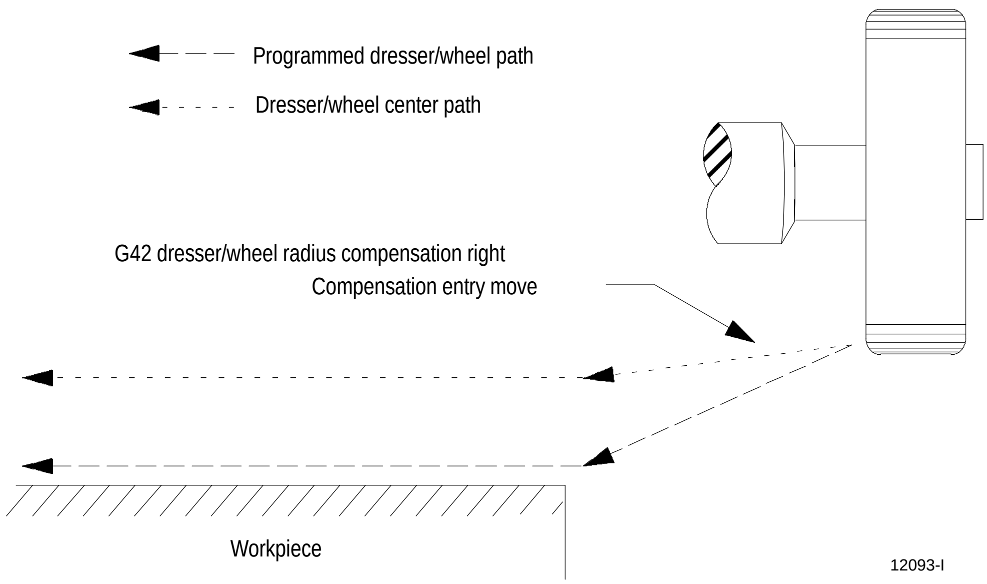
15.4 Type A Compensation Paths

We use pictorial representation to demonstrate the actual dresser/wheel paths taken when using dresser/wheel radius compensation type A. The following subsections give brief descriptions of the paths along with figures to clarify the descriptions.

15.4.1 Type A Compensation Entry Moves

We define an entry move as the path that the dresser/wheel takes when the radius compensation function first becomes activated in a program. Figure 15.14 gives an example of a typical entry move. An entry move must be a linear block.

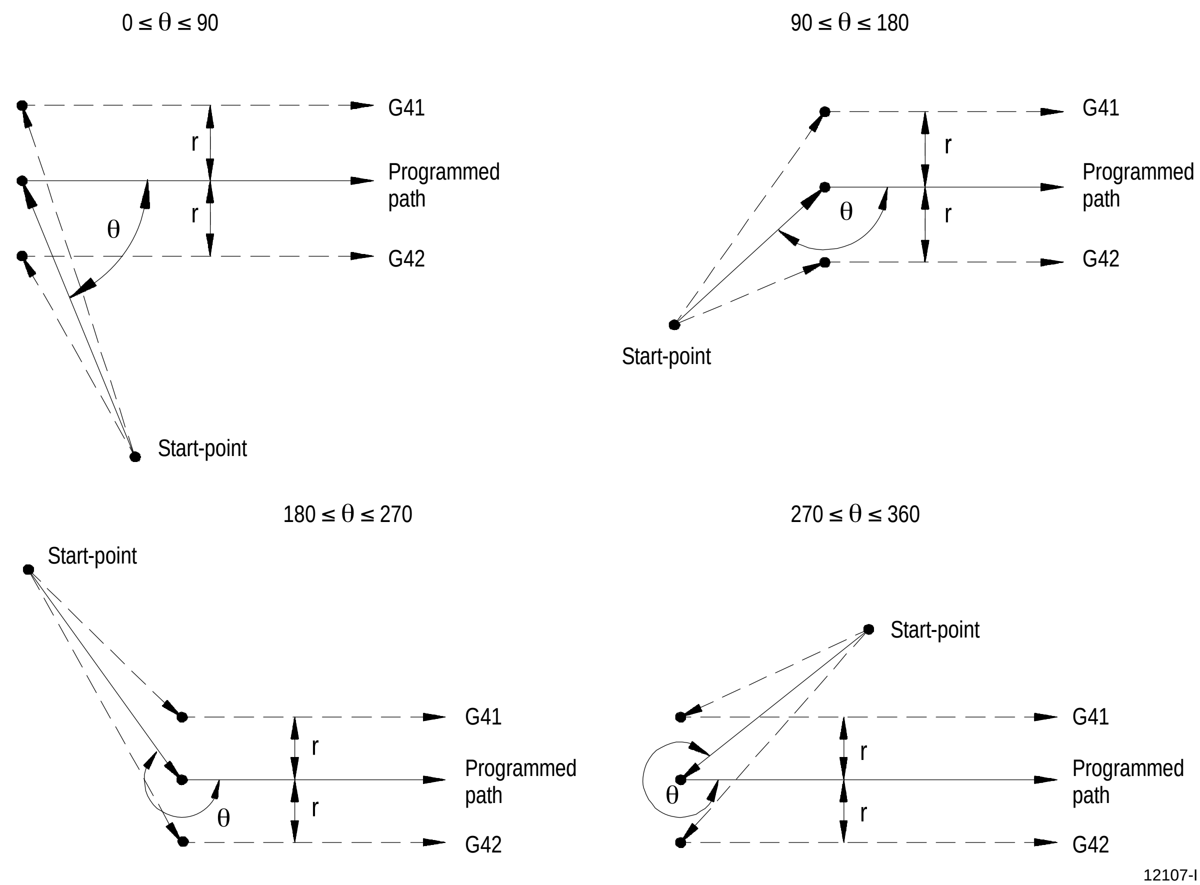
Figure 15.14
Dresser/Wheel Radius Compensation Entry Move



The entry move of the dresser/wheel for type A radius 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.

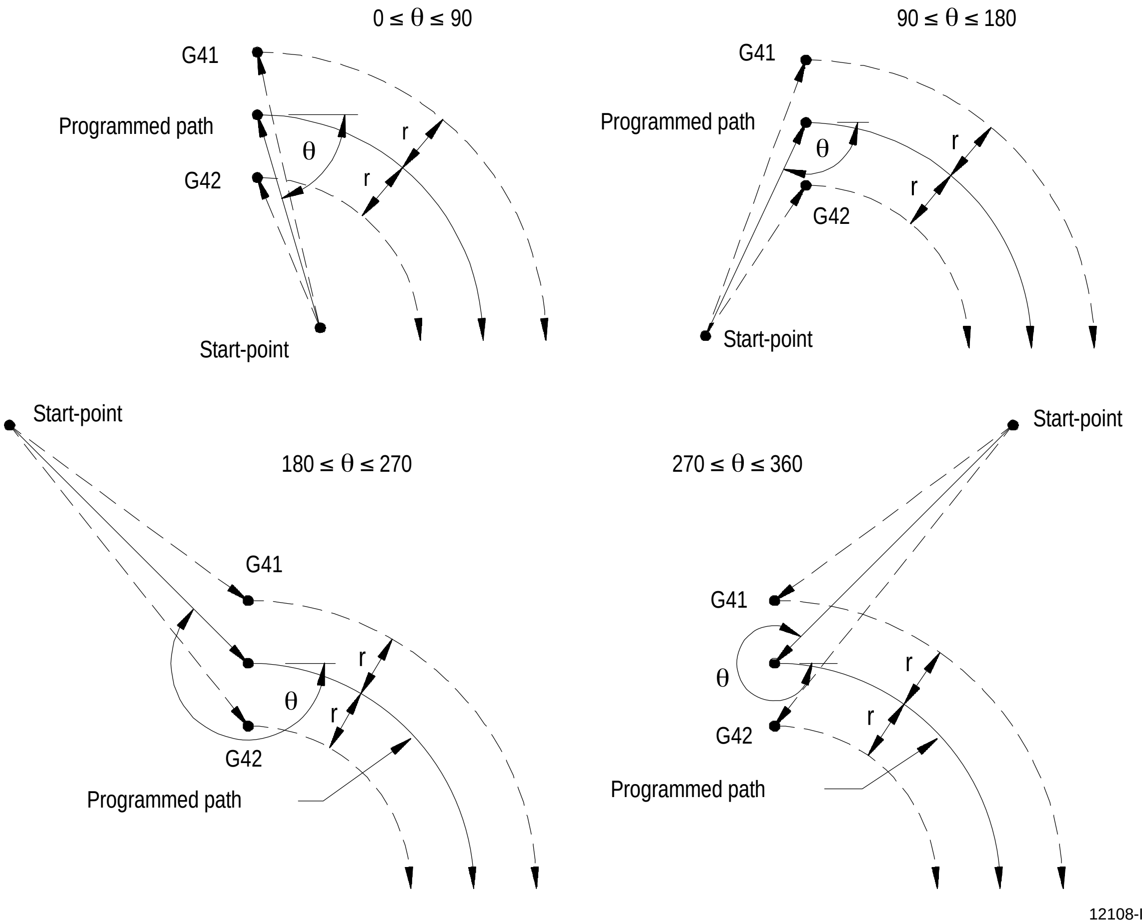
Figure 15.15 and Figure 15.16 show examples of typical entry moves using type A radius compensation.

Figure 15.15
Dresser/Wheel Path for Entry Move Straight Line-to-Straight Line



If the move following the entry move is an arc, the dresser/wheel is positioned at right angles to a tangent line drawn from the start-point of that circular move.

Figure 15.16
Dresser/Wheel Path for Entry Move Straight Line-to-Arc



There is no limit to the number of blocks that can 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 compensation plane.

Example 15.3
Sample Entry Move After Non-Motion Blocks

`Assume current compensation plane is the ZX plane.`

`N01X0Z0;`	
`N2G41T1;`	This block commands compensation left
`N3M02;`	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.
″	"
″	"
″	"
`N999X1Z1;`	This is the entry move for the previously programmed G41.

Although there is no limit to the number of non-motion blocks allowed before entering compensation, your system installer must select in AMP the number of non-motion blocks that is allowed during dresser/wheel radius compensation. If you exceed this number of non-motion blocks during compensation, you must re-initialize dresser/wheel radius compensation with a new entry move. For details on too many non-motion blocks during compensation, see page 15-46.

15.4.2 Type A Compensation Exit Moves

Cancel the radius compensation feature by programming a G40. The path that is taken when the dresser/wheel leaves radius compensation is referred to as the exit move. The path that the dresser/wheel follows during an exit move depends on:

- The direction of compensation (G41 or G42)
- The angle between the last motion made in dresser/wheel radius compensation (in the current compensation plane) and the motion of the exit move

Selecting a dresser/wheel radius offset number T0000 in a program does not cancel radius compensation and does not generate an exit move. Radius compensation continues on as if a dresser/wheel radius had been changed to a radius of zero. See page 15-51 for details on changing dresser/wheel radius. The exit move, if T0000 is the active dresser/wheel radius, is the same path as the programmed dresser/wheel path.

Important: An exit move cannot be a circular move (G02 or G03). You must program all exit moves on a linear path. Any attempt to generate an exit move using a circular path generates a block-format error.

Example 15.4
Type A Sample Exit Moves

Assume the current plane is the ZX plane and dresser/wheel radius compensation is already active before the execution of block N100 in the following program segments.

`N100X1.Z1.;`	
`N110X3.Z3.G40;`	Exit move.
`N100X1.Z1.;`	
`N110G40;`	
`N120X3.Z3.;`	Exit move.

`N100X1.Z1.;`	
`N110G40;`	
`N120;`	No axis motion in the current plane.
`N130...;`	No axis motion in the current plane.
`N140...;`	No axis motion in the current plane.
"	"
"	"
`N200X3.Z3.;`	Exit move.

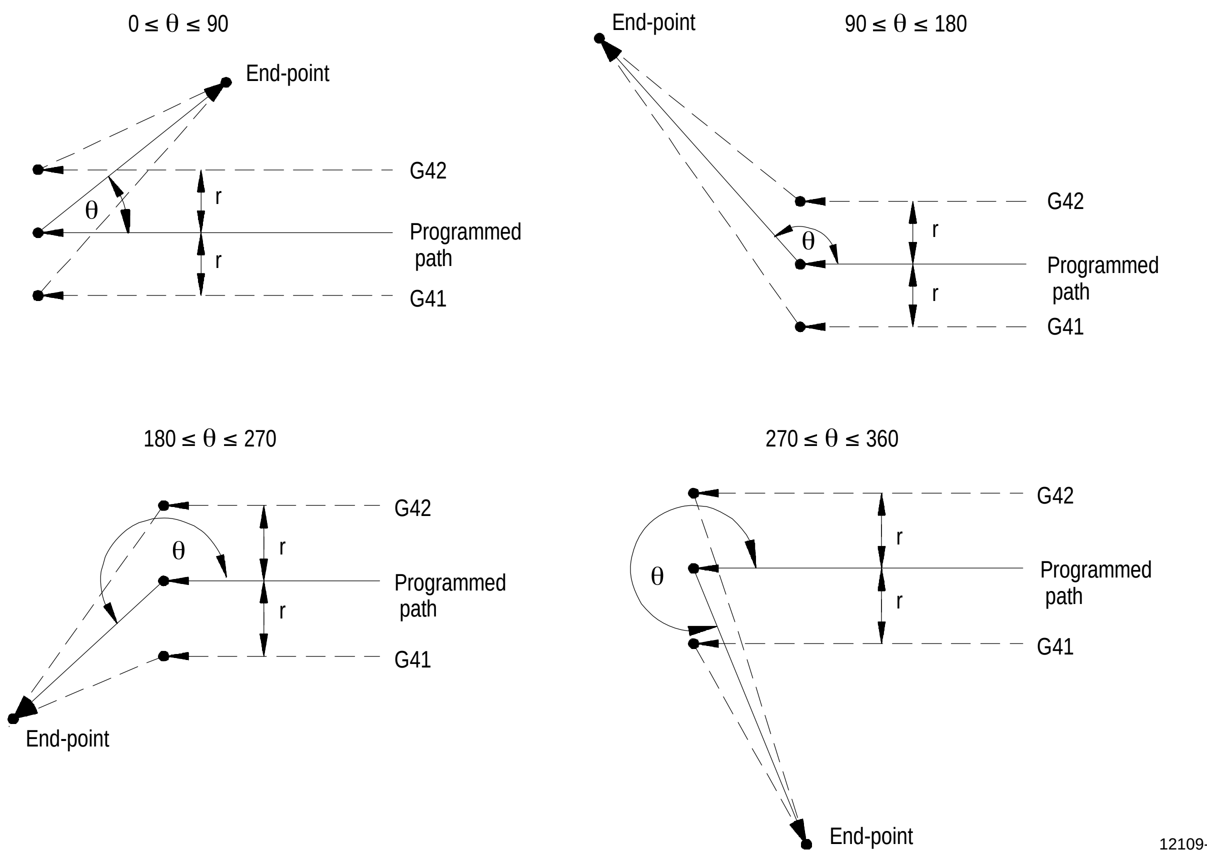
`N100X1.Z1.;`	
`N110...;`	No axis motion in the current plane.
`N120...;`	No axis motion in the current plane.
`N130...;`	No axis motion in the current plane.
"	"
"	"
`N200G40X3.Z3.;`	Exit move.

All of the program blocks in Example 15.4 produce the same exit move, provided the number of non-motion blocks in the compensation mode has not exceeded a value selected by your system installer in AMP.

The exit of the dresser/wheel for type A radius compensation takes the shortest possible path to the endpoint of the exit move. This path starts at right angles to the left or right of the endpoint (depending on G41 or G42) of the last move in the currently defined plane (you can redefine this start-point using an I and/or K word as described on page 15-24). The end-point of the exit move is no longer offset to the left or right.

Figure 15.17 through Figure 15.21 show examples of typical exit moves using type A radius compensation. All examples assume that the number of non-motion blocks before the G40 command has not exceeded the number allowed, as determined by your system installer in AMP.

Figure 15.17
Dresser/Wheel Path for Exit Move Straight Line-to-Straight Line



If the last programmed move is circular (an arc), the dresser/wheel is positioned at a right angle to a tangent line drawn from the end-point of that circular move.

Figure 15.18
Dresser/Wheel Path for Exit Move Arc-to-Straight Line

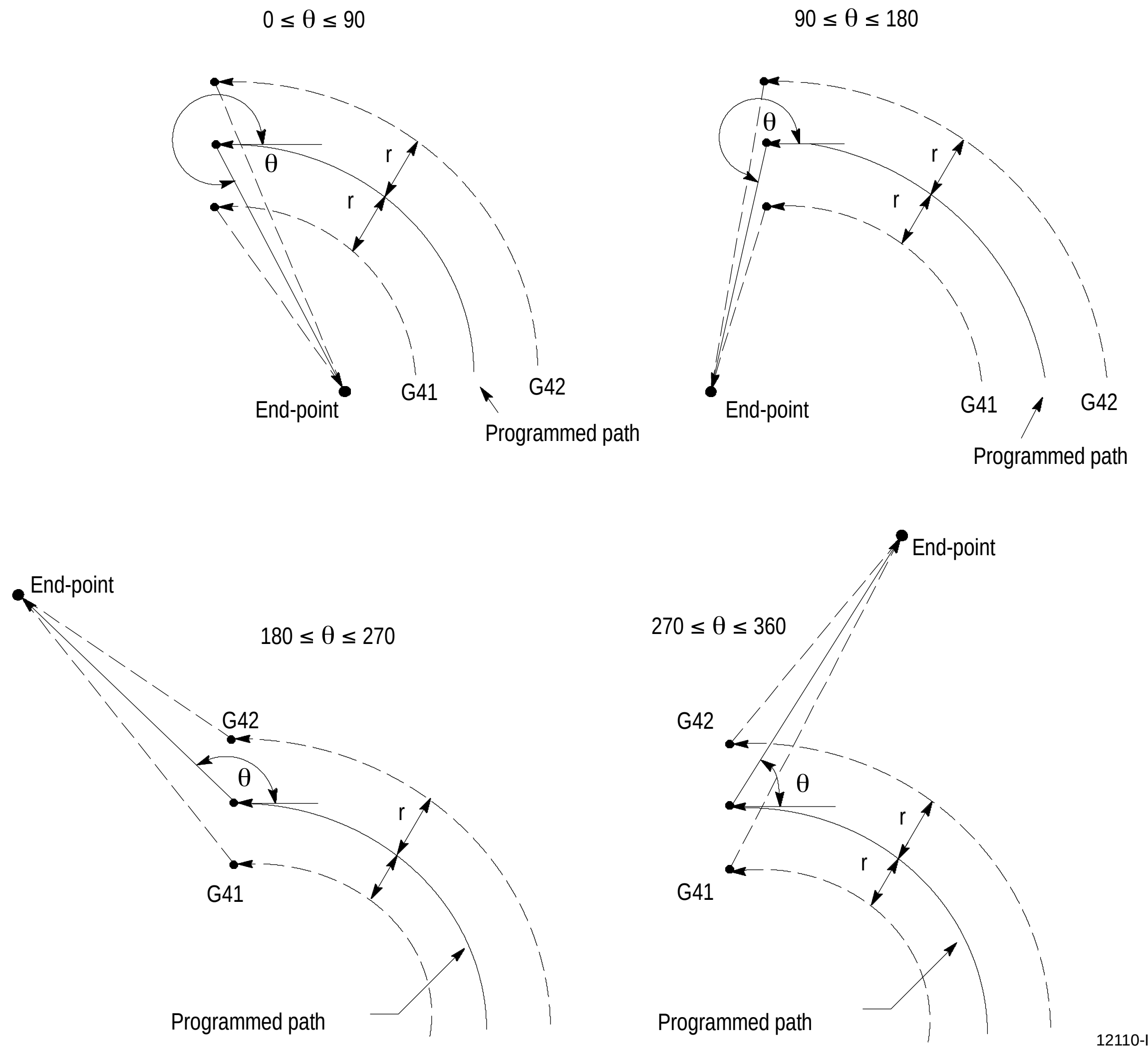


Figure 15.17 and Figure 15.18 assume that the number of non-motion blocks does not exceed an amount selected in AMP by your system installer. If the number of non-motion blocks following G40 exceeds the limit, the control generates its own exit move. This can often cause overcutting of the part. See page 15-46 for details on too many non-motion blocks.

I and K Vector in an Exit Move

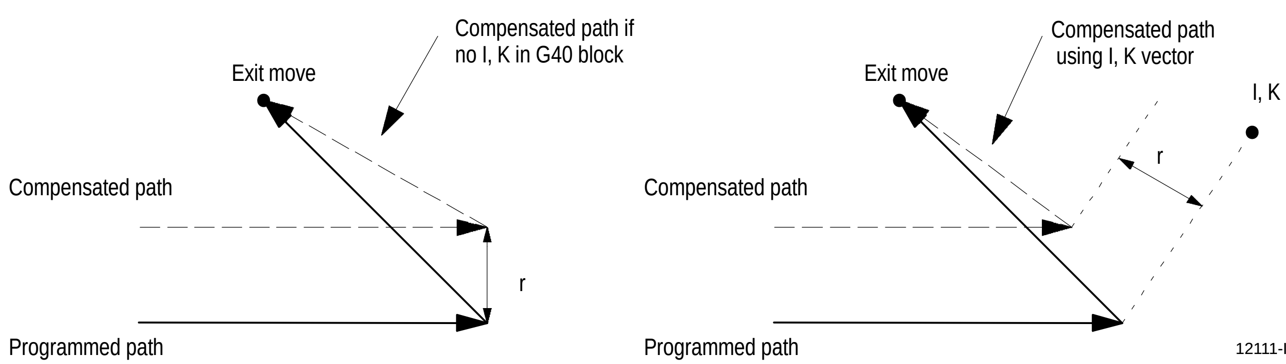
By including an I and/or K word in the exit move, you can modify the path that the dresser/wheel takes for an exit move. Only the I or K words that represent values in the current plane are programmed in the block containing the exit move.

Important: I and K are integrand words corresponding to the X and Z axes respectively. If you are using different axes, or if your system installer has assigned different names for the axis integrands, you must substitute them for the names used here.

The I and K words in the exit move block define a vector that the control uses to redefine the end-point of the previously compensated move. You always program I and K words as incremental values regardless of the current mode (G90 or G91).

Use I and K to define an end-point of an imaginary block in radius compensation. By programming an I, K, the control assumes that this is the end-point of an imaginary linear block. The control applies compensation to this imaginary block to modify the end-point of the last compensated block programmed. I and K allow you to compensate for future moves or protect areas that should not be machined.

Figure 15.19
Exit Move Defined by an I, K Vector



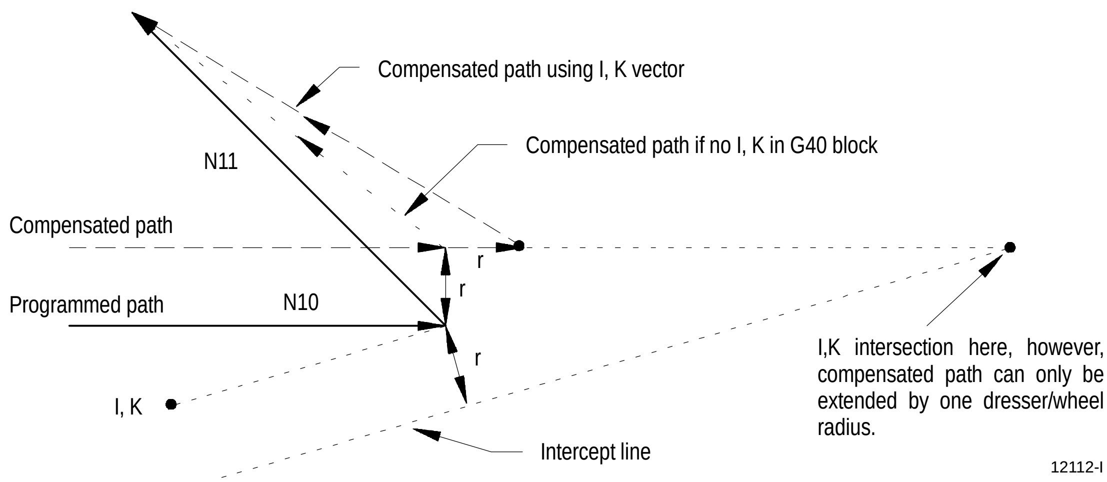
There is a limit to the amount that an I, K vector can modify the last compensated block. An I, K vector can only shorten/lengthen the last compensated block by an amount equal to one active dresser/wheel radius (see Example 15.5). The direction of the offset is towards the point of intersection of the I, K vector and the current compensated dresser/wheel path.

Example 15.5
Exit Move Defined by an I,K Vector but Limited to Dresser/Wheel Radius

```
Assume T1 radius is 3
```

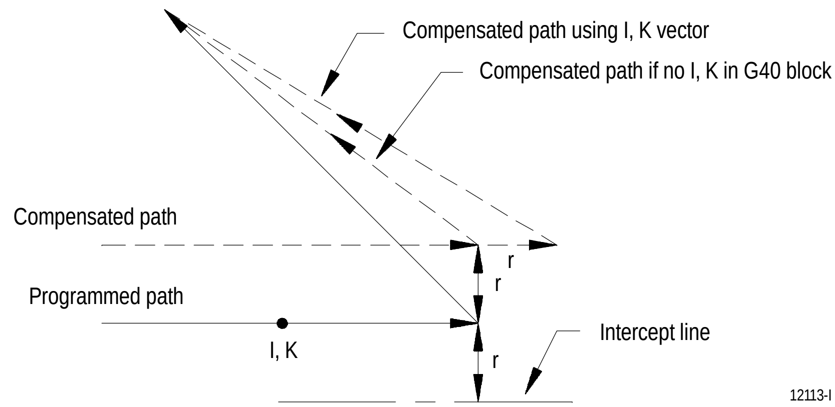
```
N10 Z10.G41T1
N11 X10.Z2.I3K-10.G40;
```

Figure 15.20
Results of Example 15.5



If the vector defined by I and/or K is parallel to the programmed dresser/wheel path, the resulting exit move is offset in the opposite direction of the I and/or K vector by one radius of the dresser/wheel. See Figure 15.21.

Figure 15.21
Exit Move When I, K Vector Is Parallel to Programmed Dresser/Wheel Path



Important: If you do not program I and/or K, their value defaults to 0.

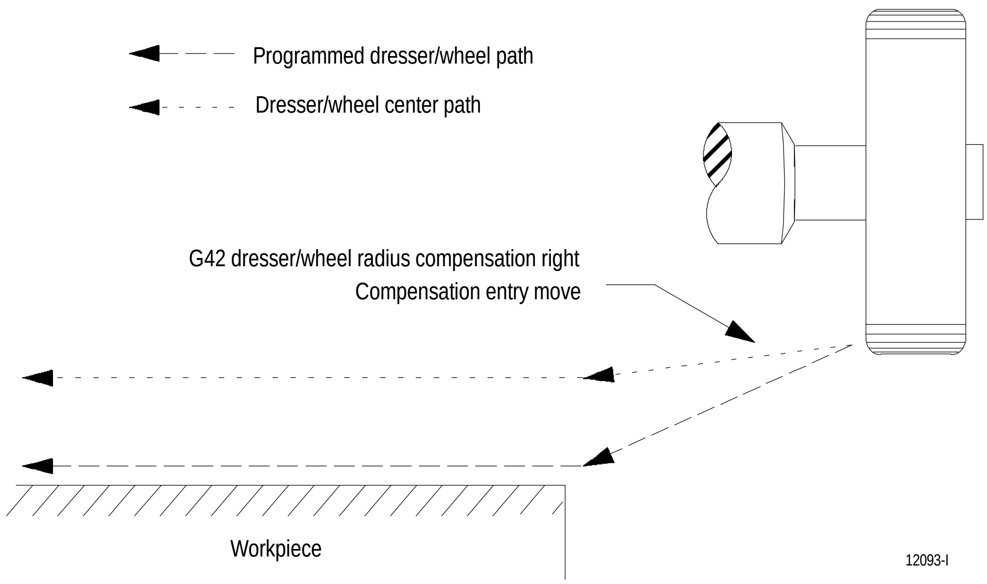
15.5 Type B Compensation Paths

We use pictorial representation to demonstrate the actual dresser/wheel paths taken by the dresser/wheel when using radius compensation type B. The following subsections give brief descriptions of the paths along with figures.

15.5.1 Type B Compensation Entry Moves

We define an entry move as the path that the dresser/wheel takes when the dresser/wheel radius compensation function first becomes activated in a program. Figure 15.22 is an example of a typical entry move. An entry move must be a linear block.

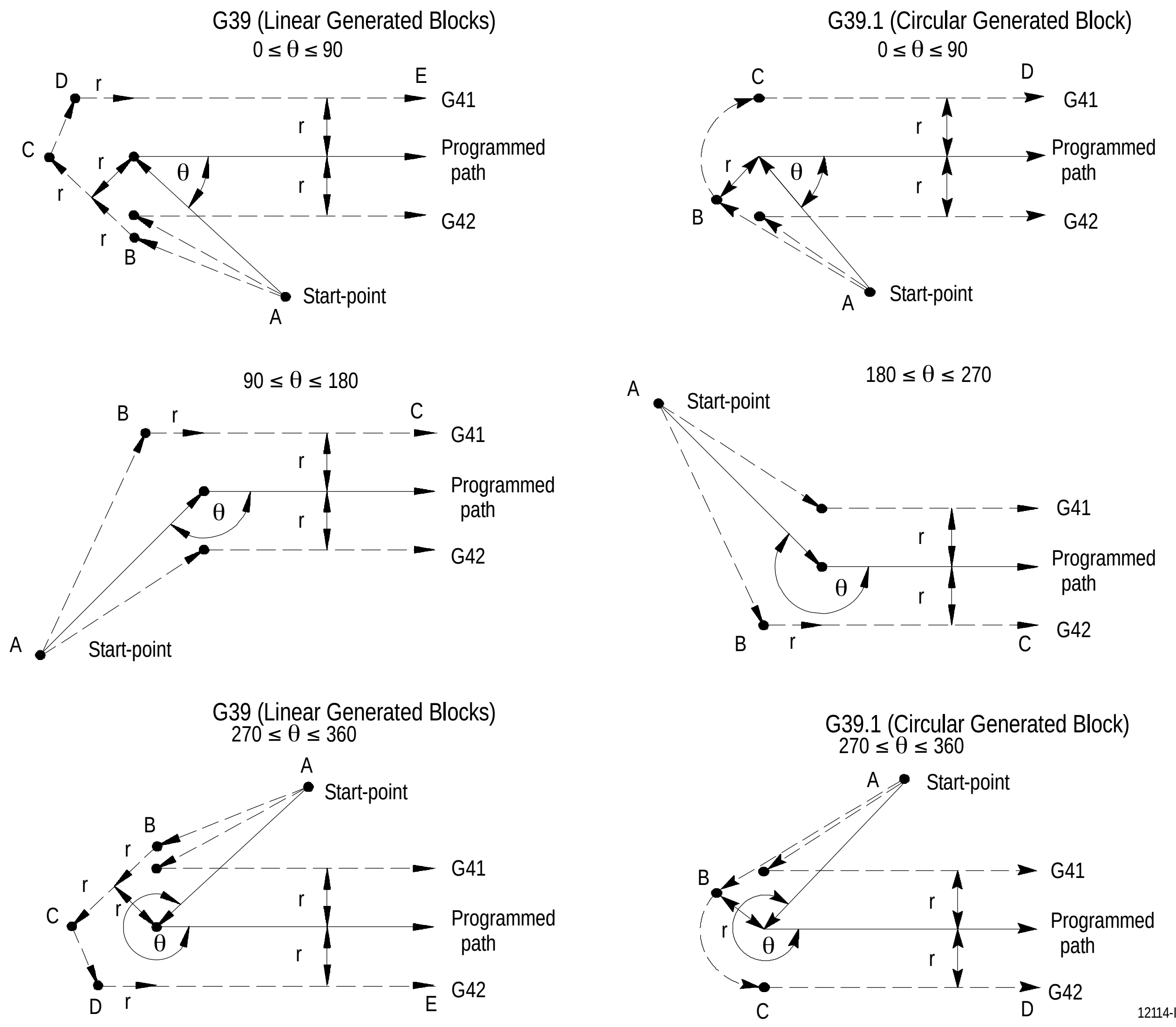
Figure 15.22
Dresser/Wheel Radius Compensation Entry Move



The entry move of the dresser/wheel for type B radius compensation can generate extra motion blocks to attempt to prevent gouging of the part, as can sometimes occur using compensation type A. Type B radius compensation keeps the dresser/wheel 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.

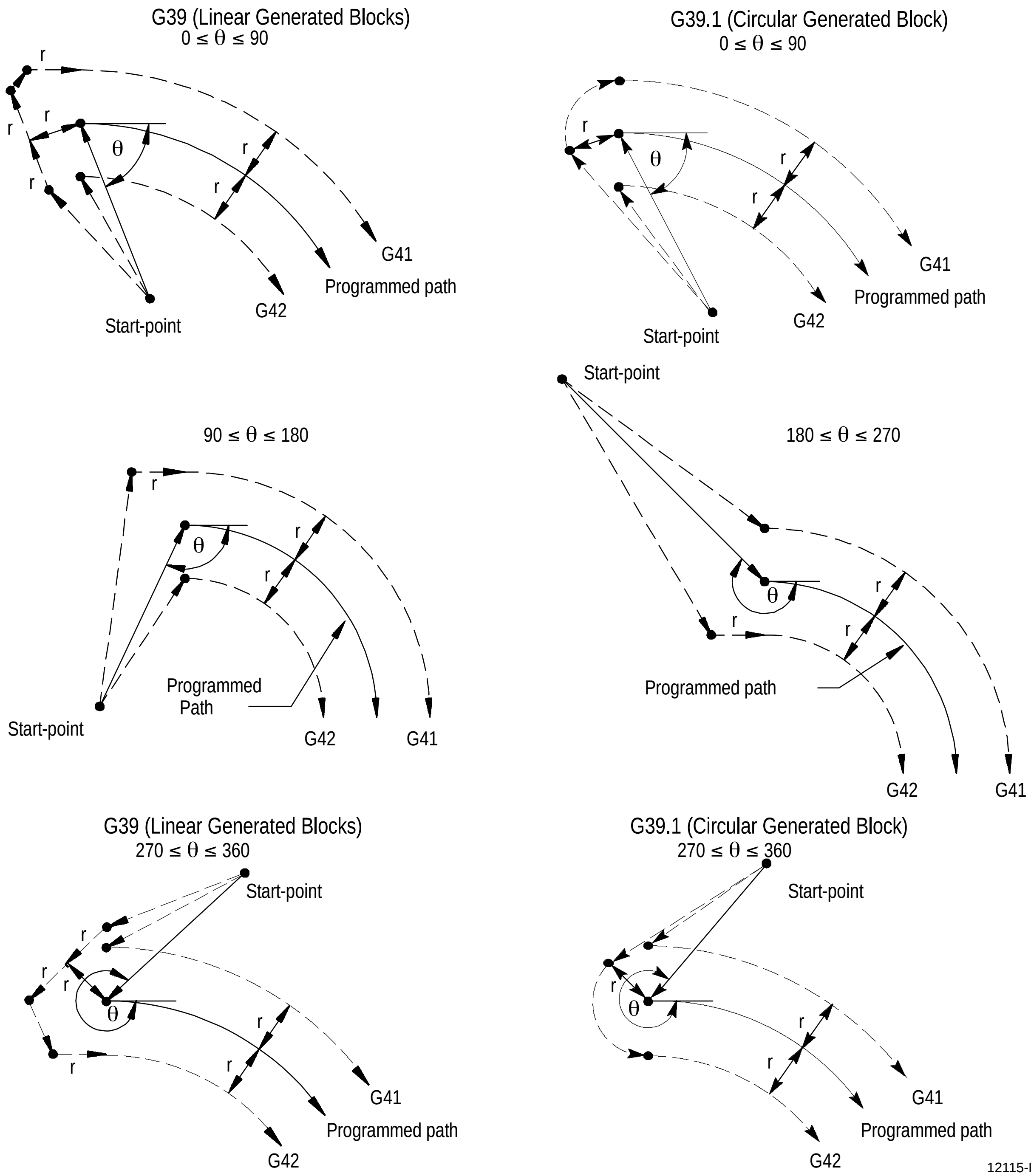
Figure 15.23 and Figure 15.24 show examples of typical entry moves using type B radius compensation.

Figure 15.23
Dresser/Wheel Path for Entry Move Straight Line-to-Straight Line



If the next programmed move is circular (an arc), the dresser/wheel is positioned at right angles to a tangent line drawn from the start-point of that circular move.

Figure 15.24
Dresser/Wheel Path for Entry Move Straight Line-to-Arc



There is no limit to the number of blocks that can 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 15.6
Sample Entry Move After Non-Motion Blocks

```
Assume current compensation plane is the ZX plane.

N01X0Z0;
N2G41;          This block commands compensation left.
N3M02;          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.
   "                 "
   "                 "
   "                 "
N999X1Z1;       This is the entry move for the previously
                programmed G41.
```

Although there is no limit to the number of non-motion blocks allowed before entering compensation, your system installer must select in AMP the number of non-motion blocks that are allowed during dresser/wheel radius compensation. If this number of non-motion blocks is exceeded during compensation, dresser/wheel radius compensation must be re-initialized with a new entry move being generated. For details on too many non-motion blocks during compensation, see page 15-46.

15.5.2 Type B Compensation Exit Moves

Cancel the radius compensation feature by programming a G40. The path that is taken when the dresser/wheel leaves radius compensation is referred to as the exit move. The path that the dresser/wheel follows during an exit move depends on:

- The direction of compensation (G41 or G42)
- The angle between the last motion made in dresser/wheel radius compensation (in the current compensation plane) and the motion of the of the exit move

Selecting a dresser/wheel offset number T0000 in a program does not cancel radius compensation and does not generate an exit move. Radius compensation continues on as if a dresser/wheel radius had been changed to a radius of zero. See page 15-51 on changing dresser/wheel radius. If T0000 is the active dresser/wheel radius, the exit move is equal to the programmed dresser/wheel path.

Important: An exit move cannot be a circular move (G02 or G03). You must program all exit moves on a linear path. Any attempt to generate an exit move using a circular path generates an error.

Example 15.7 shows examples of exit move program blocks.

Example 15.7
Sample Exit Move Segments

Assume the current plane to be the ZX plane.

```
N100X1Z1;
N110X3Z3G40;   Exit move.

N100X1Z1;
N110G40;
N120X3Z3;      Exit move.
```

```
N100X1Z1;
N110G40;
N120...;       No axis motion in the current plane.
N130...;       No axis motion in the current plane.
N140...;       No axis motion in the current plane.
   "                "
   "                "
N200X3Z3;      Exit move.
```

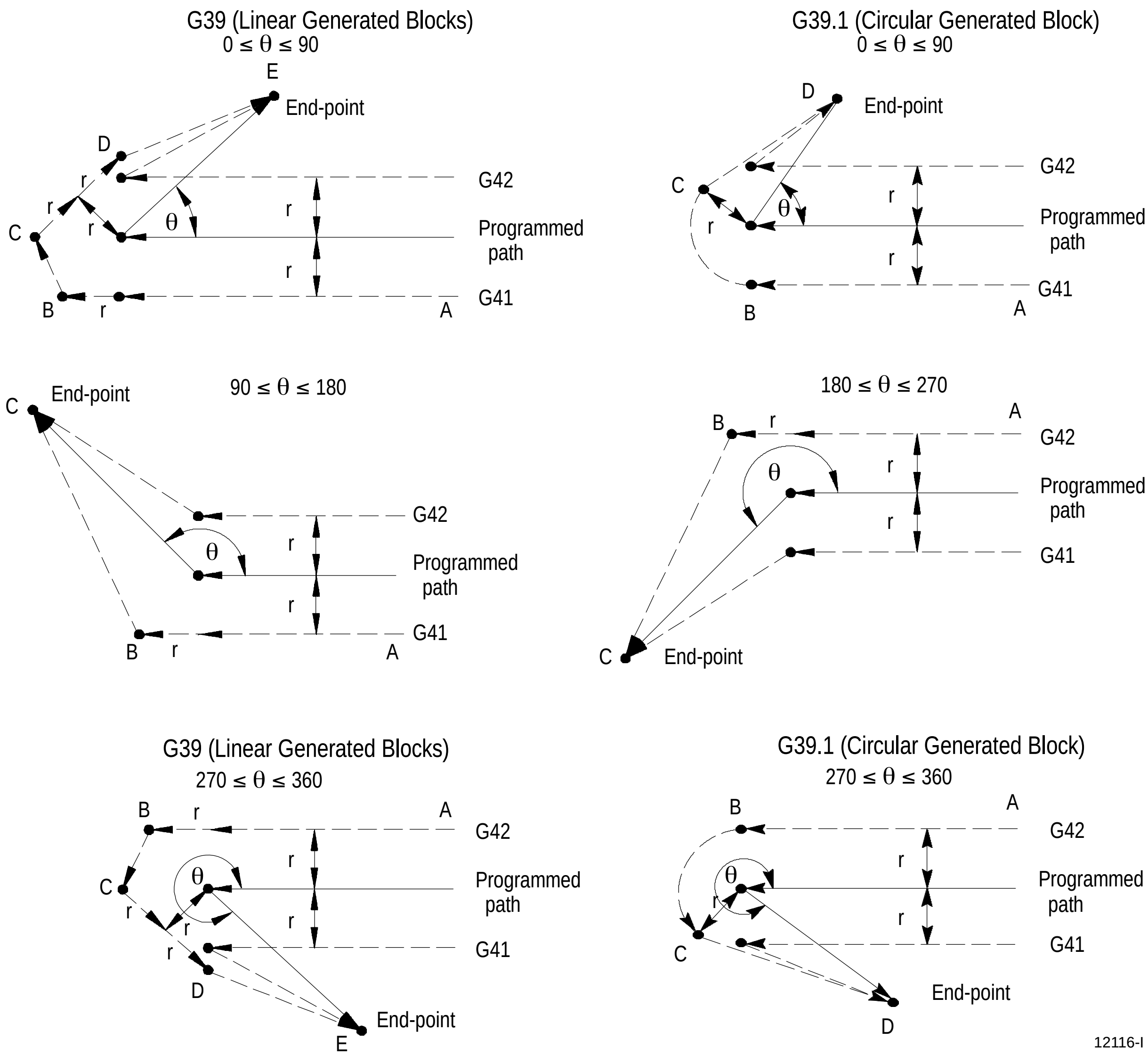
```
N100X1Z1;
N110...;       No axis motion in the current plane.
N120...;       No axis motion in the current plane.
N130...;       No axis motion in the current plane.
   "                "
   "                "
200G40X3Z3;    Exit move.
```

If the number of non-motion blocks in the compensation mode has not exceeded a value selected by your system installer in AMP, all of the program blocks in Example 15.7 produce the same exit move.

The exit of the dresser/wheel for type B radius compensation takes the shortest possible path to the end-point of the exit move for inside corners only. For outside corners, the dresser/wheel always remains at least the radius of the dresser/wheel away from the end-point of the last move in compensation. You can redefine the start-point using an I and/or K word as described in this section later. The end-point of the exit move is no longer offset to the left or right.

Figure 15.25 and Figure 15.26 show examples of typical exit moves using type B radius compensation. All examples assume the number of non-motion blocks before the designation of the G40 command has not exceeded the number allowed as determined by your system installer in AMP.

Figure 15.25
Dresser/Wheel Path for Exit Move Straight Line-to-Straight Line



If the last programmed move is circular (an arc), the dresser/wheel is positioned at a right angle to a tangent line drawn from the end-point of that circular move.

Figure 15.26
Dresser/Wheel Path for Exit Move Arc-to-Straight Line

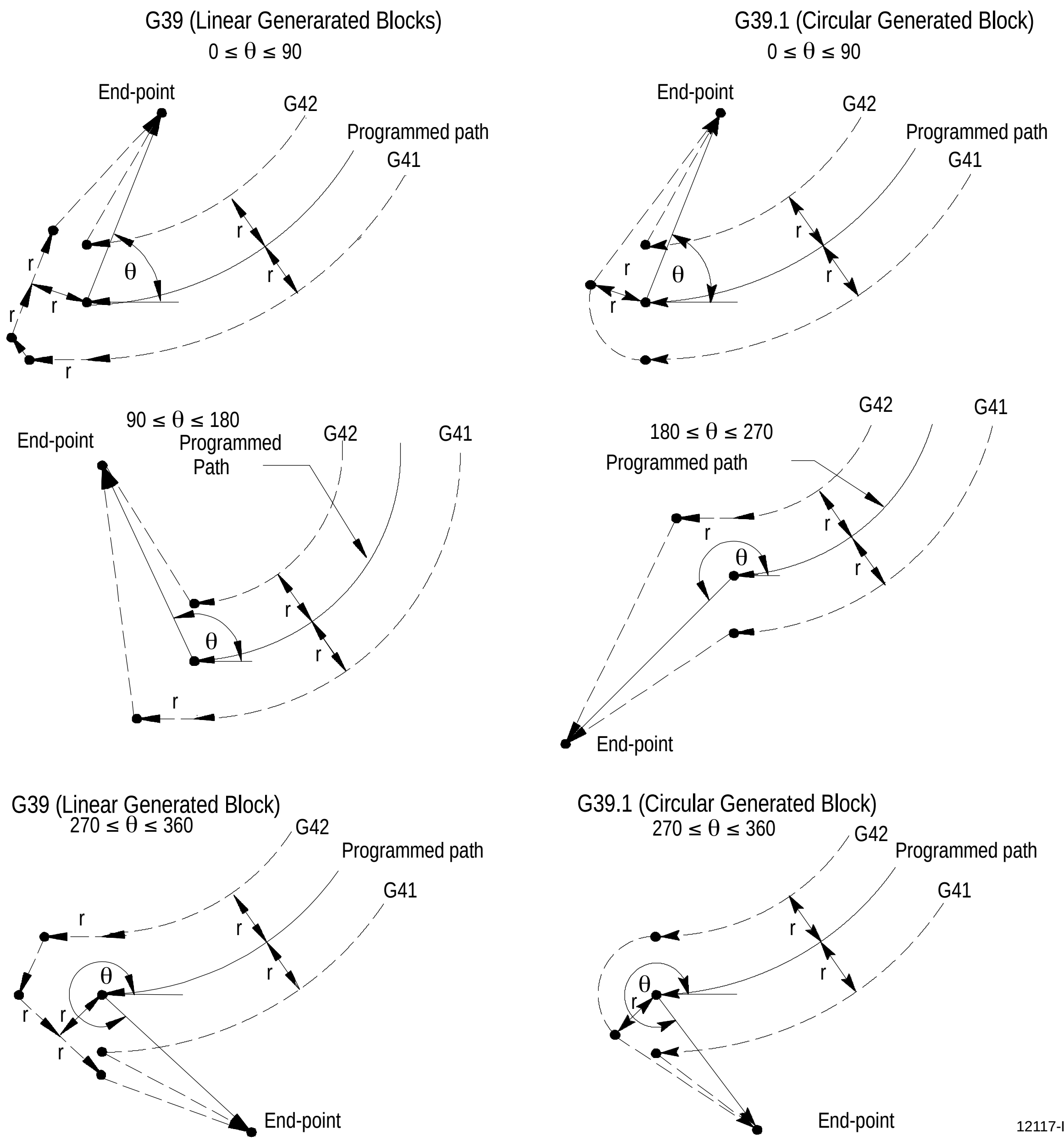


Figure 15.25 and Figure 15.26 assume that the number of blocks not containing axes motion in the currently selected plane, following G40 before the exit move takes place, does not exceed an amount selected in AMP by your system installer. If the number of non-motion blocks following G40 exceeds the limit, the control generates its own exit move. This can often cause overcutting of the part because this move is a linear path directly back to the programmed dresser/wheel path.

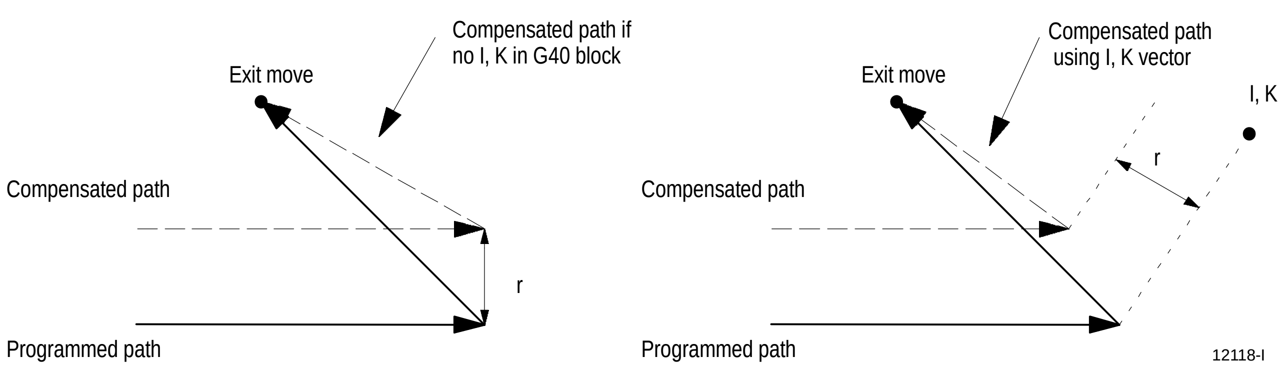
I and K Vector in an Exit Move

By including an I and/or K word in the exit move, you can modify the path that the dresser/wheel takes for an exit move. Only the I or K words that represent values in the current plane are programmed in the block containing the exit move. I and K are integrand words corresponding to the X and Z axes respectively.

The I and K words in the exit move block define a vector that the control uses to redefine the end-point of the previously compensated move. You always program I and K words as incremental values regardless of the current mode (G90 or G91).

Use I and K to define an endpoint of an imaginary block in dresser/wheel radius compensation. By programming an I, K, the control assumes that this is the end-point of an imaginary linear block. The control applies compensation to this imaginary block to modify the end-point of the last compensated block programmed. I and K allow you to compensate for future moves or protect areas that should not be machined.

Figure 15.27
Exit Move Defined By An I, K Vector



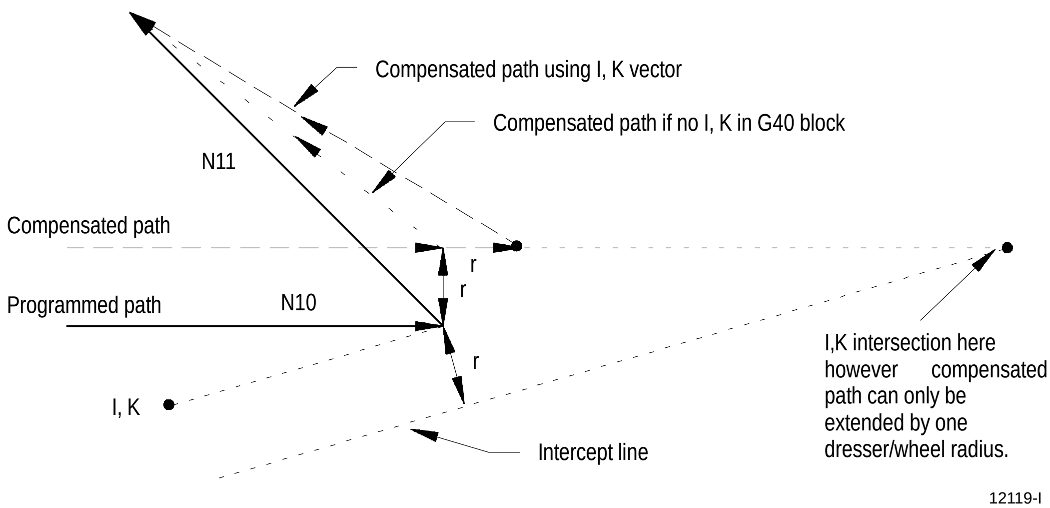
There is a limit to the size that an I, K vector can modify the last compensated block. An I, K vector can only shorten/lengthen the last compensated block by an amount equal to one active dresser/wheel radius (see Example 15.8). The offsets are directed towards the point of intersection of the I, K vector and the current compensated dresser/wheel path.

Example 15.8
Exit Move Defined By An I,K Vector But Limited To Dresser/Wheel Radius

```
Assume T1 radius is 3
```

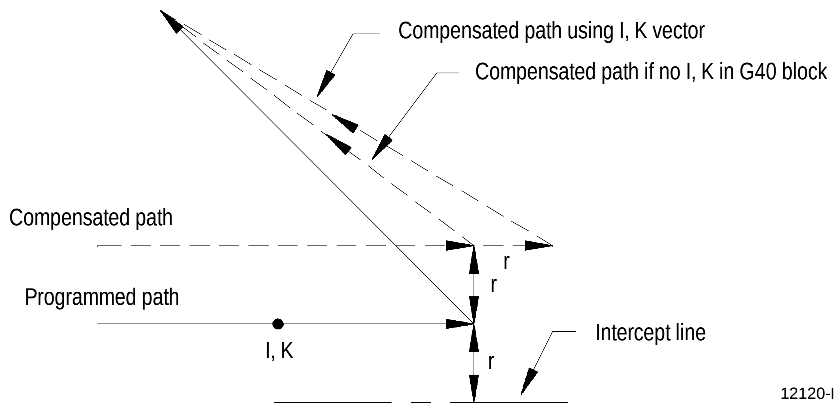
```
N10 Z10.G41T1
N11 X10.Z2.I3K-10.G40;
```

Figure 15.28
Results of Example 15.8



If the vector defined by I and/or K is parallel to the programmed dresser/wheel path, the resulting exit move is offset in the opposite direction of the I and/or K vector by one radius of the dresser/wheel (see Figure 15.29).

Figure 15.29
Exit Move When I, K Vector Is Parallel to Programmed Dresser/Wheel Path



Important: If either I and/or K is not programmed, its value defaults to 0.

15.6 Path During Compensation

Except for entry and exit moves, the basic path generated during dresser/wheel radius compensation is the same for types A and B. Whether you specify dresser/wheel left or dresser/wheel right, the path taken is a function of the angle between paths (G41 or G42) and the radius of the cutting dresser/wheel.

Important: If you perform a block reset during the execution of radius compensation, the control re-initializes the dresser/wheel radius compensation function and the next move acts as an entry move.

Important: When programming arcs with dresser/wheel radius compensation active, the control in some cases can adjust the programmed feedrate to maintain programmed speed. See chapter 12 for details on feedrates during dresser/wheel radius compensation.

When necessary, the control generates extra motion blocks to keep the dresser/wheel in tolerance of the desired path. This becomes necessary when the intersection of paths is an outside path (as defined on page 15-2) that has an angle as follows:

- Between 0° and 90° during radius compensation left (G41)
- Between 270° and 360° during radius compensation right (G42)

Figure 15.30 through Figure 15.33 illustrate the basic motion of the dresser/wheel as it executes program blocks during radius compensation.

Figure 15.30
Dresser/Wheel Radius Compensation Paths Straight Line-to-Straight Line

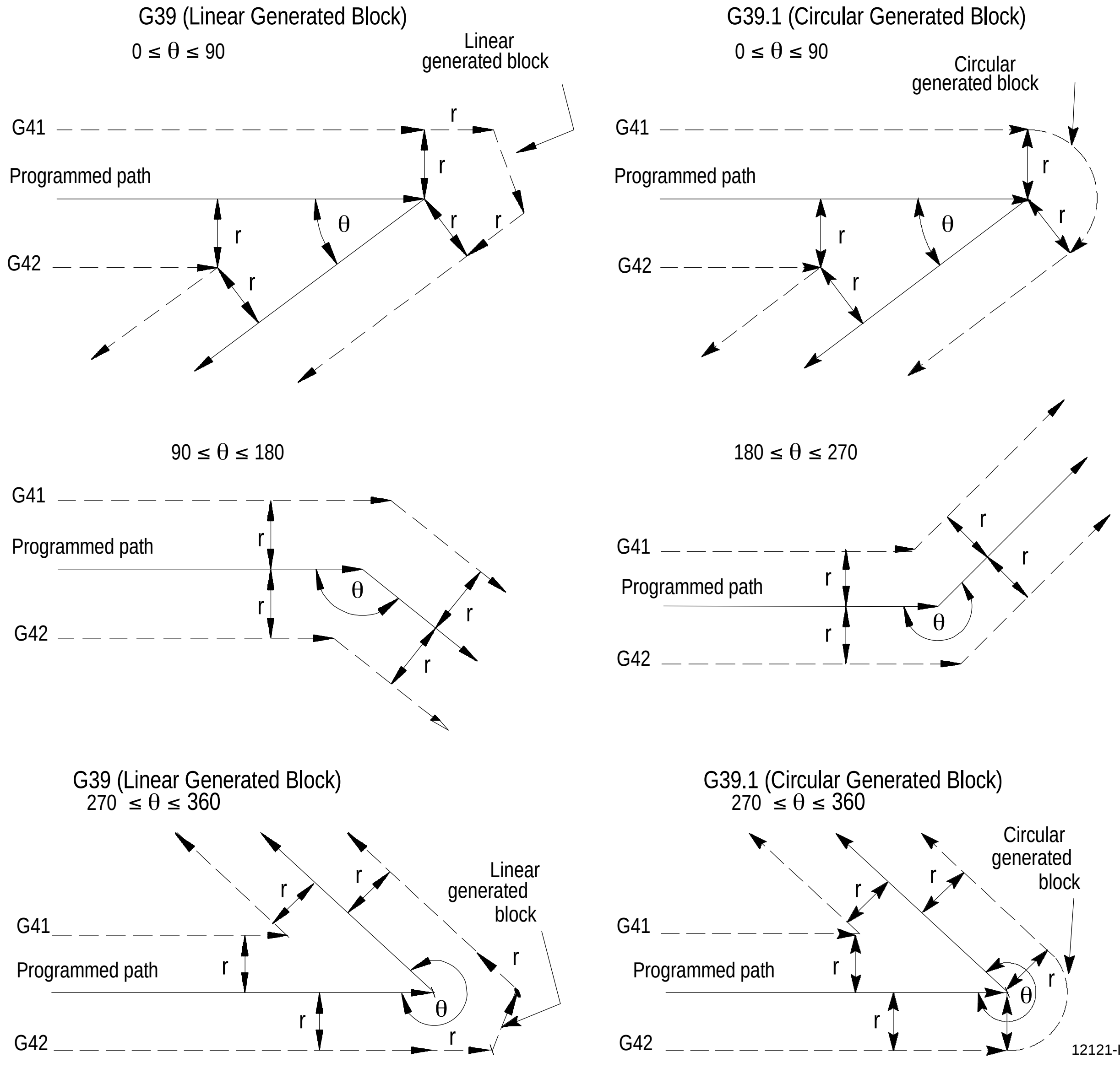


Figure 15.31
Dresser/Wheel Radius Compensation paths Straight Line-to-Arc

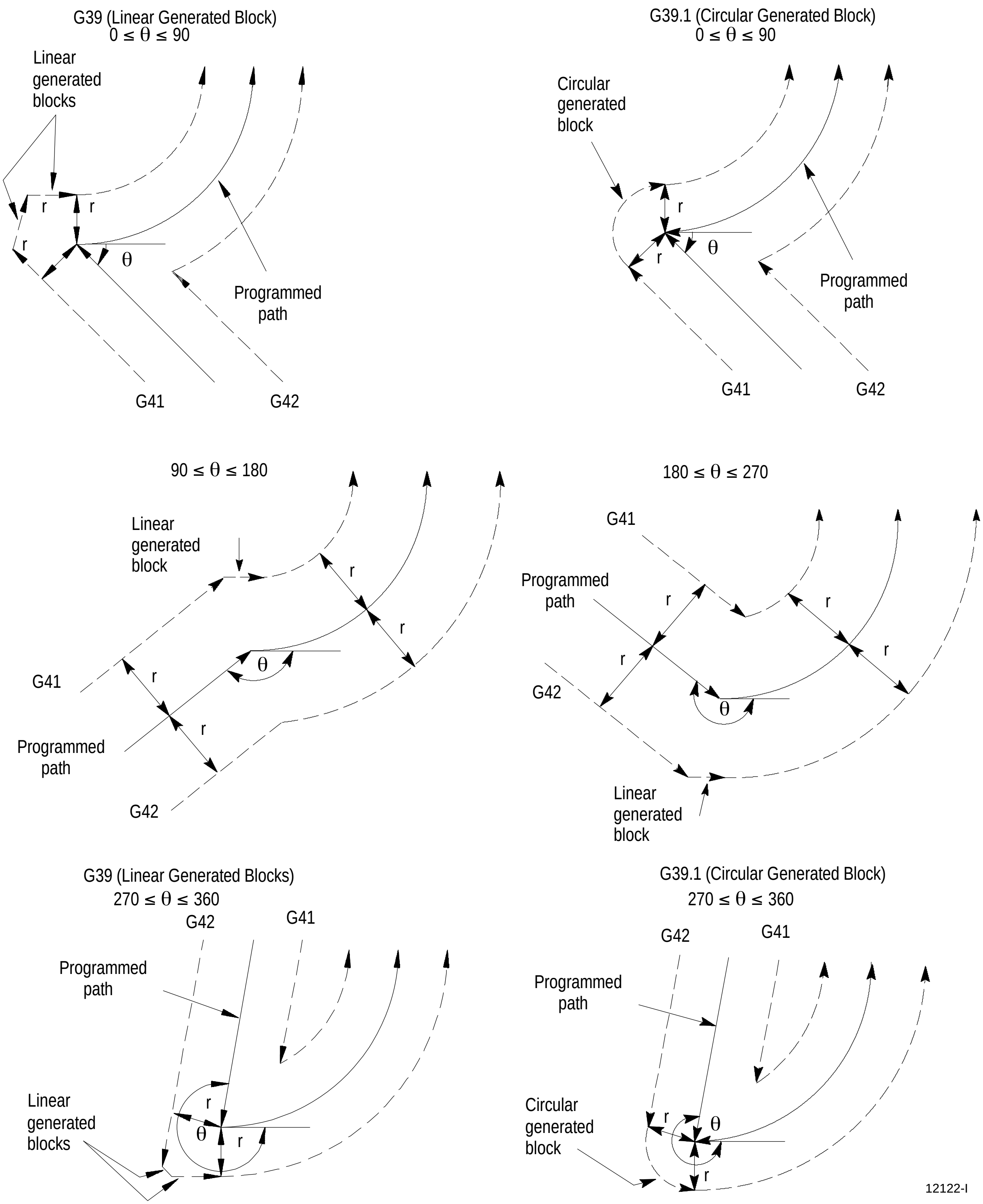


Figure 15.32
Dresser/Wheel Radius Compensation paths Arc-to-Straight Line

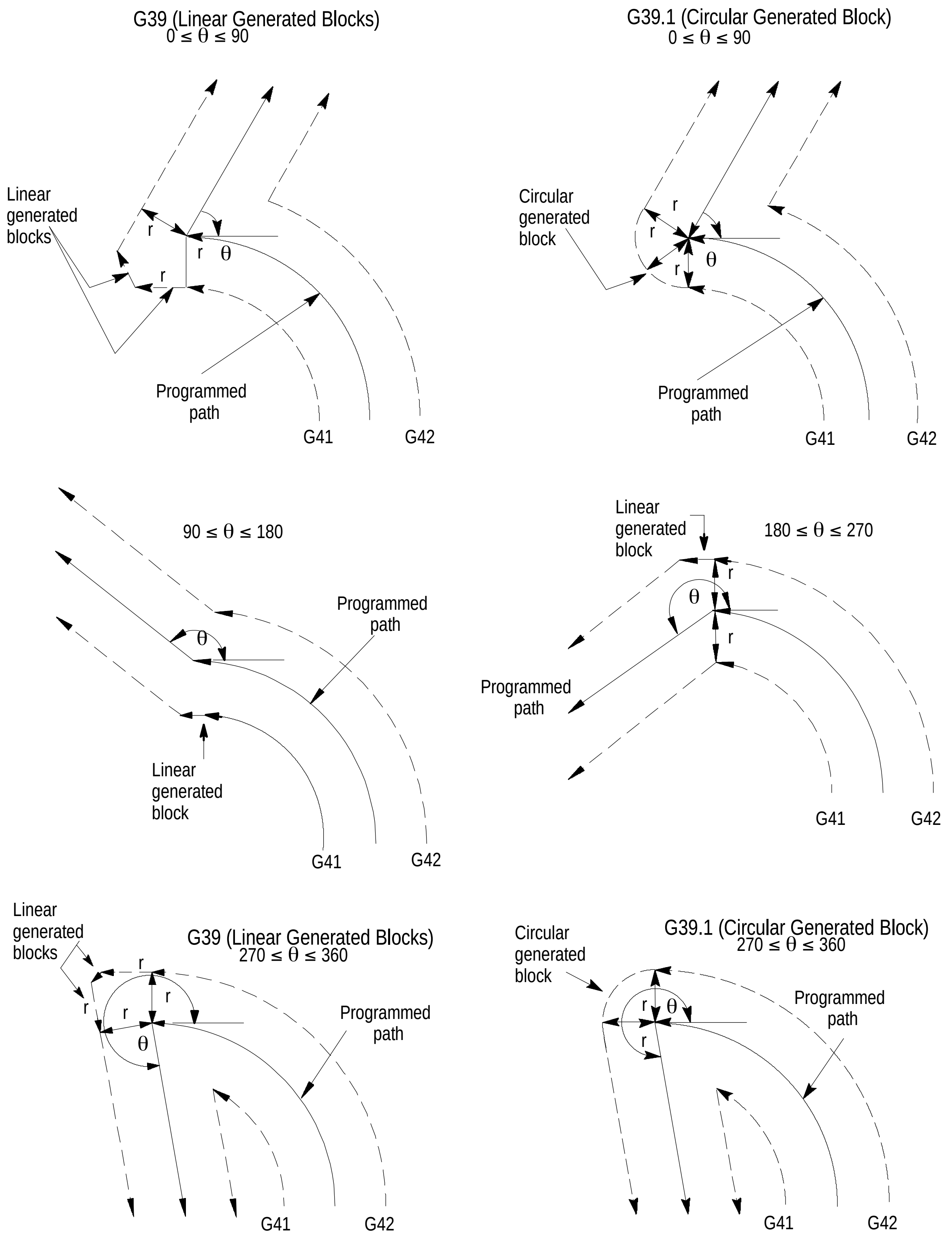
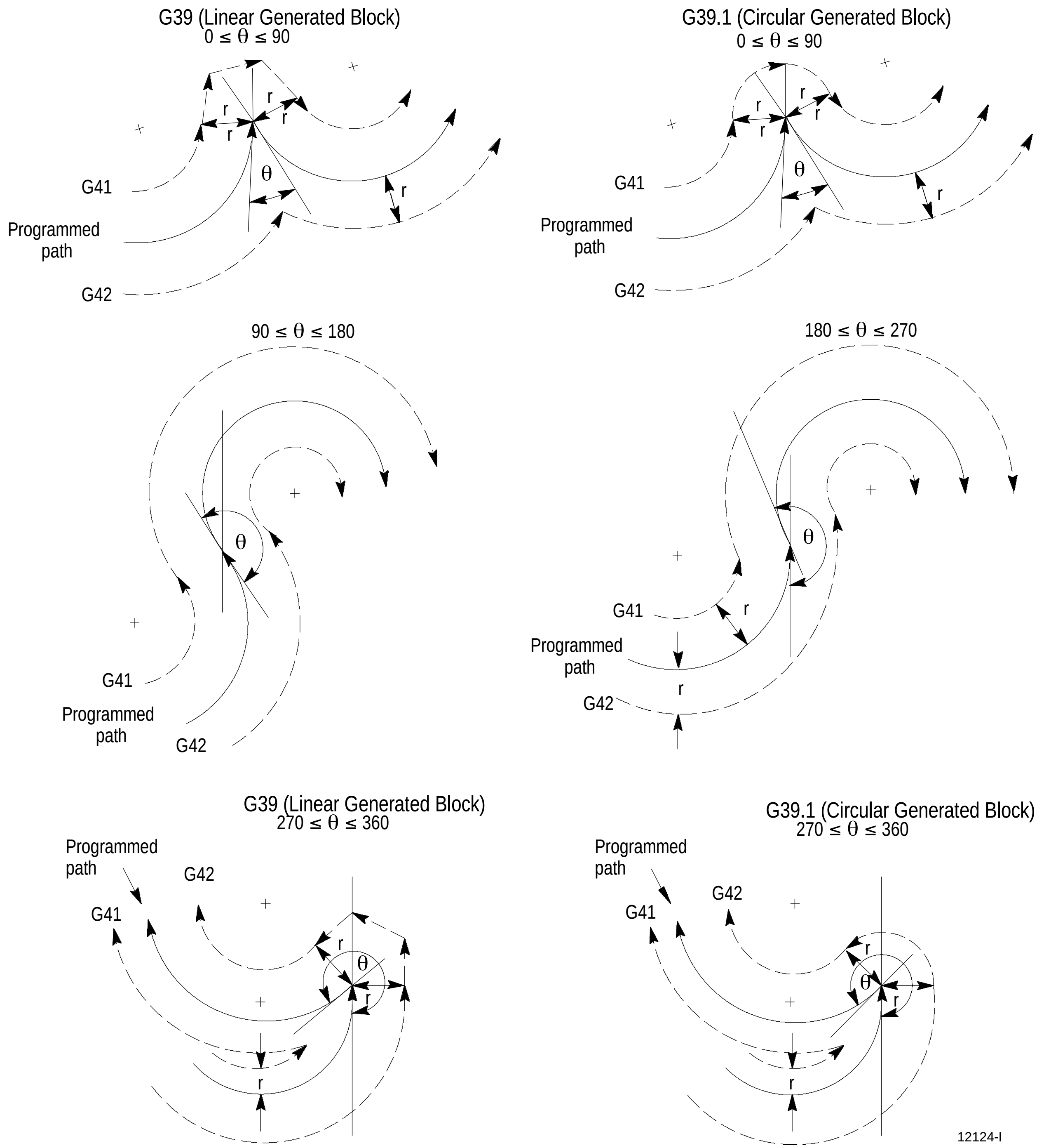


Figure 15.33
Dresser/Wheel Radius Compensation paths Arc-to-Arc



15.7 Special Compensation Cases

The following subsections cover possible paths that can be generated when programming one of these during radius compensation:

- Changing radius compensation direction (cross-over dresser/wheel paths)
- Exceeding the allowable number of consecutive, non-motion blocks during radius compensation
- Corner movement following a generated block
- Changing dresser/wheel radius during radius compensation
- Effect on radius compensation when interrupting a program to execute either a MDI program or a manual move
- Changing or offsetting current work coordinate system during radius compensation
- Moving to and from machine home and secondary machine home

15.7.1 Changing Compensation Direction

This section describes the resulting path when you program a change in compensation direction (left or right). This can result in the dresser/wheel crossing over the programmed path as compensation changes from left to right or right to left.

Linear Path-to-Linear Path

Example 15.9 through Example 15.12 show the path taken when you change radius compensation from G41 to G42 during the execution of two linear program moves.

The control generates two points when changing radius compensation direction:

- point 1

  This is the final dresser/wheel position before compensation direction is changed (at right angles to the end-point of the programmed path offset by one dresser/wheel radius).

- point 2

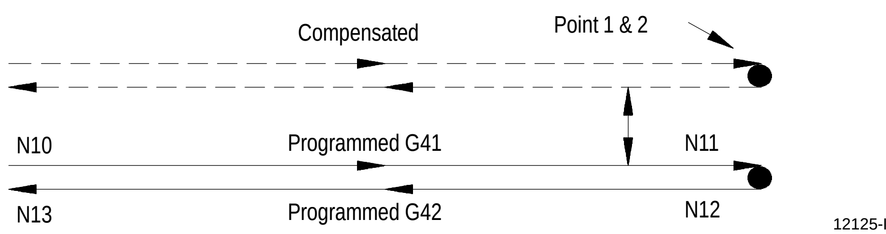
  This is the desired dresser/wheel position for the start of the first block using the changed compensation direction (at right angles to the start-point of the motion block that changes compensation direction and offset by the dresser/wheel radius).

The control generates the motion block that connects point 1 to point 2 as shown in the examples below:

Example 15.9
Linear-to-Linear Change in Dresser/Wheel Radius Compensation Direction (Reversing Path)

```
N10 Z10.G41;
N11 Z20.;
N12 Z10.G42;
N13 Z0.;
```

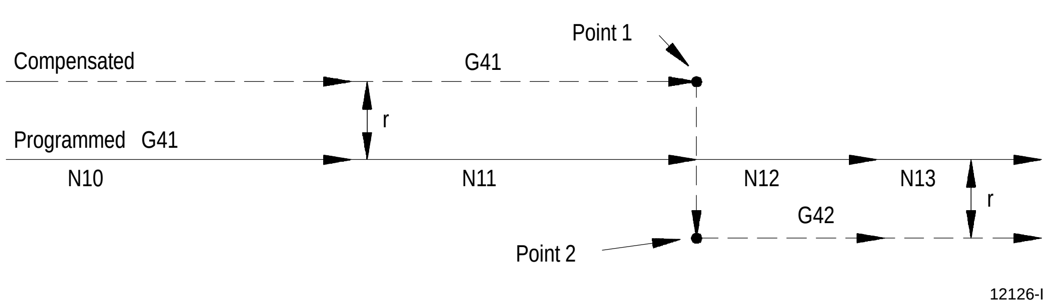
Figure 15.34
Results of Example 15.9



Example 15.10
Linear-to-Linear Change in Dresser/Wheel Radius Compensation Direction (Continuing Path)

```
N10 Z10.G41;
N11 Z20.;
N12 Z30.G42;
N13 Z35.;
```

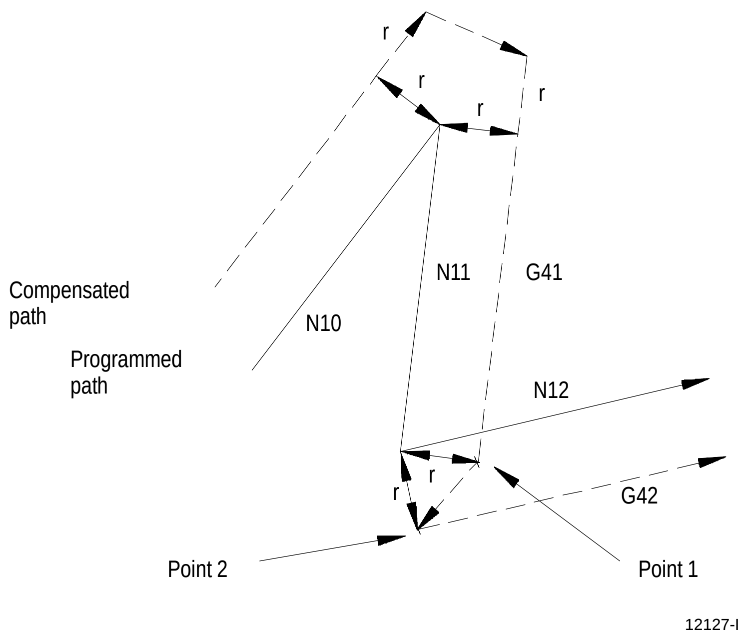
Figure 15.35
Results of Example 15.10



Example 15.11
Linear-to-Linear Change in Dresser/Wheel Radius Compensation Direction (With Generated Blocks)

```
N10 X15.Z10.G41;
N11 X-5.Z8.;
N12 X0.Z35.G42;
```

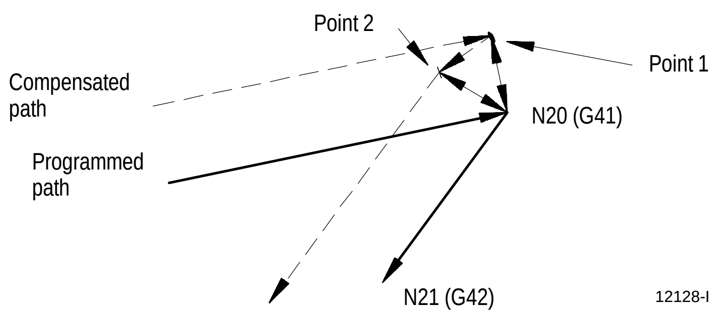
Figure 15.36
Results of Example 15.11



Example 15.12
Linear-to-Linear Change in Dresser/Wheel Radius Compensation Direction (No Generated Blocks)

```
N20 X5Z10.G41;
N21 X-5.Z7.G42;
```

Figure 15.37
Results of Example 15.12

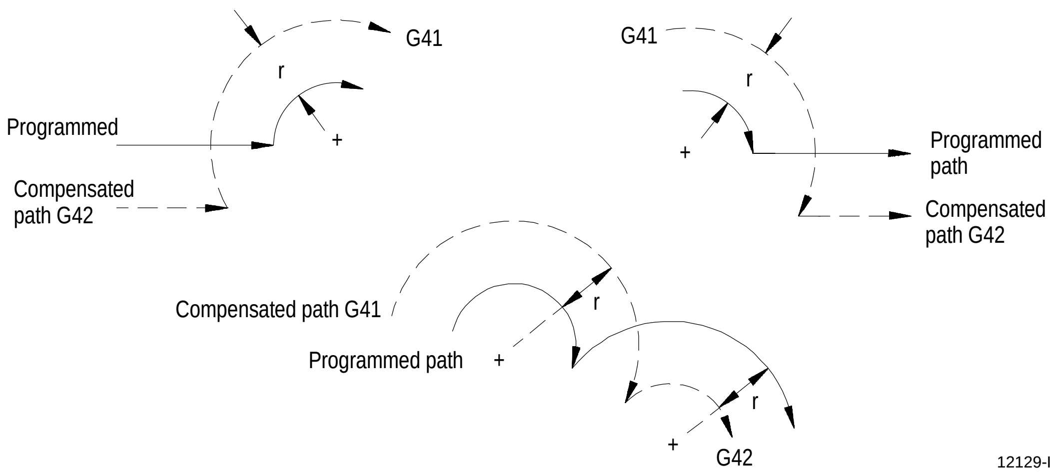


For one of these cases that changes the radius compensation direction, the control attempts to find an intersection of the actual compensated paths:

Linear-to-Circular, Circular-to-Linear, or Circular-to-Circular Paths

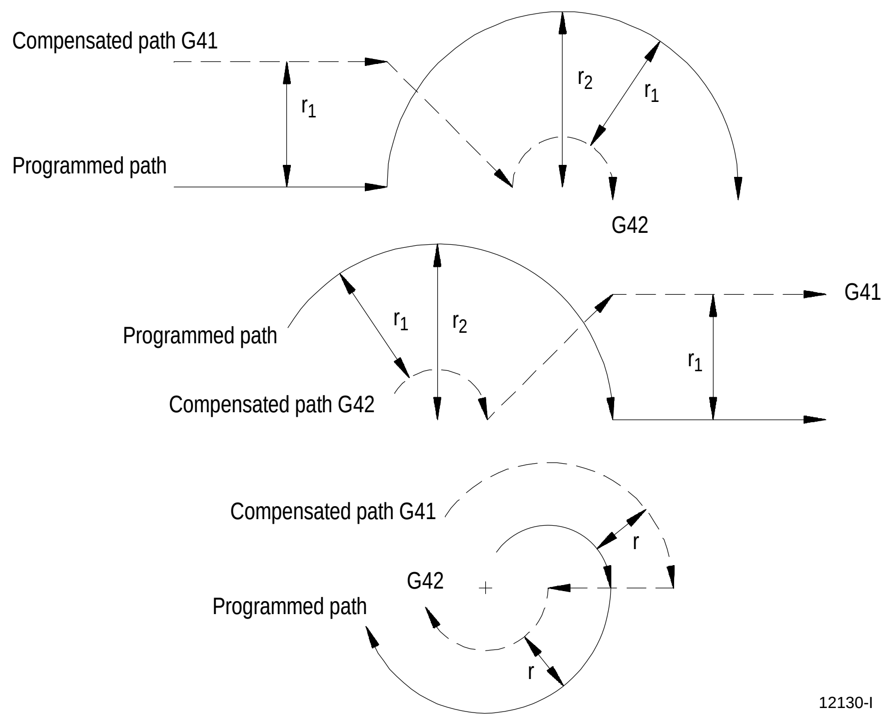
If the control finds an intersection, it modifies the end-point of the original compensated path and the start-point of the new compensated path to equal that intersection (see Figure 15.38).

Figure 15.38
Change in Compensation with Actual Path Intersection



If no intersections of the actual path exist, the compensated path is the same as if a linear-to-linear intersection had taken place (see Figure 15.39).

Figure 15.39
Change in Compensation With No Possible Path Intersections



15.7.2 Too Many Non-Motion Blocks

The control always looks ahead to the next motion block to determine the actual path taken for a motion block in radius 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. See documentation prepared by your system installer for the allowable number of non-motion blocks allowed in a specific system.

Important: We define non-motion block as 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 skip feature (/) described on page 7-2 are also counted as non-motion blocks in radius compensation regardless of the content of the skipped block.

When scanning ahead, If the control does not find a motion block before the number of non-motion blocks has been exceeded, it does not generate the normal radius compensation move. Instead the control sets up the compensation move with an end-point one-dresser/wheel radius away from and at right angles to the programmed end-point.

In many cases, this can cause unwanted over-cutting of a work piece. Figure 15.40 and Figure 15.41 are example paths of programmed motion blocks followed by too many non-motion blocks before the next move was made.

Figure 15.40
Too Many Non-Motion Blocks Following a Linear Move

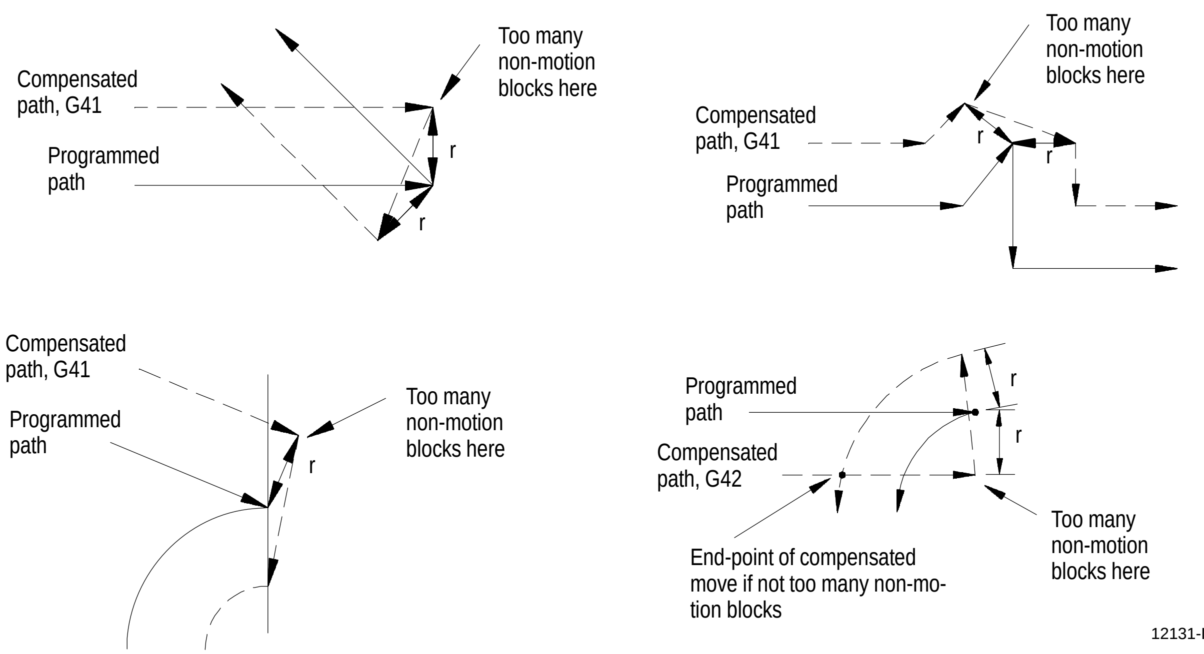
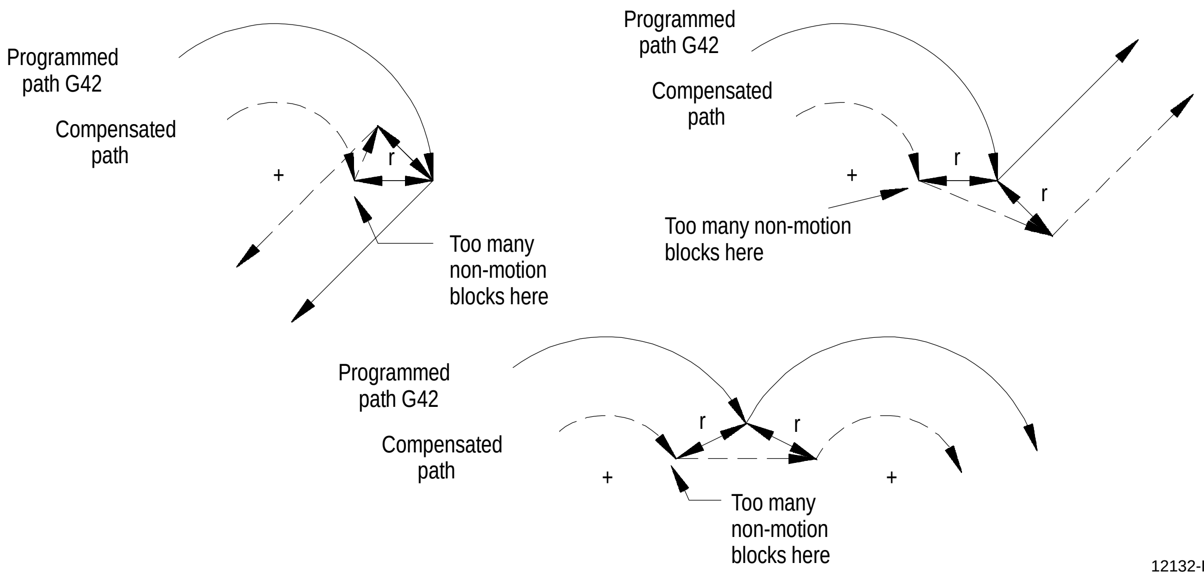


Figure 15.41
Too Many Non-Motion Blocks Following a Circular Move



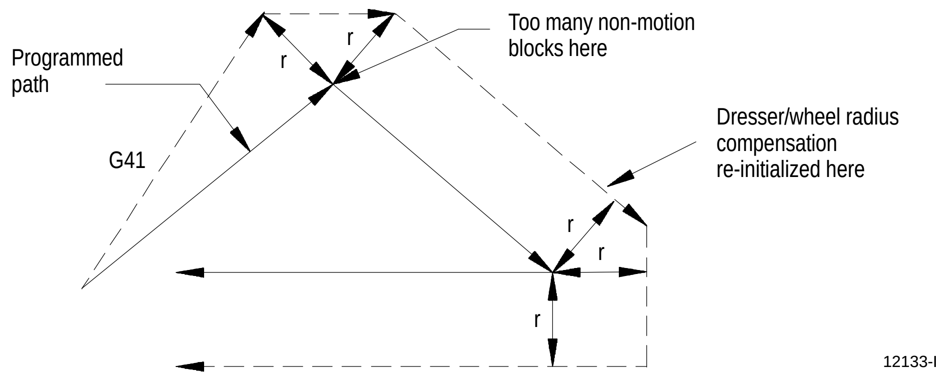
For example, assume that your system installer has determined that you can perform only two non-motion blocks during compensation. If more than two blocks following the entry move do not contain axis motion in the current plane, the control re-performs the entry move at the next block containing axis motion in the current plane.

Example 15.13
Too Many Non-Motion Blocks After Entry Block

```
Assume current plane to be the ZX plane.
```

`N1X0.Z0.;`	
`N2G41T0100X1.Z1.;`	Entry move dresser/wheel radius compensation left.
`N3;`	No axis motion in current plane.
`N4...;`	No axis motion in current plane.
`N5X4Z-4;`	New entry move dresser/wheel radius compensation left.

Figure 15.42
Results of Example 15.13

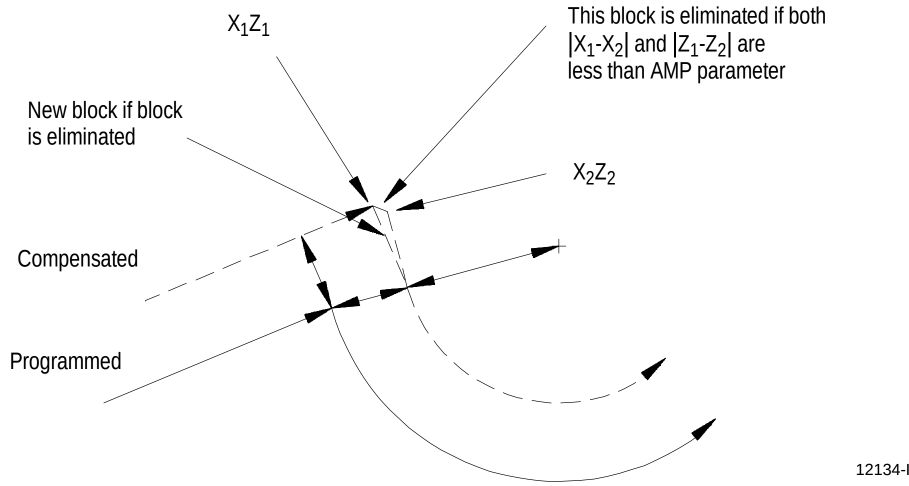


15.7.3 Corner Movement After Generated Blocks

Frequently the control needs to generate motion blocks to position the dresser/wheel in the proper alignment for a following compensated move. These blocks are generated to make sure that the dresser/wheel remains at least one radius of the dresser/wheel away from the programmed path at all times.

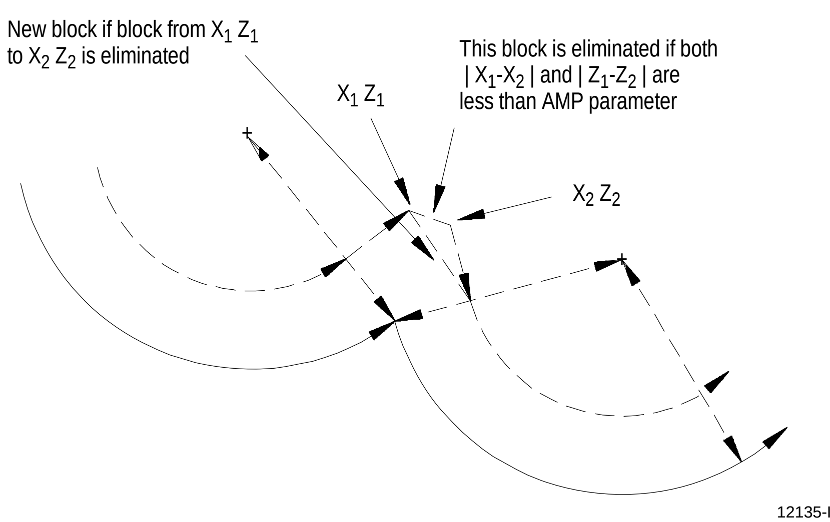
When the control generates two motion blocks, the length of the first generated block is checked against a minimum allowable length as determined in AMP by your system installer. The coordinate values for the current axes in the compensation plane are compared to the minimum allowed value. If both are less than the allowed value, the control does not execute the first generated block. The path of the second generated block is altered to position the dresser/wheel along a linear path to the original end-point of the second generated block. See Figure 15.43.

Figure 15.43
Compensation Corner Movement for Two Generated Blocks



When the control generates 3 motion blocks, the length of the second generated block is checked against a minimum allowable length, determined in AMP by your 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, the control does not execute the second generated block. The path of the third generated block is altered to position the dresser/wheel along a linear path to the original end-point of the third generated block. See Figure 15.44.

Figure 15.44
Compensation Corner Movement for Three Generated Blocks



15.7.4 Changing Dresser/Wheel Radius During Compensation

If a dresser/wheel becomes excessively worn, broken, or for any other reason requires the changing of the programmed dresser/wheel radius, radius compensation should be canceled and re-initialized after the dresser/wheel has been changed. See page 3-4 on changing the dresser/wheel radius offset value and page 13-1 on changing the active dresser/wheel offset number.

Figure 15.45 through Figure 15.47 are representations of the paths taken after programming a change in dresser/wheel radius. Assume in these figures that the programmed change to the dresser/wheel radius is entered in block N11 that also contains the motion as described in the figures.

The path taken when changing dresser/wheel radius depends 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 dresser/wheel paths.

Figure 15.45 describes the path when the programmed moves are linear-to-linear.

Example 15.14
Linear-to-Linear Change in Dresser/Wheel Radius

When the control generates blocks	When the control does not generate blocks
N10 X10.Z5.G1T0100;	N10 X10.Z10.G1T0100;
N11 X-5.Z3.T0200;	N11 Z20.T0200;
N12 Z20.G42;	N12 X0.Z30.;

Figure 15.45
Linear-to-Linear Change in Dresser/Wheel Radius During Compensation

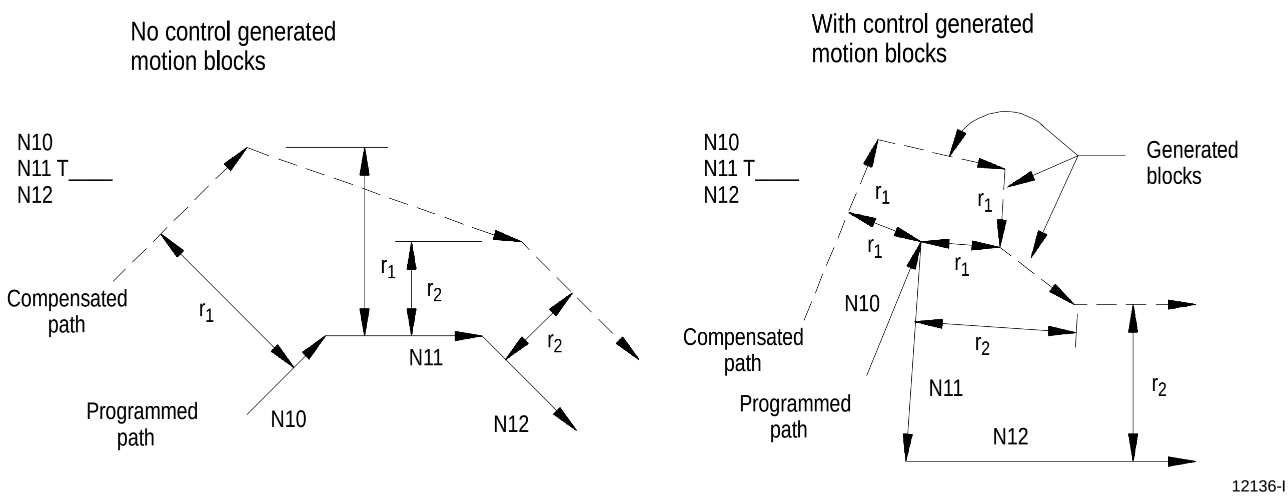


Figure 15.46 describes the path when the programmed moves are linear-to-circular.

Figure 15.46
Linear-to-Circular Change in Dresser/Wheel Radius During Compensation

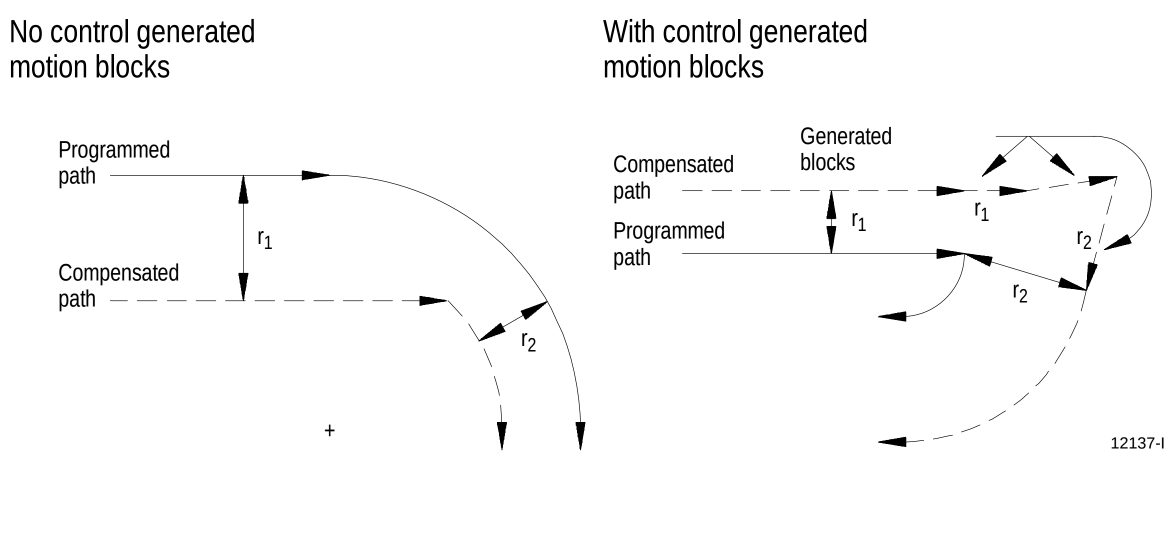
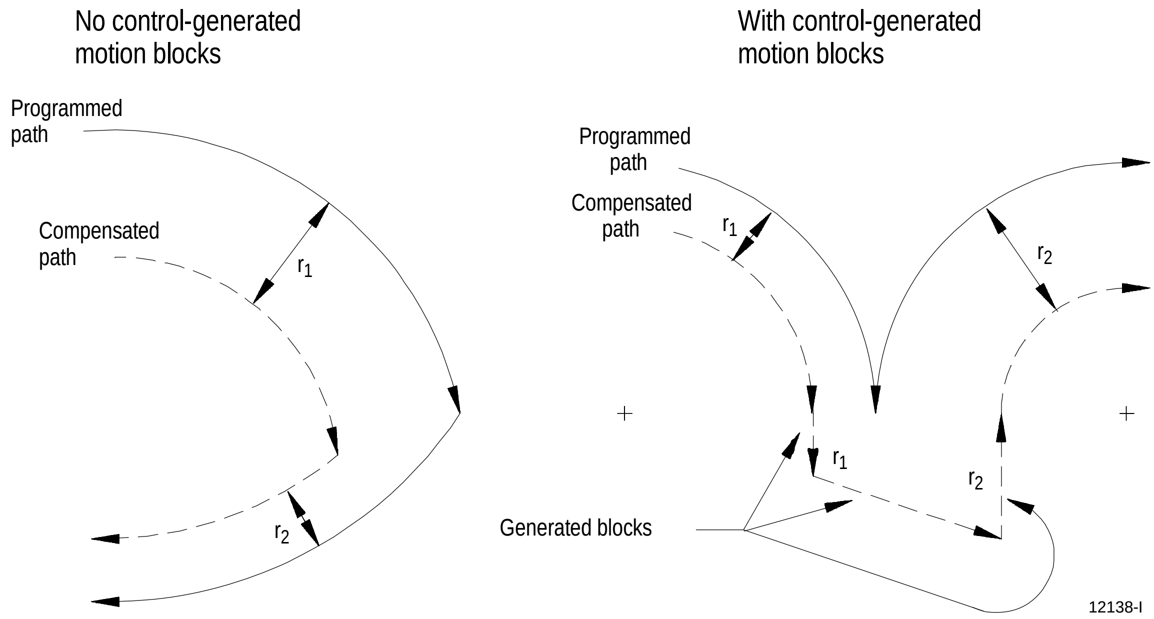


Figure 15.47 describes the path when the programmed moves are circular-to-circular.

Figure 15.47
Circular to Circular Change in Dresser/Wheel Radius During Compensation



Change in Dresser/Wheel Radius During Jog Retract

This section describes the change in the dresser/wheel radius during a jog retract operation. This is a typical operation because the jog retract feature is often used when a dresser/wheel becomes worn or is broken. If it is necessary to replace the dresser/wheel with one of a slightly different diameter, radius compensation can adjust to the new diameter.

Typically when you perform the jog retract operation, the dresser/wheel is jogged away from the workpiece and then replaced. After it is replaced, you have to activate a different dresser/wheel radius offset value. This is done in either of two ways:

- The new offset number is activated by programming a new T 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 described in chapter 4

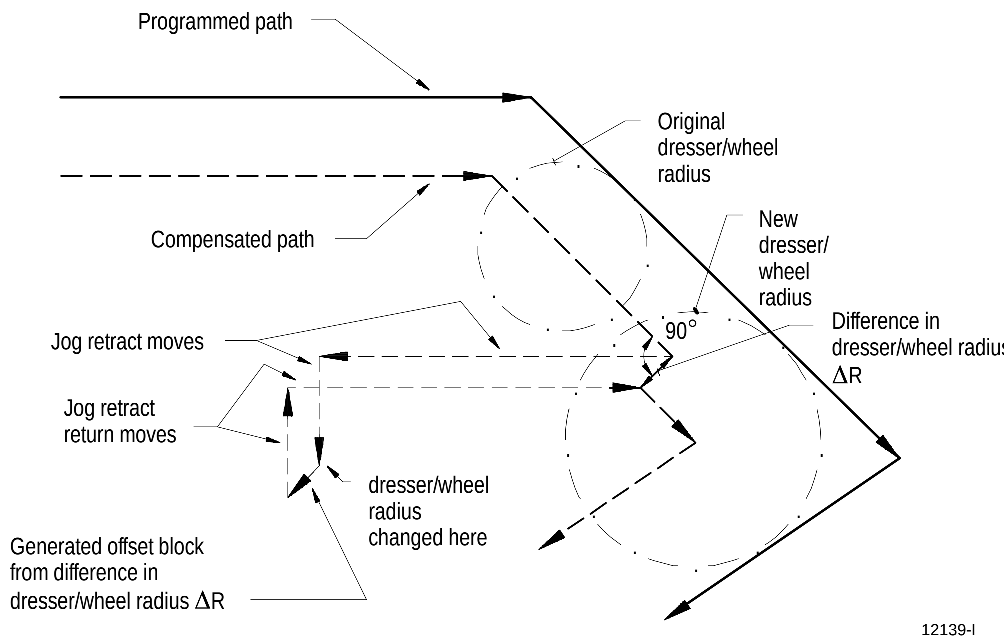
Regardless of how you activate the new offset, radius compensation can compensate for this new diameter by modifying the saved jogged path. This path is modified so that the new dresser/wheel cuts the same part as the old dresser/wheel. The absolute position of the machine is therefore different on the return path from what it was when jogging away from the part.

The control adjusts this jogged path when you press the <CYCLE STOP> button to return from the jog retract. Once you press the <CYCLE STOP> button, the control generates a move that offsets the current position by the necessary distance. This distance is determined as being the necessary distance the dresser/wheel 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 dresser/wheel radius.

ATTENTION: Make sure that this offset path does not cause any collisions with the part or the machine fixtures. The position of the dresser/wheel when the change in jog retract is made should be a safe distance from the part and machine fixtures.

Figure 15.48 shows an example of a typical change in dresser/wheel radius during jog retract with radius compensation active:

Figure 15.48
Change in Dresser/Wheel Radius During a Jog Retract



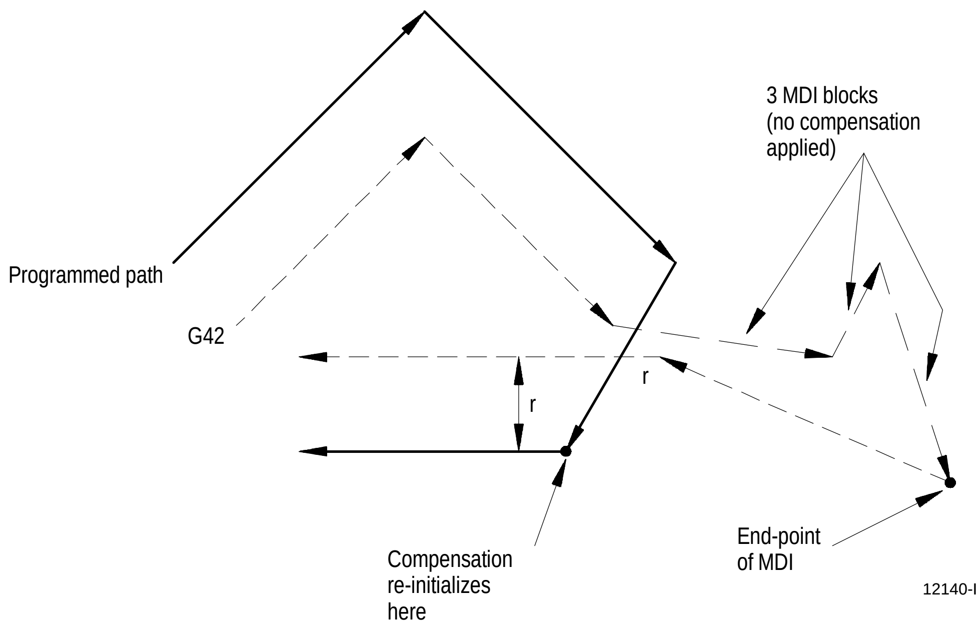
15.7.5 MDI or Manual Motion During Dresser/Wheel Radius Compensation

If exiting automatic mode, and if either an MDI motion block is executed or a manual jog motion is made, the radius compensation feature, if active, is re-initialized when the next motion block is executed in automatic mode. The compensation feature compensates the dresser/wheel one radius perpendicular to the 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 dresser/wheel radius compensation feature is not available for any motion blocks that are programmed in MDI mode. The dresser/wheel radius compensation mode can be altered by programming either G41, G42, or G40, or the dresser/wheel radius can be changed in an MDI program. However, none of the paths executed in MDI are compensated. Any changes made to dresser/wheel radius compensation are not applied until the next block executed in automatic mode.

Figure 15.49 is an example of the possible path taken when interrupting automatic operation during radius compensation to execute MDI motion blocks. The same path would apply if interrupting radius compensation to perform a manual jog move.

Figure 15.49
Dresser/Wheel Radius Compensation Interrupted with MDI Blocks



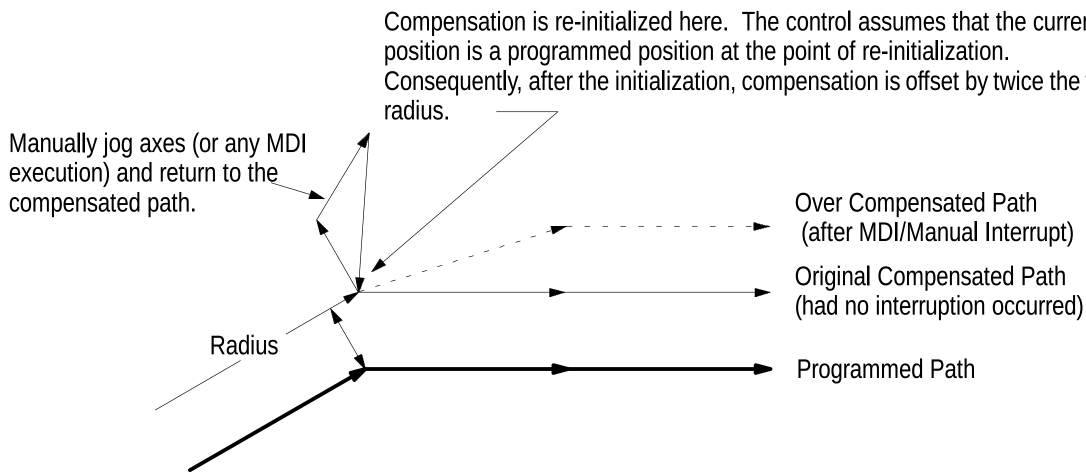
Important: If during cutter compensation, you switch out of automatic mode and either:

- generate axis motion in manual mode on an axis in the compensation plane, or
- execute any block in MDI mode,

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 dresser/wheel radius. Figure 15.50 illustrates these conditions.

Figure 15.50
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 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 compensation. You can avoid this by using the jog retract feature instead of manual or MDI when you need to interrupt compensation.

15.7.6 Moving To/From Machine Home

We recommend that you use a G40 command to cancel the dresser/wheel radius compensation before the execution of a return-to or a return-from the machine home or the secondary machine home. This refers to the operations performed when the control executes either the G28, G29, or G30 commands as described in chapter 12.

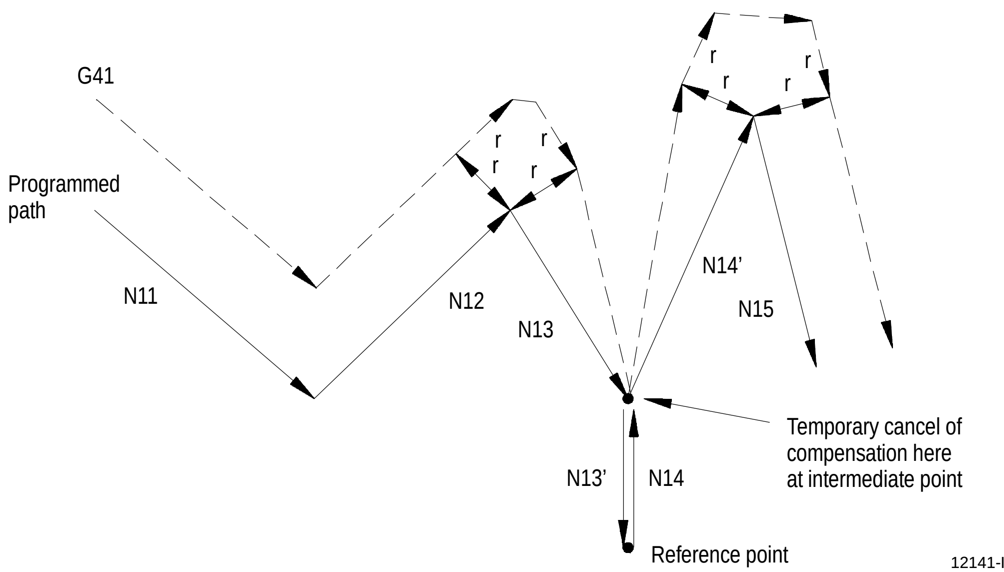
If compensation is not canceled using a G40 command, the control temporarily cancels compensation for the return-to machine home or secondary machine home operations. This is done by using the move to the intermediate point, as designated when the operation was performed, as an exit move for compensation.

Important: You should always program an intermediate point for a return-to home operation if dresser/wheel radius 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 can cause undesired overcutting of the part.

If compensation was not canceled using a G40 command before returning to machine or secondary home points, the control automatically re-initializes dresser/wheel radius compensation for the return from machine or secondary home points. This is done by using the move to the intermediate point, as designated when the operation was performed, as an entry move for compensation.

Figure 15.51 gives an example of either a G28 or G30 block followed by a G29 block.

Figure 15.51
Dresser/Wheel Radius Compensation During G28, G30, and G29 Blocks



15.7.7 Changing or Offsetting Work Coordinate System in Dresser/Wheel Radius Compensation

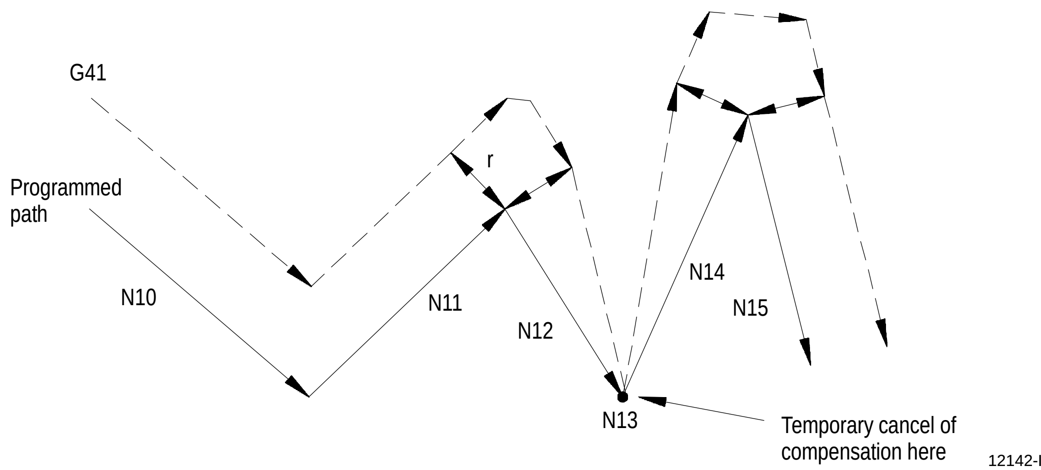
We recommend that you cancel dresser/wheel radius compensation using a G40 command before you make any modifications to the current work coordinate system, including any offsets or any change of the coordinate system (G54-G59.3).

If compensation is not canceled 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.

If compensation was not canceled using a G40 command before a change in the work coordinate system was performed, the control automatically re-initializes dresser/wheel radius 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 a entry move for compensation.

Figure 15.52 gives an example of programming a G92; however, this would apply to any change in the work coordinate system.

Figure 15.52
Dresser/Wheel Radius Compensation During G92 Offset to Work Coordinate System



15.7.8 Block Look-Ahead

During normal program execution, the control is constantly scanning ahead several blocks to set up the necessary motions to correctly execute the current block. This is called Block Look-Ahead.

The control has 21 set-up buffers. Different features require some of these set-up buffers. One is always used for the currently executing block. Dresser/wheel radius compensation requires at least 3 of these buffers. The control uses any remaining set-up buffers for block look-ahead, with one buffer used for each block.

At times (especially during dresser/wheel radius compensation) the control cannot 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 re-traceable block. The number of re-traceable blocks is set in AMP by your system installer (a maximum of 15 is possible).

If necessary, the control decreases the number of available re-traceable 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 has to eliminate some of the block retrace blocks.

We recommend that you use as few set-up buffers as possible for block retrace. The larger the number of look-ahead blocks that the control has available to set up future part program motion requests, the more efficiently the control executes programs. We recommend that the number of set-up buffers available to the block retrace feature be kept as low as possible.

15.8 Error Detection

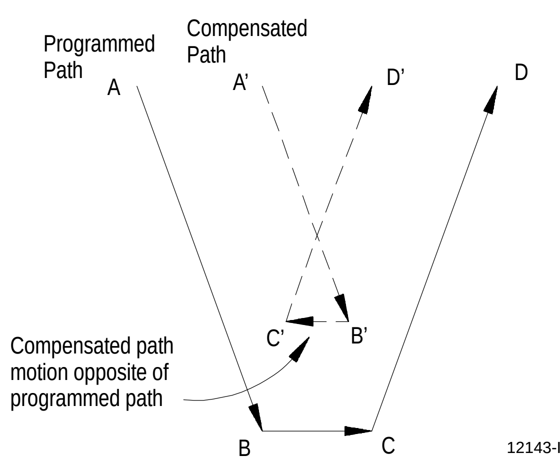
Error detection for dresser/wheel radius compensation blocks can be separated into three categories:

- Backwards motion detection
- Circular departure too small
- Interference

Backwards Motion Detection

The compensated path is parallel to but in the opposite direction of the programmed path.

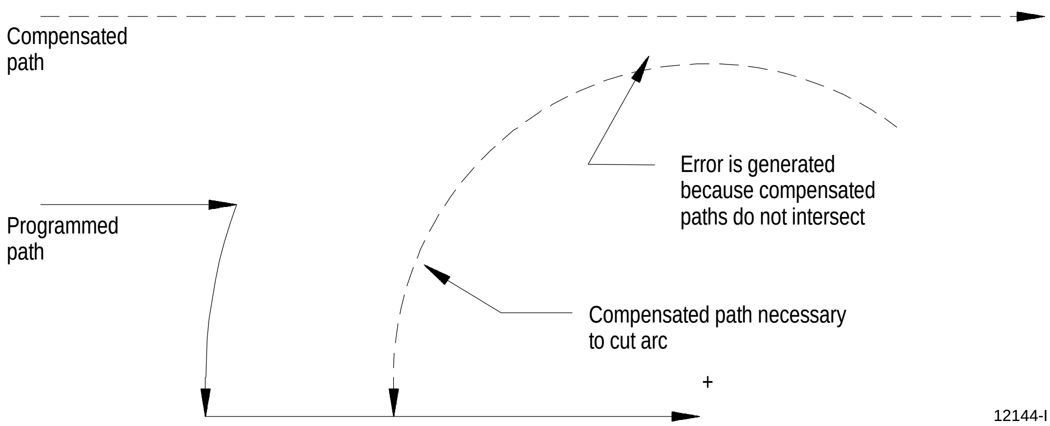
Figure 15.53
Typical Backwards Motion Error



Circular Departure Too Small

No intersection can be generated between two consecutive compensated paths.

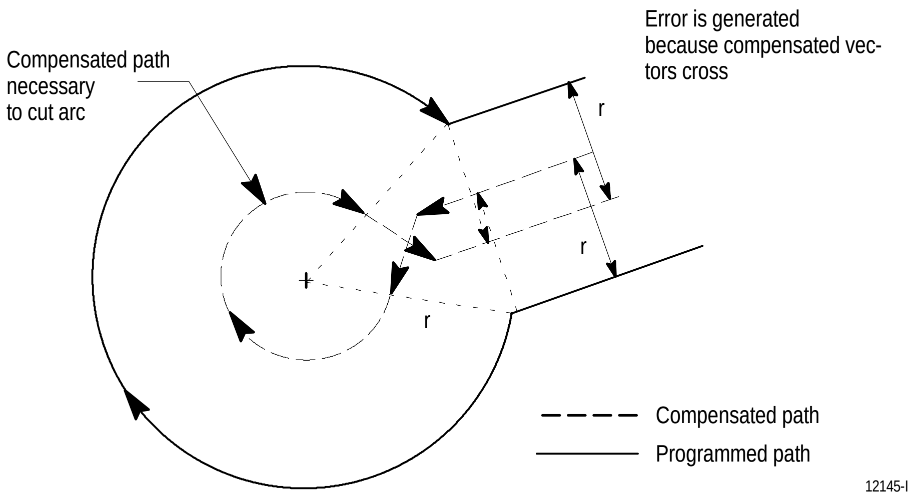
Figure 15.54
Typical Circular Departure Error



Interference

This error occurs when compensation vectors intersect. Normally when this intersection occurs, the control generates a backwards motion error; however, a few special cases exist that are caught only by interference error detection.

Figure 15.55
Typical Interference Error



Disabling Error Detection

You can disable all of the above error detection (with the exception of circular departure too small cases) for a specific block or portion of a part program. To disable the error detection for a specific block, your system installer must have defined an M-code in AMP. By programming this M-code in a block, you can disable all error detection for dresser/wheel radius compensation. Error detection is disabled until another M-code defined in AMP to re-enable error detection is programmed in a block.

Important: You cannot disable circular departure too small cases. It is not possible for the control to execute a compensated path when this error occurs.

The default condition is error detection enabled. Default values for these M-codes are:

```
M800 - disables error detection

M801 - enables error detection
```

See documentation prepared by your system installer for the M-codes used on a specific system.

END OF CHAPTER

Chapter 16

Surface Grinding Fixed Cycles

16.0 Chapter Overview

This chapter describes the surface grinding cycles available with the control. You can use these cycles to program axis motions to perform common grinding operations. Topics include:

Figure 16.1 shows the basic motions of the five surface grinding cycles described in this chapter. This figure shows only reciprocating and crossover motions as they would appear over the part surface. Plunge motion is omitted for clarity.

Figure 16.1
Surface Grinding Cycles

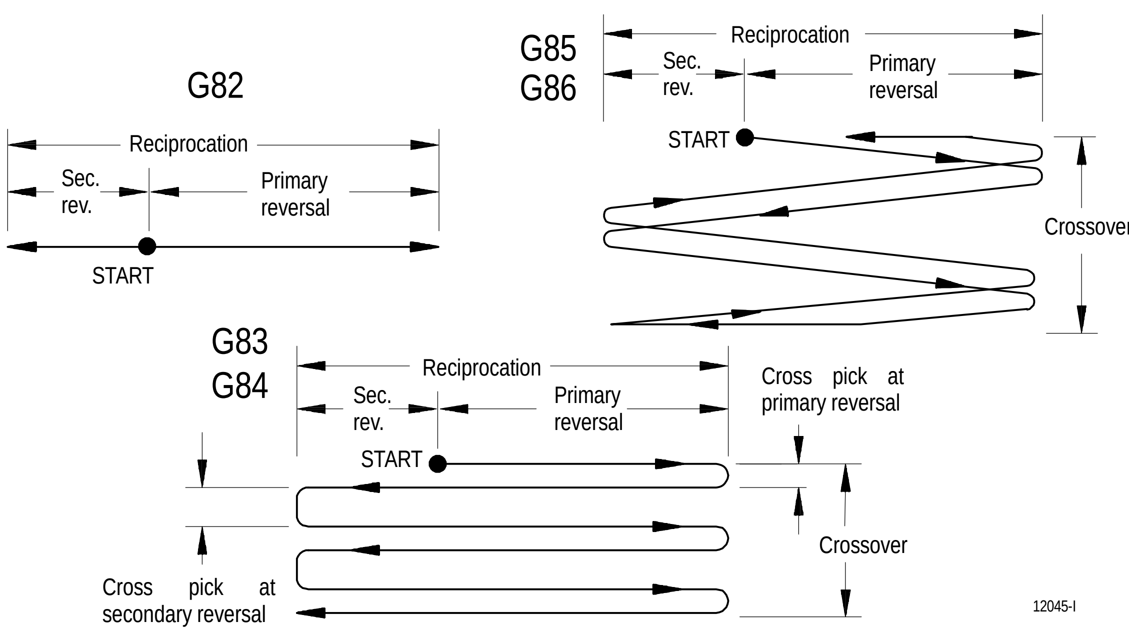
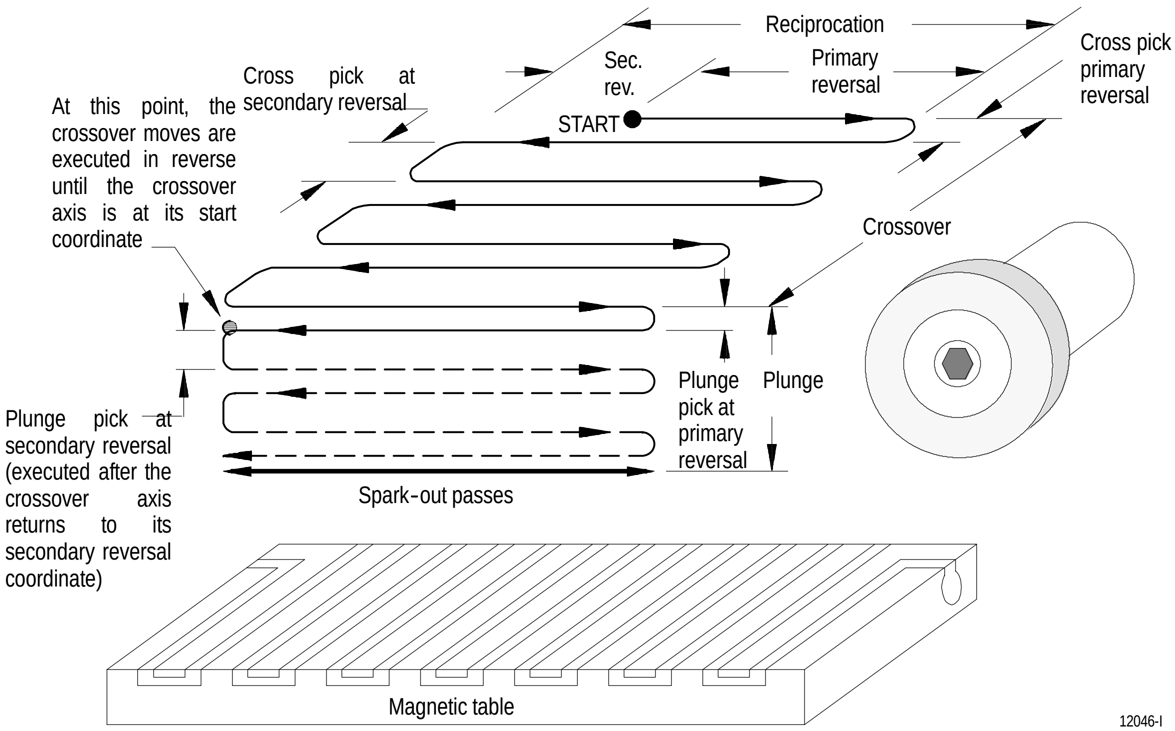


Figure 16.2 illustrates the reciprocation, plunge, and crossover motions of a typical surface grinding cycle (G83 Incremental Plane grinding in this case) .

Figure 16.2
Reciprocation, Crossover, and Plunge Motions



16.1 Surface Grinding Considerations

Modality and Programming

These surface grinding cycles are modal. Once programmed, the cycle is executed in each subsequent block that contains the appropriate parameters and parameter values. The G codes corresponding to these cycles do not have to be programmed in each block.

Typically crossover and plunge axis motions are coordinated with the motion of a reciprocating axis.

You can access Grinder Prompts through the Quick-view screens (described in chapter 5) to simplify programming.

Planes

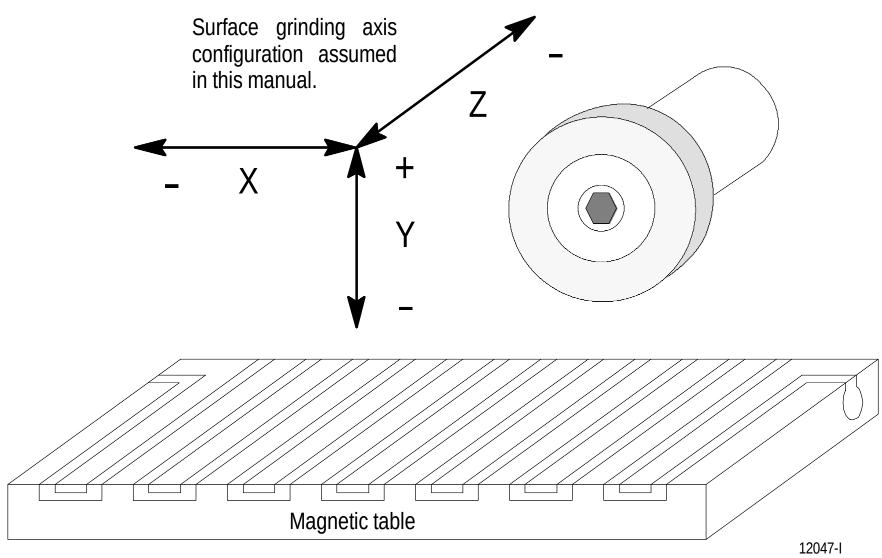
The operation of the surface grinding cycles is very dependent on plane selection. This chapter makes the following assumptions regarding plane configuration for the control. Your axis names and designations may be different. See the literature provided by your system installer.

PLANE	1st Axis	2nd Axis	1st Integrand	2nd Integrand	1st Parallel	2nd Parallel
G17	X	Y	I	J	U	V
G18	Z	X	K	I	W	U
G19	Y	Z	J	K	V	W

Plane selection must be made prior to executing a block that starts the reciprocation axis or a block that starts any of the grinding fixed cycles.

The drawings and descriptions in this chapter assume an X-Y-Z cartesian configuration for the axes on a surface grinding machine. Your machine configuration may be different. The illustrations in this chapter generally assume that the G19 (YZ) plane is active and the axes are configured as shown in Figure 16.3.

Figure 16.3
Typical Axis Configuration



Reciprocation

ATTENTION: Reciprocation differs from conventional axis motion in that the reciprocating axis has no final destination. Once axis reciprocation begins, it continues through program block execution until stopped by a G80 or an end of program (M02, M30, M99). This means pressing single block or cycle stop does not necessarily stop the reciprocating axis.

Important: The motion of the reciprocating axis and, consequently, the grinding cycles can be significantly altered through the control's PAL program. See your PAL reference manual and/or the literature prepared by your system installer.

Important: The reciprocating axis cannot be mirrored (G51.1) and cannot be rotated (G68) before or during reciprocation.

The reciprocating axis is the axis that sweeps back and forth across the part surface (or under the grinding wheel, depending on your machine configuration). Programming a G81 causes reciprocation mode to become active. The reciprocation axis is whichever axis is programmed in the G81 block. Only one axis name can be programmed in a G81 block.

For the surface grinding cycles (G82 through G86) the reciprocation axis is always an axis that is not in the active plane. For example, if you have G19 (YZ) active and configured as shown above, then typically the X axis would be programmed as the reciprocating axis for the surface grinding cycles. Any other axis not in the active plane (U, V, or W for example) could instead be programmed as the reciprocating axis.

The reciprocating motion is divided into a primary reversal and a secondary reversal. The reversal points of the primary and secondary reciprocation moves are defined by the numeric values associated with the reciprocating axis name and the reciprocating axis integrand. Reversal points can be an incremental distance from the start point (G91 active), or an absolute position (G90 active).

Axis motion can be commanded while reciprocation mode is active. If the axis to be moved is independent of the reciprocation, then motion begins as soon as the block is executed.

Certain commands or commanded motions depend on reciprocation and are delayed until the reciprocating axis reaches the secondary reversal point. Examples of such commands include:

- coordinate changes requested for the reciprocating axis
- certain fixed cycle operations as described in this chapter
- a G80 is executed to cancel reciprocation
- an M99 is executed at the end of a main program

If an incremental plunge or crossover is to be made (G82, G83 or G84), it begins as the reciprocating axis decelerates to a reversal point. If a continuous crossover is to begin (G85 or G86), it starts immediately regardless of the location of the reciprocating axis.

Programming an incremental or absolute move to the reciprocating axis in any program block while reciprocation is active, can change the coordinates of the primary and secondary reversal points. In the case of programming moves to the reciprocating axis while in incremental mode, the incremental change is made relative to the secondary reversal point, as illustrated in this example:

Program Block	Prim. Rev. Point	Sec. Rev. Point	Comments
G90G80F100X10.0;	none	none	no reciprocation
G81G91X5.0I-10.0E100;	X axis, 15.0	X axis, 5.0	from start point of X = 10
X5.0I-8.0;	X axis, 10.0	X axis, 2.0	from sec. rev. point of X = 5
X-20.0I5.0;	X axis, -18.0	X axis, -13.0	from sec. rev. point of X = 2

The preceding example applies not only to the G81 shown, but to the reciprocating axis in any grinding fixed cycles.

If already active, reciprocation motion does not stop or pause when a grinding cycle block or a block changing the reciprocation reversal points is executed. Instead, changes to the reciprocation motion take place at the secondary reversal point.

A plane change (G17, G18, G19) can be requested while the reciprocation axis is in motion. If the next grinding cycle block to be executed after the plane change requires a different reciprocation axis, then there must be valid reciprocation axis data included with that grinding cycle block. If not, the error message “RECIPROCATION AXIS IN WRONG PLANE” appears and block execution stops. Reciprocation does not stop.

Reciprocation stops when an emergency stop condition occurs. No motion occurs when the emergency stop is reset. If your control's AMP is configured such that the control is not reset after an E-STOP reset, then typically reciprocation resumes with the next cycle start. This lets part programs resume immediately without having to first restart reciprocation. For details, see the documentation prepared by your system installer.

M00 blocks do not affect reciprocation.

The feedrate for the reciprocation motion is determined by the E or F words as described below in order of priority:

- the E word in the reciprocation block
- the E word in a previous reciprocation block executed since the last end of program (M02, M30, M99) or control reset
- the F word in the reciprocation block
- the F word in a previous block executed since the last end of program (M02, M30, M99) or control reset

If there is no active E or F word feedrate, an error is generated as soon as the reciprocation motion is commanded. Your PAL program ultimately determines reciprocation axis speed and motion. See the documentation prepared by your system installer.

Acceleration and deceleration for the reciprocating axis is always linear, regardless of the ACC/DEC type selected for that axis in AMP.

You can determine (through PAL) whether block transitions are made after the reciprocation axis reaches its in-position band (positioning mode) or whether the control executes blocks regardless of the in-position band (non-positioning mode). See the documentation prepared by your system installer.

Crossover

A crossover consists of a series of reciprocation and cross pick moves that cover the desired surface area. You can specify two cross pick increments for the incremental surface grinding cycles (G83 and G84):

- the cross pick at the primary reversal of the reciprocating axis and
- the cross pick at the secondary reversal of the reciprocating axis.

The continuous surface grinding cycles (G85 and G86) use a continuous cross feed move that begins at the very start of the reciprocating motion.

The Plunge

Plunge refers to the axis motion towards the part surface. You can specify two plunge pick increments:

- the plunge pick at start and
- the plunge pick at crossover

Parallel Axes

You can use parallel axes in the surface grinding cycle blocks in place of the primary axes to which they are parallel. For example, you can program:

```
G19VZ;
G83X__I__V__J__Z__K__;
```

where the Y axis has been replaced with its parallel axis, V. The integrand for the Y axis, J, is used for the V axis. The active plane becomes the VZ plane where V is axis 1 and Z is axis 2 of the plane.

The cycle begins with a reciprocating move (X axis) followed by a cross pick move (Z axis) that begins as the X axis decelerates. After the entire crossover is completed, a plunge move (V axis) is made, again beginning as the X axis decelerates. See the description of the G83 cycle for details on cycle operation.

Block Displays while Executing Grinding Fixed Cycles

When executing part program blocks, the main screen of the control displays a “*” before a block it is executing and a “@” before a block it has completed. This is typical of any block executed on the control.

Be aware that when block execution involves the reciprocation axis, the block may appear to be handled differently, but actually is not. What frequently happens is that the “@” appears before a grinding fixed cycle block and a “*” appears before the block that follows it even though that block is being held in pre-block. It is being held while waiting for the reciprocation axis to reach its secondary reversal point.

Similarly, the G code display may show the G code of the following block as active when, again, that block is being held in pre-block waiting for the reciprocation axis to move to its secondary reversal point.

Cancel Grinding and Reciprocation

Use a G80 to cancel all surface grinding cycles. Programming a G80 cancels a G81, G82, G83, G84, G85, or G86. When executed, a G80 also stops the reciprocating axis.

Once reciprocating motion begins, it continues through program block execution until a G80 is executed. If there is no G80, reciprocation continues until an end of program (M02, M30, M99) occurs. An M99 in a subprogram simply returns program execution to the main program and does not cancel reciprocation.

When reciprocation or a grinding cycle is canceled by a G80 or an M99, the reciprocating axis continues to its secondary reversal point before the control considers the G80 or M99 block completed.

16.2 Surface Grinding Parameters

This section provides a description of the parameters that apply to each of the surface grinding cycles. For ease of explanation, we assume that the G19 (YZ) plane is active and the axes in that plane are configured as shown in the table in the previous section. If you are using a different plane or your axes were named differently, then your parameters can be different.

Also it is assumed that axis one (Y) of the active plane has been selected as the plunge axis (G83 and G85). If axis two (Z) of the active plane is selected as the plunge axis (G84 and G86), then the descriptions for parameters Z, K, and R should be swapped with the descriptions for parameters for Y, J, and Q.

X - reciprocating axis distance, primary reversal point.

If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the distance from the start point to the end of the primary reciprocating motion (the primary reversal point).

The start point for the reciprocating axis is the coordinate of the X axis prior to execution of the surface grinding cycle. If the X axis is already reciprocating when the surface grinding cycle block is executed, then the start point (when incremental mode is active) is the secondary reversal point. This was explained on page 16-4.

If in absolute mode (G90 active) then the value entered here is the X coordinate of the primary reversal point.

This parameter is “program modal,” meaning that it needs to be programmed only once in a part program. Any subsequent surface grinding blocks (G82 – G86) use a previously programmed value if a new value is not programmed. Programming an X word in any program block while reciprocation is active changes the coordinate of the primary reversal point.

I - reciprocating axis distance, secondary reversal point.

If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the secondary reciprocating motion. If in absolute mode (G90 active) then the value entered here is the X coordinate of the end point of the secondary reciprocating motion. If no value is entered for this, then the start point is used as the end point of the secondary reciprocating motion.

The start point for the reciprocating axis is the coordinate of the X axis prior to execution of the surface grinding cycle. If the X axis is already reciprocating when the surface grinding cycle block is executed, then the start point (when incremental mode is active) is the secondary reversal point. This was explained on page 16-4.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent surface grinding blocks (G82 – G86) use a previously programmed value if a new value is not programmed. Programming an I word in <u>any</u> program block while reciprocation is active changes the coordinate of the secondary reversal point.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the <u>last mode</u> programmed in the block, regardless of their position in the block.

Important: It is the programmer's responsibility to make sure that the crossover and reciprocating moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Crossover and plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

Z - crossover position.

If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the crossover motion. If in absolute mode (G90 active) then the value entered here is the Z coordinate of the end point of the crossover move.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent surface grinding blocks (G82 – G86) use a previously programmed value if a new value is not programmed.

K - cross pick amount at primary reversal.

The value entered here is an incremental value used to indicate the distance that the crossover axis moves as soon as the reciprocating axis begins decelerating from its primary reciprocating move.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent surface grinding blocks (G82 – G86) use a previously programmed value if a new value is not programmed.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the last mode programmed in the block, regardless of their position in the block.

R - cross pick amount at secondary reversal.

The value entered here is an incremental value used to indicate the distance that the crossover axis moves as soon as the reciprocating axis begins decelerating from its secondary reciprocating move. If no value is entered for this, then the K value is used.

This parameter is not "program modal." Any subsequent surface grinding blocks (G82 – G86) use the K value if an R value is not programmed.

Y - plunge depth.

If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the plunge motion. If in absolute mode (G90 active) then the value entered here is the Z coordinate of the end point of the plunge move.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent surface grinding blocks (G82 – G86) use a previously programmed value if a new value is not programmed.

J - plunge pick amount at start.

The value entered here is an incremental value used to indicate the distance that the plunge axis moves as soon as the reciprocating axis begins decelerating. This is the plunge distance moved after the crossover axis has returned to its start point coordinate.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent surface grinding blocks (G82 – G86) use a previously programmed value if a new value is not programmed.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the last mode programmed in the block, regardless of their position in the block.

Q - plunge pick amount at crossover.

The value entered here is an incremental value used to indicate the distance that the plunge axis moves as soon as the reciprocating axis begins decelerating. This is the plunge distance moved after the crossover axis reaches the crossover position defined by Z. If no value is entered for this, then the J value is used

This parameter is not "program modal." Any subsequent surface grinding blocks (G82 – G86) use the J value if a Q value is not programmed.

L - number of spark-out passes.

The number entered here (any integer from 0 to 999) indicates how many spark-out passes are made after the final plunge depth has been reached. Spark-out refers to repeating a final grinding pass without changing plunge depth until there are no "sparks" (no material being removed) from the part surface. It is a complete repetition of the final grinding pass, whether it was only reciprocating motions (G82) or reciprocating and crossover motions. If the last spark out pass leaves the crossover axis at its start coordinates instead of the end point of the crossover move, then one additional spark-out pass is made.

This parameter is not "program modal." If a value for the number of spark-out passes is not programmed, then no spark-out passes are executed for that cycle.

F - cross and plunge pick feedrate.

The feedrate entered here is for the cross and plunge axes. It must be within the range of legal F words defined for your system. If no value is entered for this, then the last F word executed in the part program is used as the cross and plunge pick feedrate.

This parameter is the system F word. Programming it here alters the feedrate for any subsequent axis motion. Programming F0 selects the rapid feedrate.

E - reciprocation feedrate.

The feedrate entered here is for the reciprocating axis. It must be within the range of legal F words defined for your system. Programming a value of zero with the E parameter is illegal and an error results. If the E parameter is not programmed, then the F parameter value is used.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent surface grinding blocks (G81 – G86) use a previously programmed value if a new value is not programmed.

P - dress program number.

The number entered here must be a legal program number (a program that has been saved as a subprogram using the O word followed by a number of up to five digits). This dress program is used for the pre-dress and/or auto-dress operation and is executed the number of times defined by the D parameter.

If the surface grinding cycle block calls for a pre-dress operation (G81.1 to G86.1), then the first dress program execution occurs before the actual surface grinding cycle begins, regardless of the number entered for the D parameter.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent grinding cycle blocks (G81 – G86) use a value from a previous grinding cycle block if a new value is not programmed.

Important: It is the programmer's responsibility to make sure that the dress program can be executed safely from the reciprocation reversal points.

D - number of auto-dress executions.

The number entered here (any integer from 0 to 999) indicates how many times the dress program (P) is executed over the duration of the grinding cycle. The way these dress operations are distributed throughout the cycle varies, depending on the particular cycle. In general, the dress operations occur immediately after a plunge move at reciprocation reversal points.

This parameter is not "program modal." If a value for the number of dress executions is not programmed, then the dress program P is not executed for that cycle. This does not affect pre-dress requests or operator requested dressing as described in chapter 21.

16.3 G81 or G81.1 Reciprocation Without Cross Pick or Plunge

The format for the G81 reciprocation mode is as follows:

```
G81Z__K__E__P__;

G81Y__J__E__P__;

G81X__I__E__P__;
```

Table 16.A summarizes the G81 mode parameters. For a detailed description of these parameters, see the text below and page 16-8.

Table 16.A
G81 Mode Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Programmed axis [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Reciprocation axis integrand [3]	reciprocation secondary reversal point	from last cycle programmed	abs. or inc.
E [2]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G81 2 for G81.1]	dress program number	from last cycle programmed	valid program number

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Programming a G81 or G81.1 causes the control to execute a grinding reciprocation without a coordinated pick or plunge motion. The G81.1 begins with a pre-dress operation prior to the reciprocation move. Reciprocation motion is described on page 16-4 of this chapter.

This reciprocation move is executed on whichever axis is programmed in the G81 block, regardless of the active plane. The value entered with the axis programmed defines the primary reversal point for the reciprocating axis. It is entered as either a distance from the start point (G91) or as an absolute position (G90). The integrand for the axis programmed defines the secondary reversal point, either as a distance from the start point (G91) or as an absolute position (G90).

16.4 G82 or G82.1 Plunge Grinding (Slot)

The format for the G82 cycle is as follows:

```
G17(XY);
G82Z__K__X__I__Q__L__F__E__P__D__;
     or
G82Z__K__Y__J__Q__L__F__E__P__D__;

G18(ZX);
G82Y__J__Z__K__Q__L__F__E__P__D__;
     or
G82Y__J__X__I__Q__L__F__E__P__D__;

G19(YZ);
G82X__I__Y__J__Q__L__F__E__P__D__;
     or
G82X__I__Z__K__Q__L__F__E__P__D__;
```

Table 16.B summarizes the G82 cycle parameters. For a detailed description of these parameters, see the text below and page 16-8.

Table 16.B
G82 Cycle Parameters

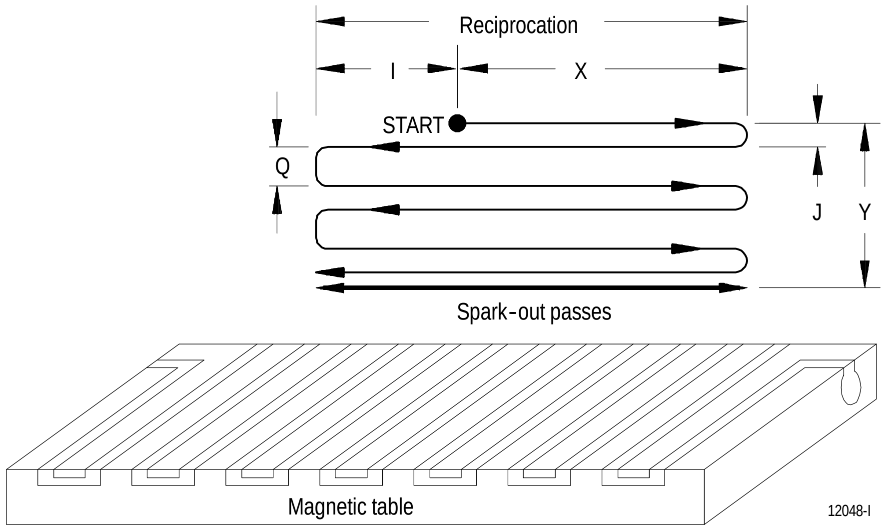
Parameter:	Definition:	Default Value:	Possible Value:
Non-plane axis [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Reciprocation axis integrand [3]	reciprocation secondary reversal point	from last cycle programmed	abs. or inc.
Either plane axis [2]	plunge depth	from last cycle programmed	abs. or inc.
Plunge axis integrand [2]	plunge pick at primary reversal point	from last cycle programmed	inc. only
Q [3]	plunge pick at secondary reversal point	plunge pick at primary reversal	inc. only
L [3]	number of spark-out passes	zero	zero to 999
F [2]	plunge pick feedrate	last feedrate programmed	valid F word
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G82 2 for G82.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed   [2] must have been programmed in some previous block if omitted   [3] optional

Programming both plane axes in the same G82 block results in the error "TOO MANY AXES PROGRAMMED"

Figure 16.4 shows the axis motions that make up the G82 plunge grinding cycle. This figure assumes that the YZ plane (G19) is active and that the Y axis is programmed in the G82 block, making it the plunge axis.

Figure 16.4
G82 Plunge Grinding Motions



Programming a G82 or G82.1 causes the control to execute a plunge grinding cycle using only two axes. You can use this cycle to grind a slot into a part or to grind a part that is narrower than the grinding wheel or grinding path. The G82.1 begins with a pre-dress operation prior to the grind.

The plunge axis for this cycle is whichever axis of the active plane that is programmed in the G82 block. Only one of the axes in the active plane should be programmed or an error results. The reciprocation axis for this cycle is the axis that is not in the active plane.

For example, assume that the G19 (YZ) plane is active and you have configured the Y axis as axis one and the Z axis as axis two for that plane. The cycle executes as follows:

For example, if the G19 (YZ) plane is active and configured such that the Y axis is axis one and the Z axis is axis two, then you could program either Y or Z in the G82 block, but not both. The X axis would be the reciprocating axis. Assuming the Y axis is to be used for the plunge depth, the cycle executes as follows:

1. The X axis moves to the primary reversal point (X) at feedrate E.

2. The plunge pick move (J) begins at feedrate F as the X axis decelerates to the primary reversal point.

Important: It is the programmer's responsibility to make sure that the reciprocation moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

3. The X axis moves to the secondary reversal point (I).
4. The plunge pick move (Q) begins as the X axis decelerates to the secondary reversal point.
5. The reciprocation and plunge pick moves continue until the plunge depth (Y) is reached.
6. After the plunge depth (Y) is reached, reciprocation continues and the programmed number of spark-out passes are executed.

16.5 G83 or G83.1 Incremental Plane Grinding (Axis 1)

The format for the G83 cycle is as follows:

```
G83X__I__Y__J__Z__K__R__Q__L__F__E__P__D__;
```

Table 16.C summarizes the G83 cycle parameters. For a detailed description of these parameters, see the text below and page 16-8.

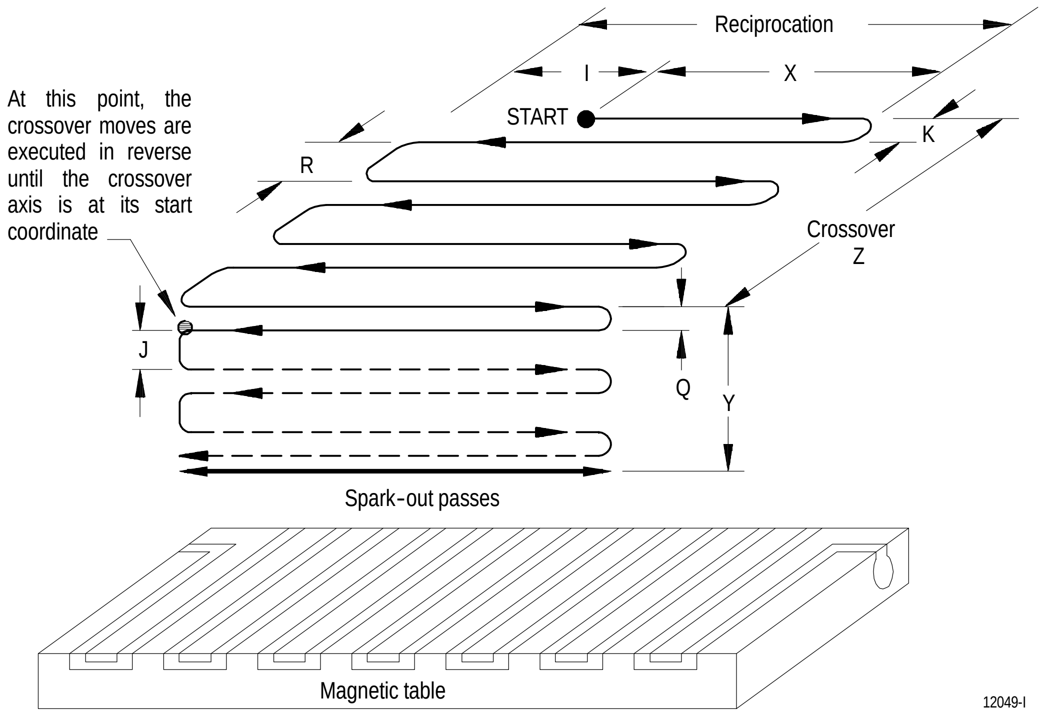
Table 16.C
G83 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Non-plane axis [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Reciprocation axis integrand [3]	reciprocation secondary reversal point	from last cycle programmed	abs. or inc.
Plane axis 1 [2]	plunge depth	from last cycle programmed	abs. or inc.
Plane axis 1 integrand [2]	plunge pick at start	from last cycle programmed	inc. only
Q [3]	plunge pick at crossover	plunge pick at start	inc. only
Plane axis 2 [2]	crossover amount	from last cycle programmed	abs. or inc.
Plane axis 2 integrand [2]	cross pick at primary reversal point	from last cycle programmed	inc. only
R [3]	cross pick at secondary reversal point	cross pick at primary reversal	inc. only
L [3]	number of spark-out passes	zero	zero to 999
F [2]	plunge pick and cross pick feedrate	last feedrate programmed	valid F word
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G83 2 for G83.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 16.5 shows the axis motions that make up the G83 incremental plane grinding cycle. This figure assumes that the YZ plane (G19) is active. Since the Y axis is plane axis 1 for G19, it is the plunge axis.

Figure 16.5
G83 Incremental Plane Grinding Motions



Programming a G83 or G83.1 causes the control to execute an incremental plane grinding cycle using three axes. "Incremental" refers to cross pick move executes in a single increment at the end of each reciprocation move. A plunge move towards the part is made after each crossover is completed. The G83.1 begins with a pre-dress operation prior to the grind.

The G83 cycle dictates that the axis configured as axis one in the active plane makes the plunge move. The axis configured as axis two in the active plane makes the cross pick move, while the axis not in the active plane is the reciprocating axis.

For example, assume that the G19 (YZ) plane is active and you have configured the Y axis as axis one and the Z axis as axis two for that plane. The cycle executes as follows:

1. The X axis moves to the primary reversal point (X) at feedrate E.

2. The cross pick move (K) begins at feedrate F as the X axis decelerates to the primary reversal point.

   Important: Reciprocation motion does not wait for pick moves to complete. It is the programmer's responsibility to make sure that the pick moves are short enough or fast enough to complete before the reciprocation axis moves back over the part.

3. The X axis moves to the secondary reversal point (I).

4. The cross pick move (R) begins as the X axis decelerates to the secondary reversal point.

5. The reciprocation and cross pick moves continue until the final crossover amount (Z) is reached.

6. After the crossover is complete, a plunge pick move (Q) begins at feedrate F as the X axis decelerates to whichever reversal point it reaches first.

   Important: It is the programmer's responsibility to make sure that the crossover and reciprocating moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Crossover and plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

7. The cross pick moves resume, this time with the cross pick moves K and R being executed in the opposite direction.

8. After the crossover is completed in reverse and the Z axis is at its start point, a plunge pick move (J) begins as the X axis decelerates to whichever reversal point it reaches first.

9. The reciprocation, cross pick, and plunge pick moves continue until the plunge depth (Y) is reached.

10. After the plunge depth (Y) is reached, reciprocation and crossover motion continues and the spark-out passes begin.

11. After the programmed number of spark-out passes have been executed, if the Z axis is not at its end point (Z), an additional spark out pass is made.

16.6 G84 or G84.1 Incremental Plane Grinding (Axis 2)

The format for the G84 cycle is as follows:

```
G84X__I__Y__J__Z__K__R__Q__L__F__E__P__D__;
```

Table 16.D summarizes the G84 cycle parameters. For a detailed description of these parameters, see the text below and page 16-8.

Table 16.D
G84 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Non-plane axis [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Reciprocation axis integrand [3]	reciprocation secondary reversal point	from last cycle programmed	abs. or inc.
Plane axis 1 [2]	crossover amount	from last cycle programmed	abs. or inc.
Plane axis 1 integrand [2]	cross pick at primary reversal point	from last cycle programmed	inc. only
Q [3]	cross pick at secondary reversal point	cross pick at primary reversal	inc. only
Plane axis 2 [2]	plunge depth	from last cycle programmed	abs. or inc.
Plane axis 2 integrand [2]	plunge pick at start	from last cycle programmed	inc. only
R [3]	plunge pick at crossover	plunge pick at start	inc. only
L [3]	number of spark-out passes	zero	zero to 999
F [2]	plunge pick and cross pick feedrate	last feedrate programmed	valid F word
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G84 2 for G84.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 16.5 in the previous section shows the axis motions that make up the G83 incremental plane grinding cycle when the G19 (YZ) plane is active. For the G84 cycle, these motions would be identical except that the Z axis (plane axis 2 for G19) would be the plunge axis.

Programming a G84 or G84.1 causes the control to execute an incremental plane grinding cycle using three axes. "Incremental" refers to the fact that the cross pick move is executed in a single increment at the end of each reciprocation move. A plunge move towards the part is made after each crossover is completed. The G84.1 begins with a pre-dress operation prior to the grind.

The G84 cycle dictates that the axis configured as axis two in the active plane makes the plunge moves. The axis configured as axis one in the active plane makes the cross pick moves, while the axis not in the active plane is the reciprocating axis.

For example, assume that the G19 (YZ) plane is active and you have configured the Y axis as axis one and the Z axis as axis two for that plane. The G84 cycle would execute as described for the G83 cycle except that the cross pick moves would be made by the Y axis and the plunge moves would be made by the Z axis.

Important: It is the programmer's responsibility to make sure that the crossover and reciprocating moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Crossover and plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

16.7 G85 or G85.1 Continuous Plane Grinding (Axis 1)

The format for the G85 cycle is as follows:

```
G17(XY);
G85Z__K__X__I__Y__Q__L__F__E__P__D__;

G18(ZX);
G85Y__J__Z__K__X__Q__L__F__E__P__D__;

G19(YZ);
G85X__I__Y__J__Z__Q__L__F__E__P__D__;
```

Table 16.E summarizes the G85 cycle parameters. For a detailed description of these parameters, see the text below and page 16-8.

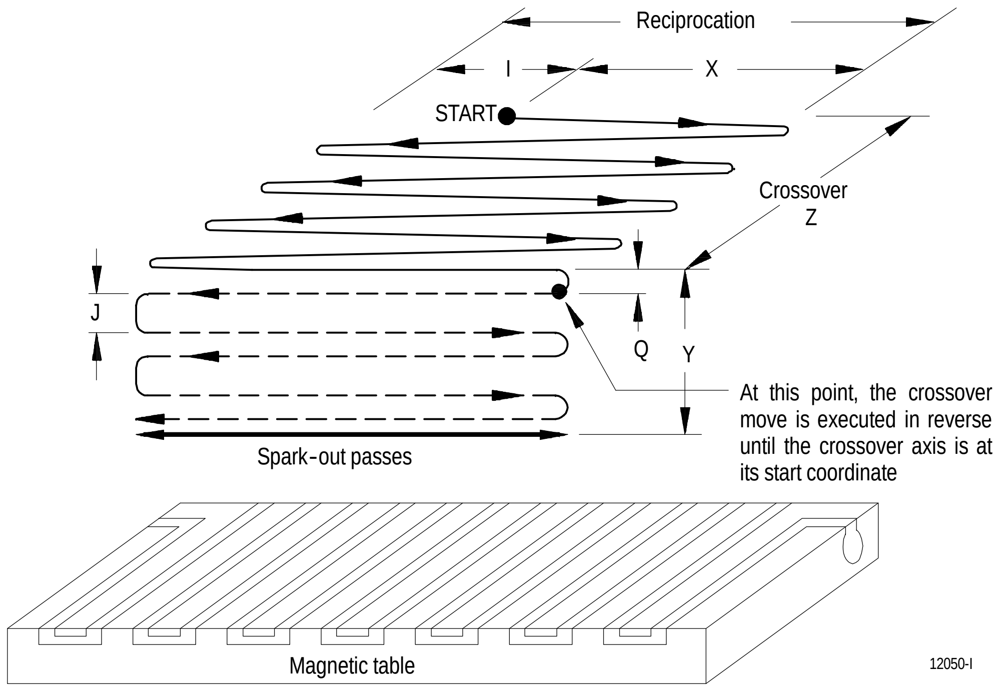
Table 16.E
G85 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Non-plane axis [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Reciprocation axis integrand [3]	reciprocation secondary reversal point	from last cycle programmed	abs. or inc.
Plane axis 1 [2]	plunge depth	from last cycle programmed	abs. or inc.
Plane axis 1 integrand [2]	plunge pick at start	from last cycle programmed	inc. only
Q [3]	plunge pick at crossover	plunge pick at start	inc. only
Plane axis 2 [2]	crossover amount	from last cycle programmed	abs. or inc.
L [3]	number of spark-out passes	zero	zero to 999
F [2]	plunge pick and crossover feedrate	last feedrate programmed	valid F word
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G85 2 for G85.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 16.6 shows the axis motions that make up the G85 continuous plane grinding cycle. This figure assumes that the YZ plane (G19) is active. Since the Y axis is plane axis 1 for G19, it is the plunge axis.

Figure 16.6
G85 Continuous Plane Grinding Motions



Programming a G85 or G85.1 causes the control to execute a continuous plane grinding cycle using three axes. “Continuous” refers to the fact that instead of making cross pick moves, the crossover axis moves continuously towards its endpoint as the reciprocating axis moves back and forth across the part. The G85.1 begins with a pre-dress operation prior to the grind.

The G85 cycle dictates that the axis configured as axis one in the active plane makes the plunge moves. The axis configured as axis two in the active plane makes the crossover move, while any axis programmed that is not in the active plane is the reciprocating axis. If reciprocation is already active, it is not necessary to program a reciprocation axis in the G85 block.

For example, assume that the G19 (YZ) plane is active and you have configured the Y axis as axis one and the Z axis as axis two for that plane. The cycle executes as follows:

1. The X axis and the Z axis begin moving simultaneously. The X axis begins its reciprocation motion by moving to the primary reversal point (X). The Z axis begins the crossover motion by moving towards the crossover endpoint defined by the Z parameter.

2. The crossover motion continues at feedrate F until the distance defined by the Z parameter is covered. At that point Z axis motion is stopped.

3. Reciprocation motion (at feedrate E) continues after the Z distance is covered until either the primary or secondary reversal point is reached.

4. A plunge pick move (Q) begins as the X axis decelerates to whichever reversal point it reaches first.

   Important: It is the programmer's responsibility to make sure that the crossover and reciprocating moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Crossover and plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

5. The crossover motion resumes, this time with the crossover motion being executed in the opposite direction.

6. After the crossover is completed in reverse and the Z axis is at its start point, a plunge pick move (J) begins as the X axis decelerates to whichever reversal point it reaches first.

7. The reciprocation, crossover, and plunge pick moves continue until the plunge depth (Y) is reached.

8. After the plunge depth (Y) is reached, reciprocation and crossover motion continues and the spark-out passes begin.

9. After the programmed number of spark-out passes have been executed, if the Z axis is not at its end point (Z), an additional spark out pass is made.

16.8 G86 or G86.1 Continuous Plane Grinding (Axis 2)

The format for the G86 cycle is as follows:

```
G17(XY);
G86Z__K__X__I__Y__Q__L__F__E__P__D__;

G18(zx);
G86Y__J__Z__K__X__Q__L__F__E__P__D__;

G19(YZ);
G86X__I__Y__J__Z__Q__L__F__E__P__D__;
```

Table 16.F summarizes the G86 cycle parameters. For a detailed description of these parameters, see the text below and page 16-8.

Table 16.F
G86 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Non-plane axis [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Reciprocation axis integrand [3]	reciprocation secondary reversal point	from last cycle programmed	abs. or inc.
Plane axis 1 [2]	crossover amount	from last cycle programmed	abs. or inc.
Plane axis 2 [2]	plunge depth	from last cycle programmed	abs. or inc.
Plane axis 2 integrand [2]	plunge pick at start	from last cycle programmed	inc. only
R [3]	plunge pick at crossover	plunge pick at start	inc. only
L [3]	number of spark-out passes	zero	zero to 999
F [2]	plunge pick and crossover feedrate	last feedrate programmed	valid F word
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G86 2 for G86.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 16.6 in the previous section shows the axis motions that make up the G85 continuous plane grinding cycle when the G19 (YZ) plane is active. For the G86 cycle, these motions would be identical except that the Z axis (plane axis 2 for G19) would be the plunge axis.

Programming a G86 or G86.1 causes the control to execute a continuous plane grinding cycle using three axes. “Continuous” refers to the fact that instead of making cross pick moves, the crossover axis moves continuously towards its endpoint as the reciprocating axis moves back and forth across the part. The G86.1 begins with a pre-dress operation prior to the grind.

The G86 cycle dictates that the axis configured as axis two in the active plane makes the plunge moves. The axis configured as axis one in the active plane makes the crossover move, while any axis programmed that is not in the active plane is the reciprocating axis. If reciprocation is already active, it is not necessary to program a reciprocation axis in the G86 block.

For example, assume that the G19 (YZ) plane is active and you have configured the Y axis as axis one and the Z axis as axis two for that plane. The G86 cycle would execute as described for the G85 cycle except that the crossover motion would be made by the Y axis and the plunge moves would be made by the Z axis.

Important: It is the programmer's responsibility to make sure that the crossover and reciprocating moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Crossover and plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

END OF CHAPTER

Chapter 17

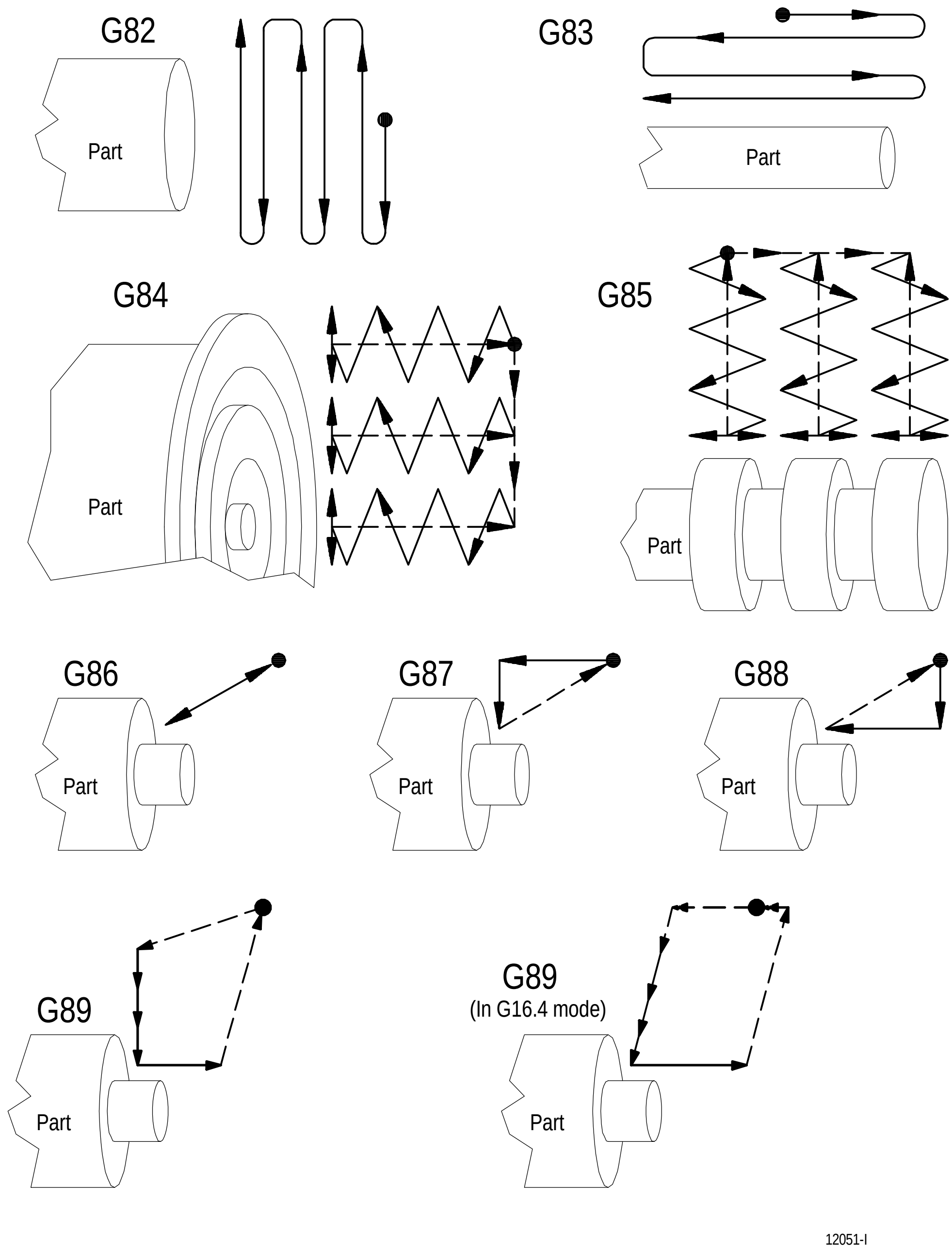
Cylindrical Grinding Fixed Cycles

17.0 Chapter Overview

This chapter describes the cylindrical grinding cycles available with the control. You can use these cycles to program axis motions to perform common grinding operations. Topics include:

Figure 17.1 shows the motions of the eight cylindrical grinding cycles described in this chapter.

Figure 17.1
Cylindrical Grinding Cycles



17.1 Cylindrical Grinding Considerations

Modality and Programming

These cylindrical grinding cycles are modal. Once programmed, the cycle is executed in each subsequent block that contains the appropriate parameters and parameter values. The G codes corresponding to these cycles do not have to be programmed in each block.

For some of these cycles, plunge axis motion is coordinated with the motion of a reciprocating or dithering axis.

You can access Grinder Prompts through the QuickView screens (described in chapter 5) to simplify programming.

Planes

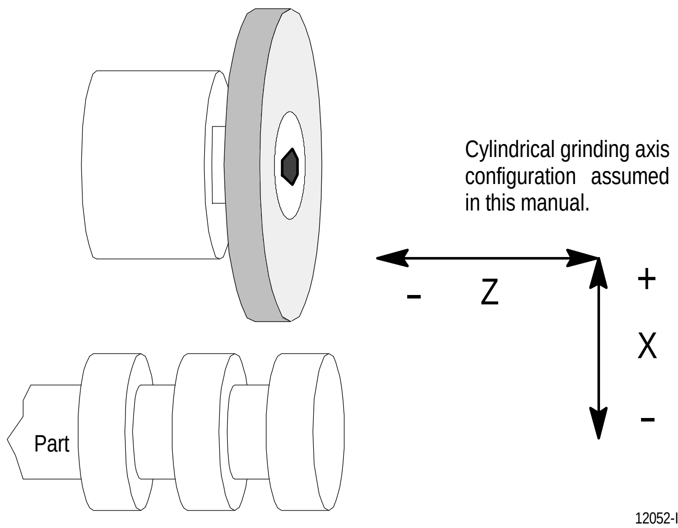
The operation of the cylindrical grinding cycles depends on plane selection. This chapter assumes the following regarding plane configuration for the control. Your axis names and designations can be different. See the literature provided by your system installer.

PLANE	1st Axis	2nd Axis	1st Integrand	2nd Integrand	1st Parallel	2nd Parallel
G17	none	none	none	none	none	none
G18	Z	X	K	I	W	U
G19	none	none	none	none	none	none
Angled Wheel Mode	Z	X	K	I	none	none

Plane selection must be made prior to executing a block that starts the reciprocation axis or a block that starts any of the grinding fixed cycles.

The drawings and descriptions in this chapter assume an Z-X configuration for the axes on a cylindrical grinding machine. Your machine configuration can be different. The illustrations in this chapter generally assume that the G18 (ZX) plane is active and the axes are configured as shown in Figure 17.2.

Figure 17.2
Typical Axis Configuration



Angled-Wheel Mode

Angled-wheel grinders (grinders that have a wheel axis that is not perpendicular to Z) have three operating modes. They are as follows:

Angled-Wheel Grinders in this mode:	Affect to Cylindrical Grinding Cycles:
G15 mode	All cylindrical grinding cycles are available in this mode. We do not recommend angled-wheel grinder programmers attempt to program cycles in G15 mode. G15 mode increases programming difficulty because of the complex geometry that must be performed by the programer to compensate for X and Z motion created by moving the W axis. The active programmed plane (G17, G18, or G19) is used in this mode.
G16.3 mode	All cylindrical grinding cycles are available in this mode. This is the recommended mode for operation of cylindrical cycles on an angled wheel grinder. The control performs all calculations to transfer X and Z axis motions into WZ motions. The control is forced into the ZX plane for all programming in this mode. The currently active selected plane, and programming changes to the active plane are ignored until the G16.3 mode is canceled.
G16.4 mode	Only the G86, G87, G88, and G89 cycles (and their pre-dress counterparts) are available in the G16.4 mode. The control is forced into the ZX plane for all programming in this mode. The currently active selected plane, and programming changes to the active plane are ignored until the G16.4 mode is canceled.

Reciprocation

ATTENTION: Reciprocation differs from conventional axis motion in that the reciprocating axis has no final destination. Once axis reciprocation begins, it continues through program block execution until stopped by a G80, an end of program (M02, M30, M99), or an emergency stop. This means pressing single block or cycle stop does not necessarily stop the reciprocating axis.

Important: You can alter the motion of the reciprocating axis significantly through the control's PAL program. This can affect cycle operation. See your PAL reference manual and/or the literature prepared by your system installer.

Important: The reciprocating axis cannot be mirrored (G51.1) and cannot be rotated (G68) before or during reciprocation.

The reciprocating axis is the axis that sweeps back and forth across the part surface. For the G81 cycle (reciprocation without fixed cycle), the reciprocation axis is whichever axis is programmed in the G81 block. Only one axis name can be programmed in a G81 block.

For the cylindrical grinding cycles, the reciprocation axis may or may not be an axis in the active plane. Different grinding cycles can use different axes for reciprocation even if the active plane does not change. Most of the cylindrical grinding cycles do not use a reciprocating axis.

The reciprocating motion is divided into a primary reversal and a secondary reversal. The reversal points of the primary and secondary reciprocation moves are defined by the numeric values associated with the reciprocating axis name and the reciprocating axis integrand. Reversal points can be an incremental distance from the start point (G91 active), or an absolute position (G90 active).

Important: If your reciprocating axis is configured as a diameter axis in AMP, your reciprocation reversal points are altered by radius or diameter mode (G07/G08).

You can command axis motion while reciprocation mode is active. If the axis to be moved is independent of the reciprocation, then motion begins as soon as the control executes the block.

Certain commands or commanded motions depend on reciprocation and are delayed until the reciprocating axis reaches the secondary reversal point. Following are examples of such commands:

- coordinate changes requested for the reciprocating axis
- certain fixed cycle operations as described in this chapter
- a G80 is executed to cancel reciprocation
- an M99 is executed at the end of a main program

If an incremental plunge is to be made using the G82 or G83 cycles, it takes place as the reciprocating axis decelerates to a reciprocation reversal point.

Programming an incremental or absolute move to the reciprocating axis in any program block while reciprocation is active, can change the coordinates of the primary and secondary reversal end points. In the case of programming moves to the reciprocating axis while in incremental mode, the incremental change is made relative to the secondary reversal point, as illustrated in the following example.

Program Block	Prim. Rev. Point	Sec. Rev. Point	Comments
G90G80F100X10.0;	none	none	no reciprocation
G81G91X5.0I-10.0E100;	X axis, 15.0	X axis, 5.0	from start point of X = 10
X5.0I-8.0;	X axis, 10.0	X axis, 2.0	from sec. rev. point of X = 5
X-20.0I5.0;	X axis, -18.0	X axis, -13.0	from sec. rev. point of X = 2

The preceding example applies not only to the G81 shown, but to the reciprocating axis in any of the grinding fixed cycles.

If already active, reciprocation motion does not stop or pause when a grinding cycle block or a block changing the reciprocation reversal points is executed. Instead, changes to the reciprocation motion take place at the secondary reversal point.

A plane change (G17, G18, G19) can be requested while the reciprocation axis is in motion. If the next grinding cycle block to be executed after the plane change requires a different reciprocation axis, then you must include valid reciprocation axis data with that grinding cycle block. If not, the error message “RECIPROCATION AXIS IN WRONG PLANE” appears and block execution stops. Reciprocation does not stop.

Reciprocation stops when an emergency stop condition occurs. No motion occurs when the emergency stop is reset. If your control's AMP is configured such that the control is not reset after an E-STOP reset, then typically reciprocation resumes with the next cycle start. This allows part programs to be resumed immediately without having to first restart reciprocation. For details, see the documentation prepared by your system installer.

M00 blocks do not affect reciprocation.

The feedrate for the reciprocation motion is determined by the E or F words as described below in order of priority:

- the E word in the reciprocation block
- the E word in a previous reciprocation block executed since the last end of program (M02, M30, M99) or control reset
- the F word in the reciprocation block
- the F word in a previous block executed since the last end of program (M02, M30, M99) or control reset

If there is no active E or F word feedrate, an error is generated as soon as the reciprocation motion is commanded. Your PAL program ultimately determines reciprocation axis speed and motion. See the documentation prepared by your system installer.

Acceleration and deceleration for the reciprocating axis is always linear, regardless of the ACC/DEC type selected for that axis in AMP.

You can determine (through PAL) whether block transitions are made after the reciprocation axis reaches its in-position band (positioning mode) or whether the control executes blocks regardless of the in-position band (non-positioning mode). See the documentation prepared by your system installer.

Plunge

Plunge refers to the axis motion towards the part surface. There are two plunge pick increments definable with the G82 and G83 cycles,

- the plunge pick at the primary reversal point
- the plunge pick at the secondary reversal point

If a plunge shift is made (G84 or G85), it is not made until the spark-out passes are completed, the dither motion has stopped, and the plunge axis has retracted to its start coordinate. As the plunge axis decelerates, the plunge shift (Q) is executed.

Parallel Axes

Parallel axes can be used in the cylindrical grinding cycle blocks in place of the primary axes to which they are parallel. For example, you can program:

```
G18;
G83Z__K__U__I__;
```

where the X axis has been replaced with its parallel axis, U. The integrand for the X axis, I, is used for the U axis. The active plane becomes the ZU plane where Z is axis 1 and U is axis 2 of the plane. The cycle begins with a reciprocating move (Z axis) followed by a plunge move (U axis) of I, made as the Z axis decelerates. See page 17-16 for details on the G83 cycle operation.

Block Displays while Executing Grinding Fixed Cycles

When executing part program blocks, the main screen of the control displays a “*” before a block it is executing and a “@” before a block it has completed. This is typical of any block executed on the control.

Be aware that when block execution involves the reciprocation axis, the block may appear to be handled differently, but actually is not. What frequently happens is that the “@” appears before a grinding fixed cycle block and a “*” appears before the block that follows it even though that block is being held in pre-block. It is being held while waiting for the reciprocation axis to reach its secondary reversal point.

Similarly, the G-code display may show the G code of the following block as active when, again, that block is being held in pre-block waiting for the reciprocation axis to move to its secondary reversal point.

Cancel Grinding and Reciprocation

Use a G80 to cancel all cylindrical grinding cycles. Programming a G80 cancels a G81, G82, G83, G84, G85, G86, G87, G88, or G89. When executed, a G80 also stops the reciprocating axis.

Once reciprocating motion begins, it continues through program block execution until a G80 is executed. If there is no G80, reciprocation continues until an end of program (M02, M30, M99), or an emergency stop condition occurs. An M99 in a subprogram simply returns program execution to the main program and does not cancel reciprocation.

When reciprocation or a grinding cycle is canceled by a G80 or an M99, the reciprocating axis continues to its secondary reversal point before the control considers the G80 or M99 block completed.

17.2 Cylindrical Grinding Parameters

This section describes the parameters that are common to all of the cylindrical grinding cycles. See the section describing each specific cycle for details on the parameters that directly control axis motion.

The system installer can define an axis as a diameter axis in AMP. Only the plunge axis (X), pick axis (Z) and the reciprocation reversal points are affected by diameter/radius programming (G07 or G08). Other parameters are not affected by the controls diameter radius mode.

L – number of spark-out passes or revolutions. The number entered here (any integer from 0 to 999) indicates how many spark-out passes or the number of spindle revolutions at which the wheel is to dwell for spark-out after the final plunge depth has been reached. Spark-out refers to repeating a final grinding pass or holding a final position without changing plunge depth until there are practically no “sparks” (no material being removed) from the part surface.

The L value refers to a number of reciprocation passes for G82 and G83. It refers to a number of dither passes for G84 with an I value programmed, and G85 with a K value programmed. It refers to a number of spindle revolutions for G84 without an I value programmed, and G85 without a K value programmed. It always refers to a number of spindle revolutions for G86, G87, and G88. L can be number of spindle revolutions or number of seconds to dwell during the G89 cycle.

This parameter is not “program modal.” If a value for the number of spark-out passes is not programmed, then no spark-out passes are executed for that cycle.

F – plunge or plunge pick feedrate. The feedrate entered here is for the plunge axis, whether it is making a continuous plunge towards the part, or making incremental plunges. It must be within the range of legal F words defined for your system. If no value is entered for this, then the last F word executed in the part program is used as the plunge pick feedrate.

This parameter is the system F word. Programming it here alters the feedrate for any subsequent axis motion. Programming F0 selects the rapid feedrate.

E – reciprocation, dither, or shoulder feedrate. The feedrate entered here is for the reciprocating axis in G82 and G83. It is the feedrate for the dither axis in G84 and G85. It is the feedrate for the shoulder grinding axis in G87 and G88. It is the medium plunge feedrate of the G89 cycle. It is not used for the G86 grinding cycle. The value entered with this parameter must be within the range of legal F words defined for your system. Programming a value of zero with the E parameter is illegal and an error results (except for the G89 cycle). If the E parameter is not programmed, then the F parameter value is used.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent grinding blocks (G81 – G85, G87, G88, and G89) use a previously programmed value if a new value is not programmed.

P – dress program number. The number entered here must be a legal program number (a program that has been saved as a subprogram using the O word followed by a number of up to five digits). This dress program is used for the pre-dress and/or auto-dress operation and is executed the number of times defined by the D parameter.

If the cylindrical grinding cycle block calls for a pre-dress operation (G81.1 to G89.1), then the first dress program execution occurs before the actual grinding cycle begins, regardless of the number entered for the D parameter.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent grinding cycle blocks (G81 – G86) use a value from a previous grinding cycle block if a new value is not programmed

Important: It is the programmer's responsibility to make sure that the dress program can be executed safely from the reciprocation reversal points.

D – number of auto-dress executions. This parameter is used for the G82 – G85 grinding cycles only and is not valid for the G86 – G89 grinding cycles. The number entered here (any integer from 0 to 999) indicates how many times the dress program (P) is executed over the duration of the grinding cycle. The way these dress operations are distributed throughout the cycle varies, depending on the particular cycle. In general, the dress operations occur immediately after a plunge pick move or immediately before a plunge shift move is made.

This parameter is not "program modal." If a value for the number of dress executions is not programmed, then the dress program P is not executed for that cycle. This does not affect pre-dress requests or operator requested dressing as described in chapter 21.

17.3 G81 or G81.1 Reciprocation Without Plunge

The format for the G81 reciprocation mode is as follows:

```
G81Z__K__E__P__;

G81Y__J__E__P__;

G81X__I__E__P__;
```

Table 17.A summarizes the G81 mode parameters. For a detailed description of these parameters, see the text below and page 17-9.

Table 17.A
G81 Mode Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Programmed axis [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Reciprocation axis integrand [3]	reciprocation secondary reversal point	from last cycle programmed	abs. or inc.
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G81 2 for G81.1]	dress program number	from last cycle programmed	valid program number

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Programming a G81 causes the control to execute a grinding reciprocation without a coordinated pick or plunge motion. The G81.1 is identical, but begins with a pre-dress operation prior to the reciprocation move. Reciprocation motion is described on page 17-5.

This reciprocation move is executed on whichever axis is programmed in the G81 block, regardless of the active plane. The value entered with the axis programmed defines the distance to (G91) or absolute coordinate of (G90) the primary reversal point for the reciprocation axis. The integrand for the axis programmed defines the distance to (G91) or absolute coordinate of (G90) the primary reversal point for the reciprocation axis.

Important: If your reciprocating axis is configured as a diameter axis in AMP, your reciprocation reversal points are altered by radius or diameter mode (G07/G08).

17.4 G82 or G82.1 Incremental Face Grinding (Axis 1)

The format for the G82 cycle is as follows:

```
G18;
G82X__I__Z__K__Q__L__F__E__P__D__;
```

Table 17.B summarizes the G82 cycle parameters. For a detailed description of these parameters, see the text below and page 17-9.

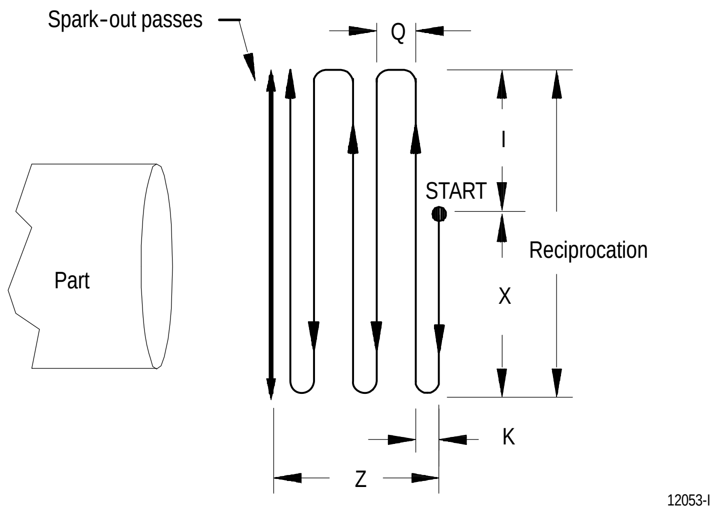
Table 17.B
G82 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Plane axis 1 [1]	plunge depth	value must be programmed	abs. or inc.
Plane axis 1 integrand [1]	plunge pick at primary reversal point	value must be programmed	inc. only
Plane axis 2 [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Plane axis 2 integrand [3]	reciprocation secondary reversal point	start point	abs. or inc.
Q [3]	plunge pick at secondary reversal point	plunge pick at primary reversal	inc. only
L [3]	number of spark-out passes	zero	zero to 999
F [2]	plunge pick feedrate	last feedrate programmed	valid F word
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G82 2 for G82.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 17.3 shows the axis motions that make up the G82 incremental face grinding cycle. This figure assumes that the ZX plane (G18) is active.

Figure 17.3
G82 Incremental Face Grinding Motions



Programming a G82 causes the control to execute a face grinding cycle. This cycle is typically used to grind the face of a part. Axis 2, the X axis in our case, is the reciprocation axis and axis 1, the Z axis, is the plunge axis. The G82.1 cycle is identical but begins with a pre-dress operation prior to the grind.

For example, if the G18 (ZX) plane is active and configured such that the Z axis is axis one and the X axis is axis two, then the Z axis would be the plunge axis and the X axis would be the reciprocating axis. The cycle executes as follows:

1. The X axis moves to the primary reversal point (X) at feedrate E.

2. The plunge pick move (K) begins at feedrate F as the X axis decelerates to the primary reversal point.

Important: It is the programmer's responsibility to make sure that the reciprocation moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

3. The X axis moves to the secondary reversal point (I).

4. The plunge pick move (Q) begins as the X axis decelerates to the secondary reversal point.

5. The reciprocation and plunge pick moves continue until the plunge depth (Z) is reached.

6. After the plunge depth (Z) is reached, reciprocation continues and the programmed number of spark-out passes are executed.

The following describes each of the cycle's parameters.

X – reciprocating axis distance, primary reversal point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the distance from the start point to the end of the primary reciprocating motion (the primary reversal point).

The start point for the reciprocating axis is the coordinate of the X axis prior to execution of the grinding cycle. If the X axis is already reciprocating when the grinding cycle block is executed, then the start point (when incremental mode is active) is the secondary reversal point. This was explained on page 17-5 in the section describing reciprocation.

If in absolute mode (G90 active) then the value entered here is the X coordinate of the primary reversal point.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent grinding blocks that use a reciprocating axis (G81 – G83) use a previously programmed value if a new value is not programmed. Programming an X word in any program block while reciprocation is active changes the coordinate of the primary reversal point.

I – reciprocating axis distance, secondary reversal point. If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the secondary reciprocating motion. If in absolute mode (G90 active) then the value entered here is the X coordinate of the end point of the secondary reciprocating motion. If no value is entered for this, then the start point is used as the end point of the secondary reciprocating motion.

The start point for the reciprocating axis is the coordinate of the X axis prior to execution of the grinding cycle. If the X axis is already reciprocating when the grinding cycle block is executed, then the start point (when incremental mode is active) is the secondary reversal point. This was explained on page 17-5 in the section describing reciprocation.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent grinding blocks that use a reciprocating axis (G81 – G83) use a previously programmed value if a new value is not programmed. Programming an I word in any program block while reciprocation is active changes the coordinate of the secondary reversal point.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the last mode programmed in the block, regardless of their position in the block.

Important: It is the programmer's responsibility to make sure that the reciprocating moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

Z – plunge depth. If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the plunge motion. If in absolute mode (G90 active) then the value entered here is the Z coordinate of the end point of the plunge move.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent cylindrical grinding blocks (G82 – G88) use a previously programmed value if a new value is not programmed.

K – plunge pick amount at primary. The value entered here is an incremental value used to indicate the distance that the plunge axis moves as soon as the reciprocating axis begins decelerating. This is the plunge distance moved as the reciprocating axis reaches the primary reversal point.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent cylindrical grinding blocks (G82 – G88) use a previously programmed value if a new value is not programmed.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the last mode programmed in the block, regardless of their position in the block.

Q – plunge pick amount at secondary. The value entered here is an incremental value used to indicate the distance that the plunge axis moves as soon as the reciprocating axis begins decelerating. This is the plunge distance moved as the reciprocating axis reaches the secondary reversal point. If no value is entered for this, then the K value is used.

L, F, E, P, and D – These parameters were described on page 17-9.

17.5 G83 or G83.1 Incremental Plunge Grinding (Axis 2)

The format for the G83 cycle is as follows:

```
G18;
G83Z__K__X__I__Q__L__F__E__P__D__;
```

Table 17.C summarizes the G83 cycle parameters. For a detailed description of these parameters, see the text below and page 17-9.

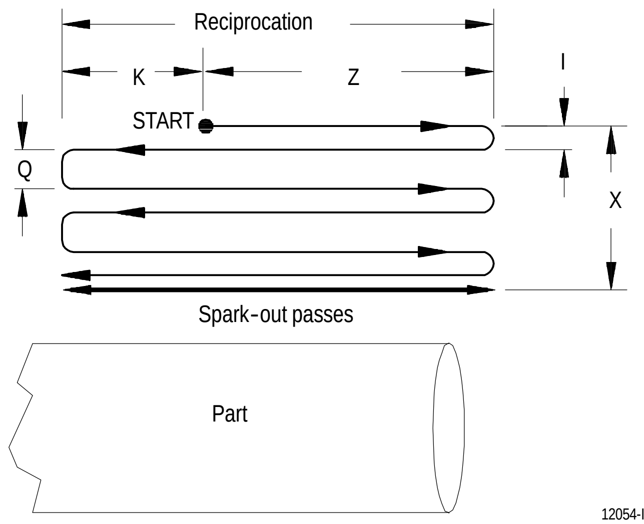
Table 17.C
G83 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Plane axis 1 [2]	reciprocation primary reversal point	from last cycle programmed	abs. or inc.
Plane axis 1 integrand [3]	reciprocation secondary reversal point	start point	abs. or inc.
Plane axis 2 [1]	plunge depth	value must be programmed	abs. or inc.
Plane axis 2 integrand [1]	plunge pick at primary reversal point	value must be programmed	inc. only
Q [3]	plunge pick at secondary reversal point	plunge pick at primary reversal	inc. only
L [3]	number of spark-out passes	zero	zero to 999
F [2]	plunge pick feedrate	last feedrate programmed	valid F word
E [3]	reciprocation feedrate	from last cycle or last F	valid F word
P [3 for G83 2 for G83.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 17.4 shows the axis motions that make up the G83 incremental plunge grinding cycle. This figure assumes that the ZX plane (G18) is active.

Figure 17.4
G83 Incremental Plunge Grinding Motions



Programming a G83 causes the control to execute a plunge grinding cycle. This cycle is typically used to grind the diameter of a part. Axis 1, the Z axis in our case, is the reciprocation axis and axis 2, the X axis, is the plunge axis. The G83.1 cycle is identical but begins with a pre-dress operation prior to the grind.

For example, if the G18 (ZX) plane is active and configured such that the Z axis is axis one and the X axis is axis two, then the X axis would be the plunge axis and the Z axis would be the reciprocating axis. The cycle executes as follows:

1. The Z axis moves to the primary reversal point (Z) at feedrate E.

2. The plunge pick move (I) begins at feedrate F as the Z axis decelerates to the primary reversal point.

Important: It is the programmer’s responsibility to make sure that the reciprocation moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

3. The Z axis moves to the secondary reversal point (K).

4. The plunge pick move (Q) begins as the Z axis decelerates to the secondary reversal point.

5. The reciprocation and plunge pick moves continue until the plunge depth (X) is reached.

6. After the plunge depth (X) is reached, reciprocation continues and the programmed number of spark-out passes are executed.

The following describes each of the cycle's parameters.

Z – reciprocating axis distance, primary reversal point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the distance from the start point to the end of the primary reciprocating motion (the primary reversal point).

The start point for the reciprocating axis is the coordinate of the Z axis prior to execution of the grinding cycle. If the Z axis is already reciprocating when the grinding cycle block is executed, then the start point (when incremental mode is active) is the secondary reversal point. This was explained on page 17-5 in the section describing reciprocation.

If in absolute mode (G90 active) then the value entered here is the Z coordinate of the primary reversal point.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent grinding blocks that use a reciprocating axis (G81 – G83) use a previously programmed value if a new value is not programmed. Programming a Z word in any program block while reciprocation is active changes the coordinate of the primary reversal point.

K – reciprocating axis distance, secondary reversal point. If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the secondary reciprocating motion. If in absolute mode (G90 active) then the value entered here is the Z coordinate of the end point of the secondary reciprocating motion. If no value is entered for this, then the start point is used as the end point of the secondary reciprocating motion.

The start point for the reciprocating axis is the coordinate of the Z axis prior to execution of the grinding cycle. If the Z axis is already reciprocating when the grinding cycle block is executed, then the start point (when incremental mode is active) is the secondary reversal point. This was explained on page 17-5 in the section describing reciprocation.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent grinding blocks that use a reciprocating axis (G81 – G83) use a previously programmed value if a new value is not programmed. Programming a K word in any program block while reciprocation is active changes the coordinate of the secondary reversal point.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the last mode programmed in the block, regardless of their position in the block.

Important: It is the programmer's responsibility to make sure that the reciprocating moves extend beyond the part sufficiently such that the plunge move can be completed before the wheel comes back in contact with the part. Plunge moves begin as the reciprocation axis decelerates. Therefore, feedrate, acceleration and deceleration should be considered for all axes involved.

X – plunge depth. If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the plunge motion. If in absolute mode (G90 active) then the value entered here is the X coordinate of the end point of the plunge move.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent cylindrical grinding blocks (G82 – G88) use a previously programmed value if a new value is not programmed.

I – plunge pick amount at primary. The value entered here is an incremental value used to indicate the distance that the plunge axis moves as soon as the reciprocating axis begins decelerating. This is the plunge distance moved as the reciprocating axis reaches the primary reciprocation point.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent cylindrical grinding blocks (G82 – G88) use a previously programmed value if a new value is not programmed.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the last mode programmed in the block, regardless of their position in the block.

Q – plunge pick amount at secondary. The value entered here is an incremental value used to indicate the distance that the plunge axis moves as soon as the reciprocating axis begins decelerating. This is the plunge distance moved as the reciprocating axis reaches the secondary reciprocation point. If no value is entered for this, then the I value is used.

L, F, E, P, and D – These parameters were described on page 17-9.

17.6 G84 or G84.1 Multi-pass Face Grinding (Axis 1)

The format for the G84 cycle is as follows:

```
G18;
G84X__I__Z__Q__L__F__E__P__D__;
```

Table 17.D summarizes the G84 cycle parameters. For a detailed description of these parameters, see the text below and page 17-9.

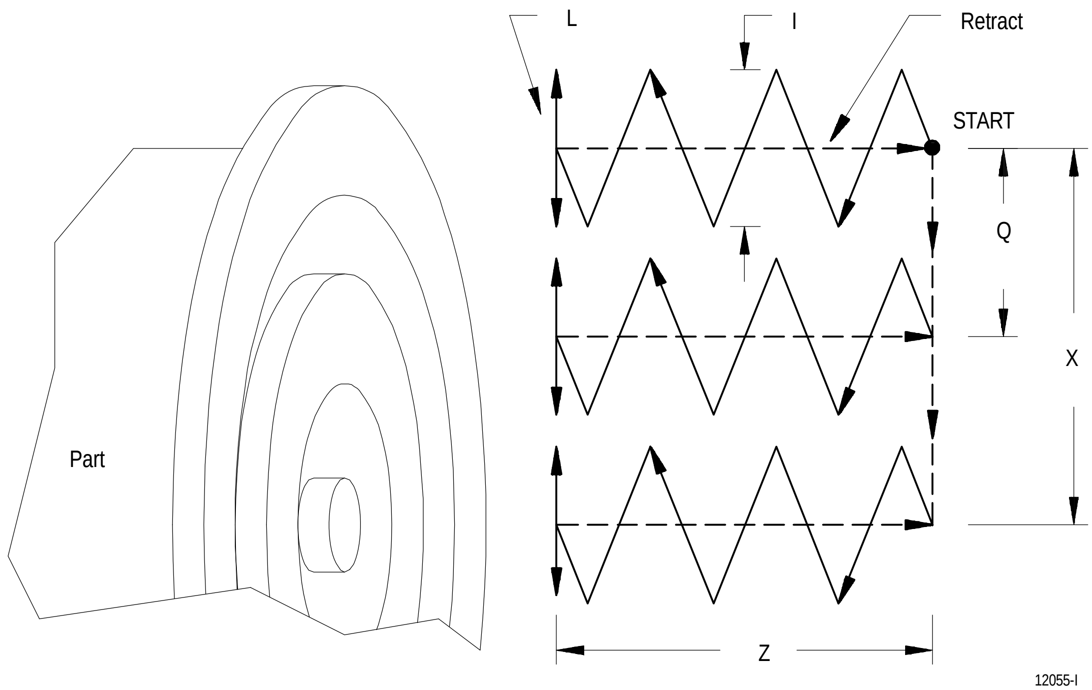
Table 17.D
G84 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Plane axis 1 [1]	plunge depth	value must be programmed	abs. or inc.
Plane axis 2 [3]	last plunge point	none (plunge at start only)	abs. or inc.
Plane axis 2 integrand [3]	dither amount	zero	inc. only
Q [3]	plunge shift	zero	inc. only
L [3]	number of spark-out passes or revolutions	zero	zero to 999
F [2]	plunge feedrate	last feedrate programmed	valid F word
E [3]	dither feedrate	plunge feedrate	valid F word
P [3 for G84] [2 for G84.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 17.5 shows the axis motions that make up the G84 multi-pass face grinding cycle. This figure assumes that the ZX plane (G18) is active.

Figure 17.5
G84 Multi-pass Face Grinding Motions



Programming a G84 causes the control to execute a multi-pass face grinding cycle. This cycle is typically used to grind the face of a part in a situation where a shifting of the grinding wheel position is necessary to cover several different surfaces to be ground on the face. Axis 2, the X axis in our case, is the dither axis and axis 1, the Z axis, is the plunge axis. The G84.1 cycle is identical but begins with a pre-dress operation prior to the grind.

The G84 cycle differs from the G82 cycle in several ways. First, the X axis dither differs from reciprocation in that it is made up motions that are centered about the X axis coordinate and are made only during the plunge move. Second, the plunge motion is a continuous move to the plunge depth (Z) instead of a series of plunge pick moves as used with the G82 cycle.

The third and most significant difference between the G84 and the G82 cycles is that the G84 incorporates an X axis shift, made after the completion of each plunge. The length of this plunge shift (distance between plunges) is defined by the Q parameter.

The following describes each of the cycle's parameters.

X – last plunge point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the distance from the start point to the point where the last plunge motion (defined by the Z parameter) is to be executed. The start point is the coordinate of the X axis prior to execution of the grinding cycle.

If in absolute mode (G90 active) then the value entered here is the X coordinate of the last plunge point.

If no value is entered for X, then the plunge move is made from the start point only. Any Q parameter value programmed is ignored.

I – dither amount. The value entered here is a signed incremental value used to indicate the <u>total</u> distance that the X axis is to dither about the start point. The dither axis oscillates half the incremental distance entered here on both sides of its plunge point. If no value or a value of zero is entered for this, then the plunge motion is executed without any X axis dither.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the <u>last mode</u> programmed in the block, regardless of their position in the block.

Z – plunge depth. If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the plunge motion. If in absolute mode (G90 active) then the value entered here is the Z coordinate of the end point of the plunge move.

The plunge move with the G84 cycle is made with one continuous motion until the plunge depth is reached. At that point, any programmed spark-out passes are made (as defined by the L parameter). After the spark-out passes are completed, the plunge axis retracts to the start point at its rapid feedrate.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent cylindrical grinding blocks (G82 – G88) use a previously programmed value if a new value is not programmed.

Q – plunge shift. The value entered here is an incremental value used to indicate the distance that the X axis is to shift after the plunge axis has retracted. This shift takes place after the plunge axis has retracted and is executed at the rapid feedrate for the X axis.

If no value is entered for Q then no shift is made. If a value greater than or equal to X is entered, then the shift is made only to the last plunge point defined by X. If the sign of the value entered here sends the axis away from the last plunge point instead of towards it, an error is generated.

L, F, E, P, and D – These parameters were described on page 17-9.

17.7 G85 or G85.1 Multi-pass Diameter Grinding (Axis 2)

The format for the G85 cycle is as follows:

```
G18;
G85Z__K__X__Q__L__F__E__P__D__;
```

Table 17.E summarizes the G85 cycle parameters. For a detailed description of these parameters, see the text below and page 17-9.

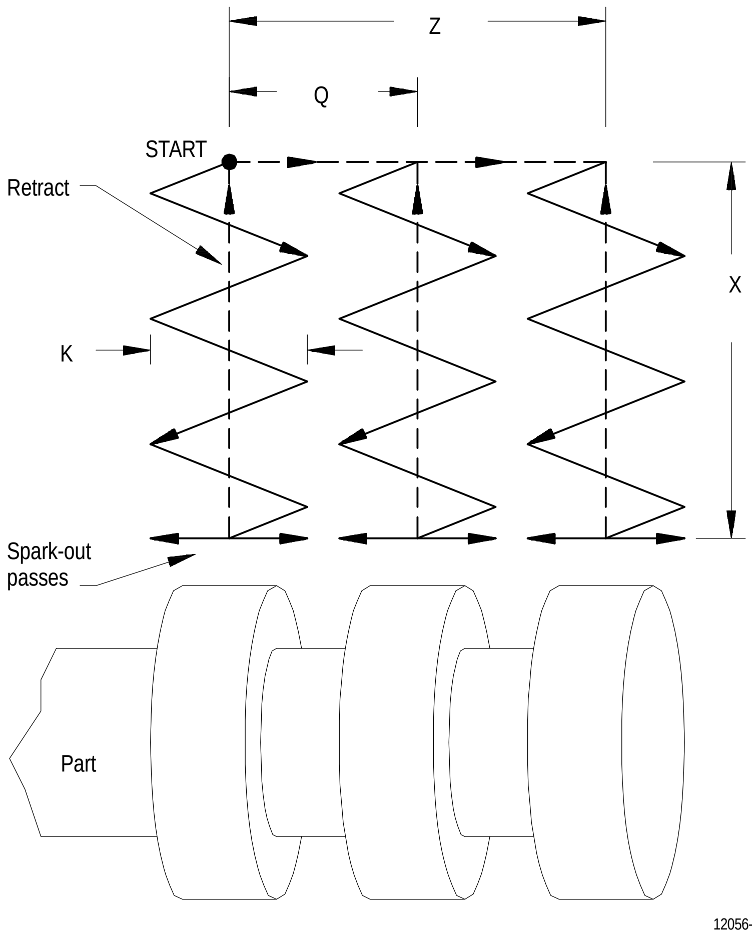
Table 17.E
G85 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Plane axis 1 [3]	last plunge point	none (plunge at start only)	abs. or inc.
Plane axis 1 integrand [3]	dither amount	zero	inc. only
Plane axis 2 [1]	plunge depth	value must be programmed	abs. or inc.
Q [3]	plunge shift	zero	inc. only
L [3]	number of spark-out passes or revolutions	zero	zero to 999
F [2]	plunge feedrate	last feedrate programmed	valid F word
E [3]	dither feedrate	plunge feedrate	valid F word
P [3 for G85 2 for G85.1]	dress program number	from last cycle programmed	valid program number
D [3]	number of auto-dress executions	zero	zero to 999

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 17.6 shows the axis motions that make up the G85 multi-pass diameter grinding cycle. This figure assumes that the ZX plane (G18) is active.

Figure 17.6
G85 Multi-pass Diameter Grinding Motions



Programming a G85 causes the control to execute a multi-pass diameter grinding cycle. This cycle is typically used to grind the diameter of a part in a situation where a shifting of the grinding wheel position is necessary to cover several different surfaces that are to be ground at the same diameter. Axis 1, the Z axis in our case, is the dither axis and axis 2, the X axis, is the plunge axis. The G85.1 cycle is identical but begins with a pre-dress operation prior to the grind.

The G85 cycle differs from the G83 cycle in several ways. First, the Z axis dither differs from reciprocation in that it is made up motions of that are centered about the Z axis coordinate and are made only during the plunge move. Second, the plunge motion is a continuous move to the plunge depth (X) instead of a series of plunge pick moves as used with the G83 cycle.

The third and most significant difference between the G85 and the G83 cycles is that the G85 incorporates a Z axis shift, made after the completion of each plunge. The length of this shift (distance between plunges) is defined by the Q parameter.

The following describes each of the cycle's parameters.

Z – last plunge point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the distance from the start point to the point where the last plunge motion (defined by the X parameter) is to be executed. The start point is the coordinate of the Z axis prior to execution of the grinding cycle.

If in absolute mode (G90 active) then the value entered here is the Z coordinate of the last plunge point.

If no value is entered for X, then the plunge move is made from the start point only. Any Q parameter value programmed is ignored.

K – dither amount. The value entered here is a signed incremental value used to indicate the total distance that the Z axis is to dither about the start point. The dither axis oscillates half the incremental distance entered here on both sides of its plunge point. If no value or a value of zero is entered for this, then the plunge motion is executed without any Z axis dither.

Important: In grinding fixed cycle blocks where changes are made from absolute to incremental or incremental to absolute modes within the block, the integrands I, J and K are always in the last mode programmed in the block, regardless of their position in the block.

X – plunge depth. If in incremental mode (G91 active) then the value entered here is a signed incremental value used to indicate the distance from the start point to the end of the plunge motion. If in absolute mode (G90 active) then the value entered here is the X coordinate of the end point of the plunge move.

The plunge move with the G85 cycle is made with one continuous motion until the plunge depth is reached. At that point, any programmed spark-out passes are made (as defined by the L parameter). After the spark-out passes are completed, the plunge axis retracts to the start point at its rapid feedrate.

This parameter is "program modal," meaning that it needs to be programmed only once in a part program. Any subsequent cylindrical grinding blocks (G82 – G88) use a previously programmed value if a new value is not programmed.

Q – plunge shift. The value entered here is an incremental value used to indicate the distance that the Z axis is to shift after the plunge axis has retracted. This shift takes place after the plunge axis has retracted and is executed at the rapid feedrate for the Z axis.

If no value is entered for this then no shift is made. If a value greater than or equal to Z is entered, then the shift is made only to the last plunge point defined by Z. If the sign of the value entered here sends the axis away from the last plunge point instead of towards it, an error is generated.

L, F, E, P, and D – These parameters were described on page 17-9.

17.8 G86 or G86.1 Shoulder Grinding

The format for the G86 cycle is as follows:

```
G18;
G86Z__X__L__F__P__;
```

Table 17.F summarizes the G86 cycle parameters. For a detailed description of these parameters, see the text below and page 17-9.

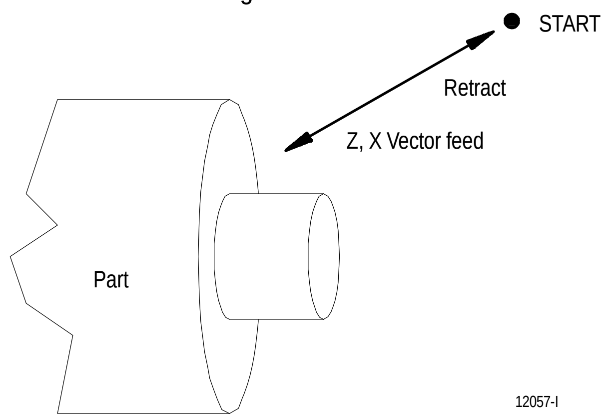
Table 17.F
G86 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Plane axis 1 [1]	plunge end point	value must be programmed	abs. or inc.
Plane axis 2 [1]	plunge end point	value must be programmed	abs. or inc.
L [3]	number of spark-out revolutions	zero	zero to 999
F [2]	plunge feedrate	last feedrate programmed	valid F word
P [3 for G86 2 for G86.1]	dress program number	from last cycle programmed	valid program number

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 17.7 shows the axis motions that make up the G86 shoulder grinding cycle. This figure assumes that the ZX plane (G18) is active.

Figure 17.7
G86 Shoulder Grinding Motions



Programming a G86 causes the control to execute a single vector move to the programmed axis 1 (Z) and axis 2 (X) plunge end points at the plunge feedrate F. If a number of spark-out revolutions (L) is programmed in the block, the axes dwell at the plunge depth for the designated number of revolutions of the spindle. Then they retract to the start position at the rapid feedrate. The G86.1 cycle is identical but begins with a pre-dress operation prior to the grind.

The following describes each of the cycle's parameters.

Z – plunge end point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the Z distance from the start point to the end point of the plunge motion. The start point is the coordinate of the Z axis prior to execution of the grinding cycle.

If in absolute mode (G90 active) then the value entered here is the Z coordinate of the plunge end point.

This parameter is not "program modal," meaning that it must be programmed in the G86 block. If not programmed, the error "FEED AXIS DATA NOT PROGRAMMED" results.

X – plunge end point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the X distance from the start point to the end point of the plunge motion. The start point is the coordinate of the X axis prior to execution of the grinding cycle.

If in absolute mode (G90 active) then the value entered here is the X coordinate of the plunge end point.

This parameter is not "program modal," meaning that it must be programmed in the G86 block. If not programmed, the error "FEED AXIS DATA NOT PROGRAMMED" results.

L, F and P – These parameters were described on page 17-9.

17.9 G87 or G87.1 Shoulder Grinding With Face Plunge

The format for the G87 cycle is as follows:

```
G18;
G87Z__X__L__F__E__P__;
```

Table 17.G summarizes the G87 cycle parameters. For a detailed description of these parameters, see the text below and page 17-9.

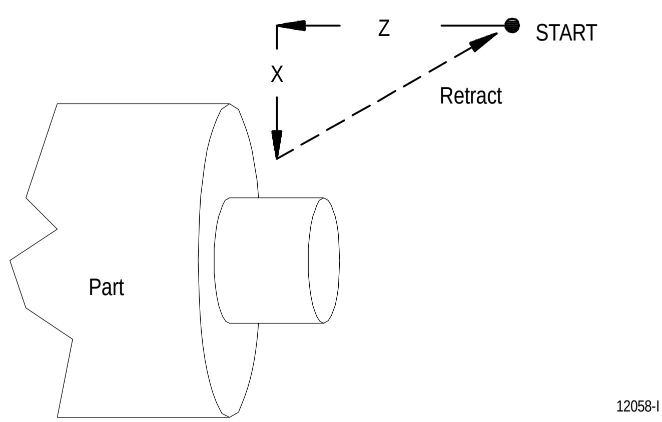
Table 17.G
G87 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Plane axis 1 [1]	plunge end point	value must be programmed	abs. or inc.
Plane axis 2 [1]	shoulder end point	value must be programmed	abs. or inc.
L [3]	number of spark-out revolutions	zero	zero to 999
F [2]	plunge feedrate	last feedrate programmed	valid F word
E [3]	shoulder feedrate	plunge feedrate	valid F word
P [3 for G87 2 for G87.1]	dress program number	from last cycle programmed	valid program number

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 17.8 shows the axis motions that make up the G87 shoulder grinding with face plunge cycle. This figure assumes that the ZX plane (G18) is active.

Figure 17.8
G87 Shoulder Grinding with Face Plunge Motions



Programming a G87 causes the control to execute two moves to arrive at the final plunge position. First axis 1 (Z) plunges into the part face at feedrate F. Then axis 2 (X) makes a shoulder grind at feedrate E.

If a number of spark-out revolutions (L) is programmed in the block, the axes dwell at the plunge and shoulder depth for the designated number of revolutions of the spindle. Then they simultaneously retract to the start position at the rapid feedrate. The G87.1 cycle is identical but begins with a pre-dress operation prior to the grind.

The following describes each of the cycle's parameters.

Z – plunge end point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the Z distance from the start point to the end point of the plunge motion. The start point is the coordinate of the Z axis prior to execution of the grinding cycle.

If in absolute mode (G90 active) then the value entered here is the Z coordinate of the plunge end point.

This parameter is not "program modal," meaning that it must be programmed in the G87 block. If not programmed, the error "FEED AXIS DATA NOT PROGRAMMED" results.

X – shoulder end point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the X distance from the start point to the end point of the shoulder grinding motion. The start point is the coordinate of the X axis prior to execution of the grinding cycle.

If in absolute mode (G90 active) then the value entered here is the X coordinate of the shoulder grind end point.

This parameter is not "program modal," meaning that it must be programmed in the G87 block. If not programmed, the error "FEED AXIS DATA NOT PROGRAMMED" results.

L, F, E and P – These parameters were described on page 17-9.

17.10 G88 or G88.1 Shoulder Grinding With Diameter Plunge

The format for the G88 cycle is as follows:

```
G18;
G88Z__X__L__F__E__P__;
```

Table 17.H summarizes the G88 cycle parameters. For a detailed description of these parameters, see the text below and page 17-9.

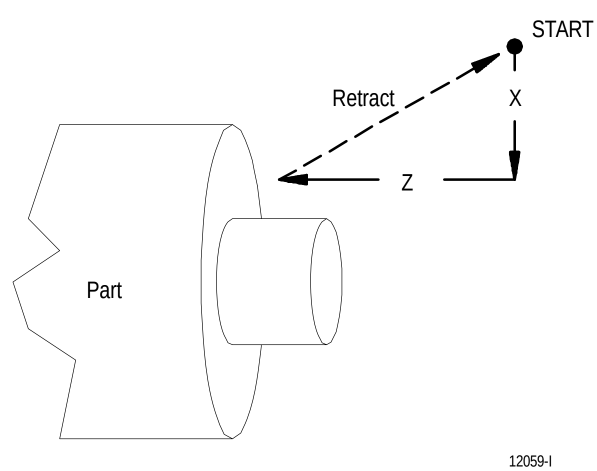
Table 17.H
G88 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
Plane axis 1 [1]	shoulder end point	value must be programmed	abs. or inc.
Plane axis 2 [1]	plunge end point	value must be programmed	abs. or inc.
L [3]	number of spark-out revolutions	zero	zero to 999
F [2]	plunge feedrate	last feedrate programmed	valid F word
E [3]	shoulder feedrate	plunge feedrate	valid F word
P [3 for G88 2 for G88.1]	dress program number	from last cycle programmed	valid program number

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

Figure 17.9 shows the axis motions that make up the G88 shoulder grinding with diameter plunge cycle. This figure assumes that the ZX plane (G18) is active.

Figure 17.9
G88 Shoulder Grinding with Diameter Plunge Motions



Programming a G88 causes the control to execute two moves to arrive at the final plunge position. First axis 2 (X) plunges into the part diameter at feedrate F. Then axis 1 (Z) makes a shoulder grind at feedrate E.

If a number of spark-out revolutions (L) is programmed in the block, the axes dwell at the plunge and shoulder depth for the designated number of revolutions of the spindle. Then they simultaneously retract to the start position at the rapid feedrate. The G88.1 cycle is identical but begins with a pre-dress operation prior to the grind.

The following describes each of the cycle's parameters.

X – plunge end point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the X distance from the start point to the end point of the plunge motion. The start point is the coordinate of the X axis prior to execution of the grinding cycle.

If in absolute mode (G90 active), the value entered here is the X coordinate of the plunge end point.

This parameter is not "program modal," meaning that it must be programmed in the G88 block. If not programmed, the error "FEED AXIS DATA NOT PROGRAMMED" results.

Z – shoulder end point. If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the Z distance from the start point to the end point of the shoulder grinding motion. The start point is the coordinate of the Z axis prior to execution of the grinding cycle.

If in absolute mode (G90 active), the value entered here is the Z coordinate of the shoulder grind end point.

This parameter is not "program modal," meaning that it must be programmed in the G88 block. If not programmed, the error "FEED AXIS DATA NOT PROGRAMMED" results.

L, F, E and P – These parameters were described on page 17-9.

17.11 G89 or G89.1 Multi-Step Plunge with Blend

The G89 cycles contain special features that separate it from other available cycles. These features include:

- Three independent plunge steps occurring at three independent plunge feedrates.
- A micro-feed feature compensates for any wheel wear that occurs during the plunge portion of the cycle.
- A blend operation performed after the wheel has completed the plunge operation with its own blend feedrate.
- This cycle allows programming in both angled-wheel modes (G16.3 and G16.4). You can also execute this cycle on non-angled wheel cylindrical grinders. These non-angled wheel grinders execute the cycle as if in G16.3 normal angled-wheel mode.

Important: The G89 Multi-step Plunge with Blend cycle is not available on 9/240 grinder systems.

To cancel the G89 cycle use either a G80 or one of the other grinder fixed cycles. The first G89 block that you program to enter the G89 mode must at least contain a plunge depth (X) value.

The G89.1 cycle is identical to the G89 cycle but begins with a pre-dress operation prior to the grind. For clarity this manual only discusses the operation of the G89 cycle. Refer to page 17-10 for details on the P word and pre-dress in the G89.1 cycle.

The format for the G89 cycle is as follows:

```
G89Z__X__R__I__Q__K__F__E__,F__,E__L__P__;
```

See Table 17.I for a summary of the G89 cycle parameters. See page 17-9 for a detailed description of these parameters. Figure 17.10 illustrates some of these parameters.

Figure 17.10
G89 Multi-Step Plunge with Blend

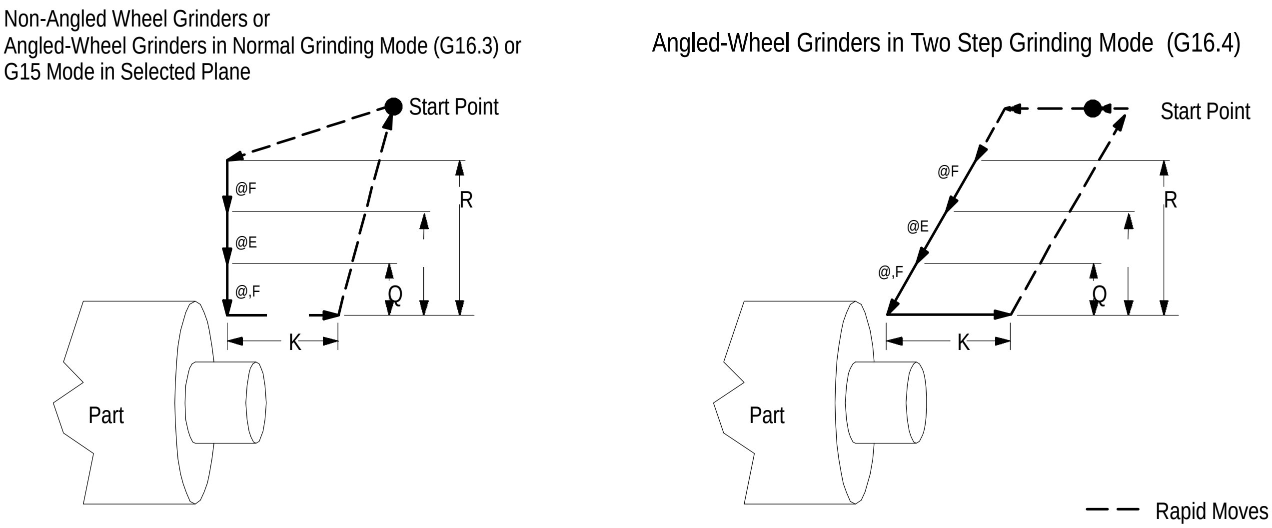


Table 17.I
G89 Cycle Parameters

Parameter:	Definition:	Default Value:	Possible Value:
X [1] (plane axis 1)	plunge end point	value must be programmed	abs. or inc.
Z [1] (plane axis 2)	shoulder end point	modal while in cycle/value must be programmed	abs. or inc.
R [3]	clearance plane	modal while in cycle/zero if not programmed	incremental, unsigned, radius or diameter value
I [3]	medium plunge distance	modal while in cycle/zero if not programmed	incremental, unsigned, radius or diameter value
Q [3]	fine plunge distance	modal while in cycle/zero if not programmed	incremental, unsigned, radius or diameter value
K [2] [3]	blend distance	modal while in cycle/zero if not programmed	incremental, signed, radius or diameter value
F [2]	coarse plunge feedrate	last feedrate programmed	valid F word
E [3]	medium plunge feedrate	modal while in cycle/zero if not programmed	valid F word
,F [3]	fine and micro-feed feedrate	modal while in cycle/zero if not programmed	valid F word
,E [3]	blend feedrate	modal while in cycle/zero if not programmed	valid F word
L [3]	number of spark-out revolutions in IPR mode, seconds in IPM mode, or V/D mode	must be programmed	integer from zero to 999
P [3 for G89 2 for G89.1]	dress program number	from last cycle programmed	valid program number

[1] must be programmed [2] must have been programmed in some previous block if omitted [3] optional

The following describes each of the cycle's parameters.

X – plunge end point
If in absolute mode (G90 active) then the value entered here is the X coordinate of the plunge end point.

If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the X distance from the start point to the end point of the plunge motion. The start point is the coordinate of the X axis prior to execution of the grinding cycle. The plunge end point programmed with the X word must be a different position than the start point of the cycle.

This parameter is "program modal," meaning that it must be programmed in the G89 block but will remain active once it is programmed. If not programmed initially, the error "PLUNGE MOTION NOT PROGRAMMED" results.

Z – shoulder end point
If in absolute mode (G90 active) then the value entered here is the Z coordinate of the shoulder grind end point.

If in incremental mode (G91 active) then the value entered here is a signed incremental value indicating the Z distance from the start point to the end point of the shoulder grinding motion. The start point is the coordinate of the Z axis prior to execution of the grinding cycle.

R – coarse plunge distance
R defines the coarse plunge distance and the clearance plane for the G89 cycle. If R is not programmed the coarse plunge phase of the cycle is not performed. Any sign programmed with the X word is ignored. Always program R as an incremental value. Measure R along the X axis from the final plunge depth programmed with the X word. The R plane must not be within the contours of the rough part. Programming an R plane greater than the starting position (starting position between the R plane and the final depth X) is valid and will not generate an error. The axes rapid to the R plane and perform the coarse plunge at the feedrate F.

Important: If your system installer has installed a means of enabling a micro-feed for this cycle, the micro-feed must be enabled before the axis completes the initial rapid traverse.

I – medium plunge distance
I defines the medium plunge distance. If I is programmed it also defines the endpoint of the previous plunge motion (end point of rough plunge if R is programmed or rapid if no rough plunge is programmed). If I is not programmed the medium plunge phase of the cycle is not performed. Always program I as an unsigned incremental value. Measure I along the X axis from the final plunge depth programmed with the X word. I must be smaller than the previously programmed plunge distance (typically R). The wheel begins to plunge the medium plunge distance once the R plane has been reached. The I plunge is performed at the feedrate E.

Q – fine plunge distance
Q defines the fine plunge distance. If Q is programmed it also defines the endpoint of the previous plunge motion. Always program Q as an unsigned incremental value. Measure Q along the X axis from the final plunge depth programmed with the X word. Q must be smaller than the previously programmed plunge distance (typically I). The wheel begins to plunge the fine plunge distance after the I plunge is completed. The Q plunge is performed at the ,F feedrate.

K – blend distance
K defines the distance of the blend move in the G89 cycle. Always program K as an incremental value. K can be a signed value (+ or –). Measure K in the signed direction along the Z axis from the shoulder end point determined with the Z word. The blend is performed at the ,E feedrate. The wheel performs the blend move after the X plunge depth has been reached and any spark-out (L) has completed.

F – coarse plunge feedrate
The feedrate entered here is for the coarse plunge phase R. F must be within the range of legal F words defined for your system. If no F word is programmed, then the last F word executed in the part program is used as the coarse plunge feedrate. An F word must be programmed when or before the G89 mode is entered. Single digit F words are valid for the coarse plunge feedrate. Programming F0 selects the maximum cutting feedrate configured in AMP.

This parameter is the system F word. Programming it here alters the feedrate for any subsequent axis motion. This feedrate can be programmed in IPM, IPR, or V/D feed modes.

E – medium plunge feedrate
The feedrate entered here is for the medium plunge phase programmed with I. E must be within the range of legal F words defined for your system. The Single Digit F Word feature can not be used to assign E word values. If no value is entered, then the feedrate used for the previous plunge is also used as the medium plunge feedrate.

This parameter is "cycle modal," meaning that it needs to be programmed only once while in the G89 mode. This parameter is reset to zero every time the G89 mode is canceled. This feedrate can be programmed in IPM, IPR, or V/D feed modes.

,F – fine and micro feedrate
The feedrate entered here is for the fine plunge phase programmed with Q and for any micro feed amount appended on to the fine plunge depth. The ,F word must be within the range of legal F words defined for your system. The Single Digit F Word feature can not be used to assign ,F word values. If no value is entered, then the feedrate used during the medium plunge is also used as the fine and micro feedrates.

This parameter is "cycle modal," meaning that it needs to be programmed only once while in the G89 mode. This parameter is reset to zero every time the G89 mode is canceled. This feedrate can be programmed in IPM, IPR, or V/D feed modes. When programmed the comma must immediatly precede the F word (no space).

,E – blend feedrate
The feedrate entered here is for the blend phase programmed with the K word. The ,E word must be within the range of legal F words defined for your system. The Single Digit F Word feature can not be used to assign ,E word values. If no value is entered, then the feedrate used during the fine and micro plunge is also used as the blend feedrate.

This parameter is "cycle modal," meaning that it needs to be programmed only once while in the G89 mode. This parameter is reset to zero every time the G89 mode is canceled. This feedrate can be programmed in IPM, IPR, or V/D feed modes. When programmed the comma must immediately precede the E word (no space).

L – spark-out
If a number of spark-out revolutions (L) is programmed in the block, the axes dwell at the plunge and shoulder depth. L is specified as either the number of revolutions of the part spindle (feed per rev mode G95), or number of seconds (feed per minute mode G94 and V/D mode G93). The L word must be a whole number. You can not program L in fractions of a revolution or fractions of a second.

P – dress program number

The number entered here must be a legal program number (a program that has been saved as a subprogram or macro using the O word followed by a number of up to five digits). This dress program is used for the pre-dress operation. Refer to page 17-10 for details on the P word and it's format.

A D word (number of pre-dressing operations) is not valid in this cycle. If a D word is programmed with the dress program number, its value is ignored for the G89.1 cycle but available for any other cycles activated later in the part program that allow the D word.

Modal Parameters

Once programmed in the G89 cycle, the R plane, the two plunge distances (I and Q), the distance of the blend move (K), and the medium (E), fine (,F), and blend (,E) feedrates are modal in the G89 cycle. They need only be programmed once while in the G89 mode. These modal parameters are all cleared when the G89 mode is canceled.

Failing to Program Plunge Increment or their Feedrates

If you do not program a plunge increment or it's corresponding feedrate, they will default to zero and that plunge phase is skipped. For example if you do not program a coarse plunge distance R, then the coarse plunge phase is not performed. In this example, the rapid to R plane is extended to the next programmed plunge start point (typically I). Programming a zero for a plunge phase or feedrate also will result in that phase being skipped (except F0 which is a single digit feedrate).

ATTENTION: Failing to program a clearance plane (R) in this cycle will cause the control to skip the coarse plunge and rapid to the end point of the next programmed plunge. For example, if the medium plunge (I) is programmed but no R plane is programmed the axes will rapid to the start point of the medium plunge. Injury to the operator or damage to the machine can occur. Any plunges distances not programmed will always be performed at the feedrate of the previous plunge. If none of the plunge phase increments are programmed, the axes will rapid to the end point (shoulder point) of the cycle.

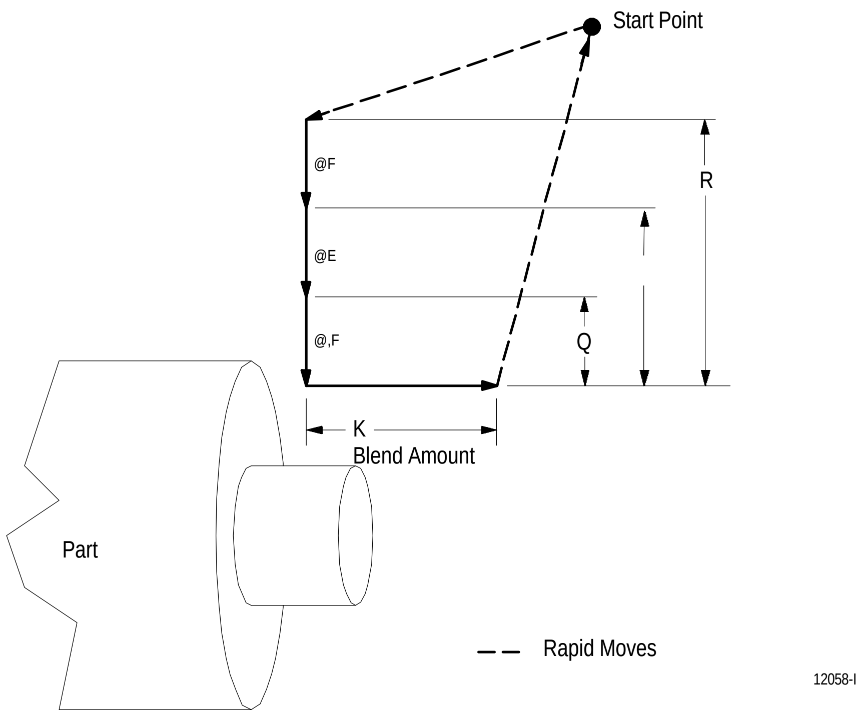
17.11.1 G89 for Normal Single-Step Grinders

Normal single-step grinders are:

- Any non-angled wheel grinder (all linear axes are perpendicular)
- Angled-wheel grinders executing in the normal angled-wheel mode (G16.3).

Figure 17.11 shows the axis motions that make up the G89 multi-step plunge with blend grinding cycle when executing in one of these grinders. This figure assumes that the ZX plane (G18) is active.

Figure 17.11
G89 Multi-Step Plunge with Blend in normal angled-wheel mode



Execution of the G89 multi-step plunge with blend cycle performs the following moves:

1. The X and Z axes travel from the start point to the R plane directly above where the plunge is programmed to take place. This move takes place at the rapid feedrate.
2. The X axis plunges to the I level at the feedrate F.
3. The X axis plunges to the Q level at the feedrate E.

4. The X axis plunges to the final depth programmed for X at the feedrate ,F. If the system installer has configured and activated a micro-feed, it takes place in this phase. Any micro-feed amount extends the plunge endpoint beyond the programmed X plunge depth. Micro-feeds are performed at the ,F feedrate.

5. If a spark-out is programmed (L word) it is performed after the X axis has completed its plunge (including any micro-feed amount).

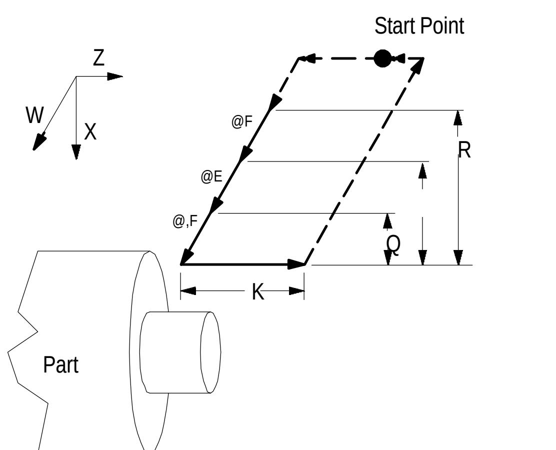
6. The Z axis performs the blend. This is programmed with the K parameter and is performed at the ,E feedrate.

7. Both axes retract at rapid feedrate to the start point.

17.11.2 G89 for Two-Step Grinders

Two-step grinders are angled-wheel grinders that are executing in the two-step angled-wheel grinding mode (G16.4). These grinders have a virtual X axis (no associated hardware) and a real Z and W axis. W is a non-orthogonal axis (it is not perpendicular to Z).

Figure 17.12
G89 Multi-Step Plunge with Blend in Two Step Mode



The above figure assumes your system installer has selected the default PAL state for two step grinding mode. The initial move from the start point and the exit move back to the start point (all rapid moves in the above figure) could be reversed. This cycle would then start with a plunge to the R plane followed by a Z axis move and end with a Z axis move followed by a rapid W move back to start point. Refer to chapter 14 for details on the different operating modes of the G16.4 mode.

Execution of the G89 multi-step plunge with blend cycle performs the following moves:

1. All Z axis motion that must occur to reach the proper Z endpoint by the end of the plunge takes place first. This move takes place at the rapid feedrate.

2. The W axis rapids to the R plane.

3. The W axis plunges to the I level at the feedrate F.

4. The W axis plunges to the Q level at the feedrate E.

5. The W axis plunges to the final depth programmed for X at the feedrate ,F. If the system installer has configured a micro-feed, it takes place in this phase. Any micro-feed amount modifies the end point of the programmed X plunge depth. Micro-feeds are performed at the ,F feedrate. Be aware that any micro-feed that occurs is not compensated for on the Z axis. This can cause overcutting (see page 17-42).

6. If a spark-out is programmed (L word) it is performed after the W axis has completed its plunge.

7. The Z axis performs the blend. This is programmed with the K parameter and is performed at the ,E feedrate.

8. The W axis retracts at rapid to the level of the start point.

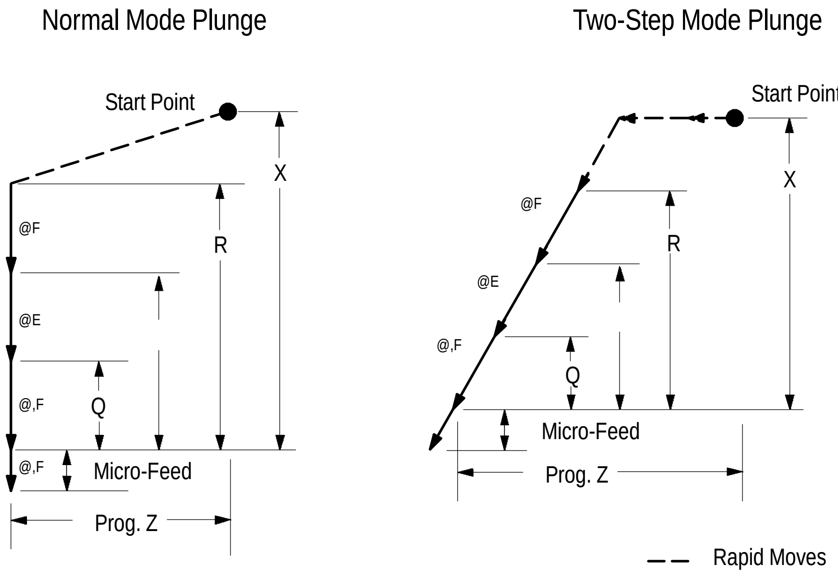
9. The Z axis returns to the start point at rapid if any micro-feed has occurred during the W axis plunge.

17.11.3 Micro-Feed During the G89/G89.1 Cycles

On some machines and processes, a substantial amount of wheel wear can occur during the execution of the G89 cycle. This can result in a part that is not within specifications. Use the micro-feed feature to correct for wheel wear that can occur between dressing operations.

Wheel wear is compensated for during the fine feed portion of the G89 cycle by adding an additional micro-feed amount to the programmed X axis plunge depth.

Figure 17.13
G89 Plunge with Micro-Feed

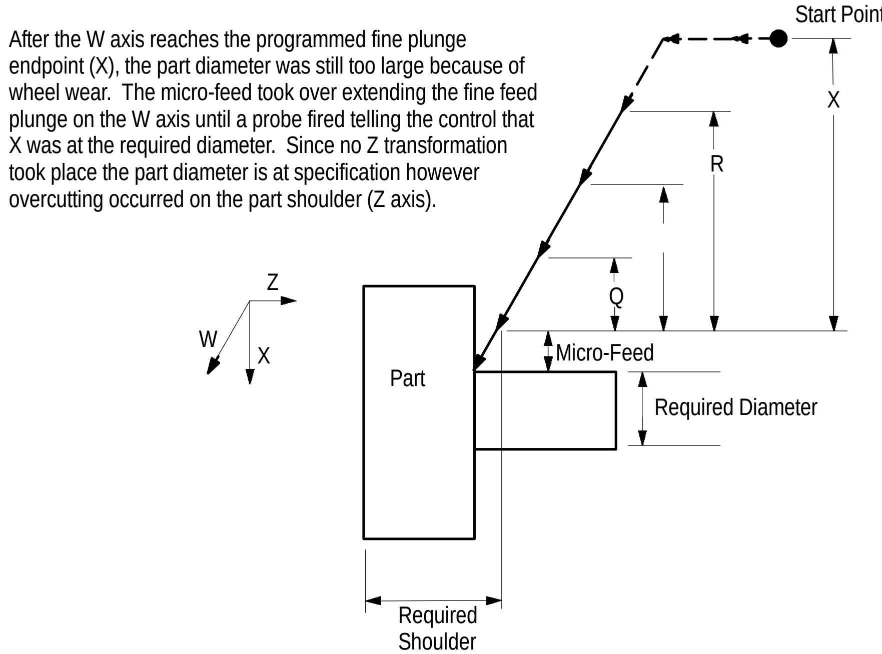


When the rapid positioning at the start of the cycle ends, the control checks to see if PAL has enabled the micro-feed feature. Refer to your system installers documentation for details on how this feature is enabled. If micro-feed is not enabled, the fine plunge stops when the programmed X axis endpoint is reached.

If the micro-feed is enabled, the fine plunge endpoint is extended beyond the programmed X plunge depth. The system installer should have devised an input to PAL that aborts the fine plunge at any point (before or after the micro-feed portion of the fine plunge). This input is called a skip signal. Refer to the system installers documentation for details on how to abort the fine plunge.

Important: If no fine feed portion of the cycle is programmed (no Q word in the cycle) then the micro feed feature can not be enabled and is not performed.

ATTENTION: Overcutting of the shoulder can occur when a micro-feed is performed on an angled-wheel grinder operating in G16.4 two step angled wheel grinder mode. All micro-feed motion in G16.4 mode occurs on the W axis. No transformation of Z axis motion occurs. This means micro-feed on the W axis can exceed the actual programmed Z axis endpoint causing overcutting of the parts shoulder.



Typically the skip signal for a fine plunge is performed by an external device measuring the current part diameter such as a gage. In this case when the actual desired plunge depth is reached, the gage fires indicating to the control to abort the fine plunge operation. The system installer determines in AMP the maximum length of the micro-feed portion of the fine feed. If no skip signal is received before the control reaches the AMP maximum micro-feed length, the control proceeds on with spark out and blend operations from this new plunge depth.

The system installer can write PAL to modify the offset tables for the new wheel diameter. In this example the control will calculate the difference between the programmed X depth and the actual X depth when the skip signal was received. This value is then subtracted from the currently active wheel diameter table value.

END OF CHAPTER

Chapter 18

Turning Operations

18.0 Chapter Overview

Turning operations generate a series of predetermined grinding/dressing motions to turn or thread a part. The major topics covered in this chapter include:

Important: We assume that your control is in the G18 plane and you define that plane as the ZX plane. If your system has a different plane active, operation of these features is different. Parameters defined here assume that Z is the first axis in the plane and that X is the second axis in the plane. If, for example, the XZ plane is the currently active plane, descriptions in this document should be interpreted accordingly (i.e., Z-axis description applies for X-axis and X-axis description applies to Z-axis). See your system installer’s documentation for details on the plane definitions on your system.

18.1 Single Pass Turning Cycles

Single pass turning cycles consist of these cycles:

- G20 - Single Pass O.D. and I.D. Roughing Cycle
- G24 - Single Pass Rough Facing Cycle

We call these cycles single pass cycles because each time the cycle is executed, it makes only one grinding pass over the workpiece or one dressing pass over the grinding wheel. Typically, single pass cycles are modal and repeat after any block that commands axis motion.

Your system installer can select in AMP to repeat the cycle after every block following the single pass cycle block. If your control is configured this way, the motions of the last executed single pass cycle repeat after every program block until you cancel the cycle.

This manual assumes that your system is configured to repeat the cycle only after blocks that command axis motion.

Cancel single pass cycles by programming a different G code in the same modal group (see G code table in Appendix C). G codes in the same group include G00, G01, G02, G03, G33, and G34.

You can program single pass cycles in diameter or radius mode. All examples in this section are given in the radius programming mode. When programming in diameter mode, remember that the value entered for the X-axis is half the actual motion for that axis. Single pass cycle examples in this manual are programmed in absolute programming mode. You can also use incremental mode.

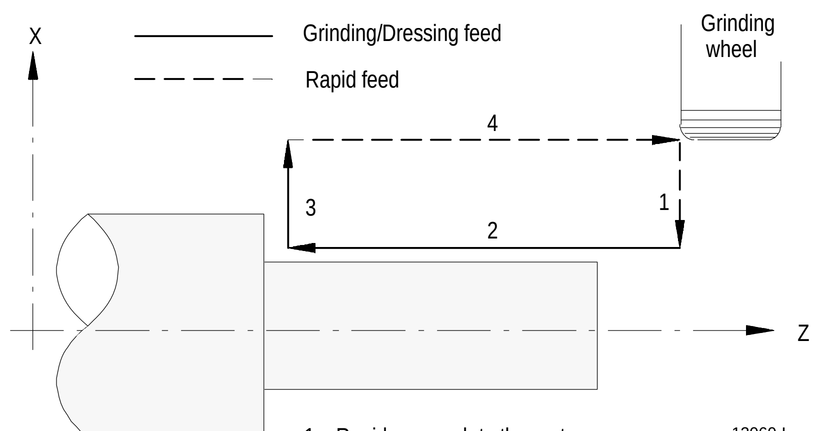
Use the G20 cycle to grind about the diameter of a part or dress the grinding wheel. Use the G24 cycle to grind on the face of a part.

18.1.1 Single Pass O.D. and I.D. Roughing Cycle (G20)

G20 calls a straight or a taper grinding/dressing cycle. This cycle is a single pass cycle; it makes only one grinding pass over the workpiece or one dressing pass over the wheel each time it is called.

Use the G20 cycle to grind along the diameter of a workpiece (in this manual that means grind parallel to the Z-axis) or to dress the grinding wheel. The G20 cycle consists of the moves shown in Figure 18.1.

Figure 18.1
G20 Straight Grinding Cycle



1 - Rapid approach to the part
2 - Grinding feed into the part
3 - Grinding feed out of the part
4 - Rapid return to the start point

ATTENTION: When programming the single pass cycle, the first move to the grinding/dressing depth is a rapid move. Make sure that the grinding wheel or dresser does not contact the part on this initial move.

The single pass cycle uses the currently active programmed grinding/dressing feedrate. You can specify a different grinding/dressing feedrate in the single pass cycle block.

Use the rapid feedrate (for the axis in motion as assigned in AMP) for the approach feedrate to the part and the return feedrate from the part to the start point.

G20 Straight O.D. and I.D. Grinding/Dressing

The format for the G20 straight grinding/dressing cycle is:

```
G20X__ Z__;
```

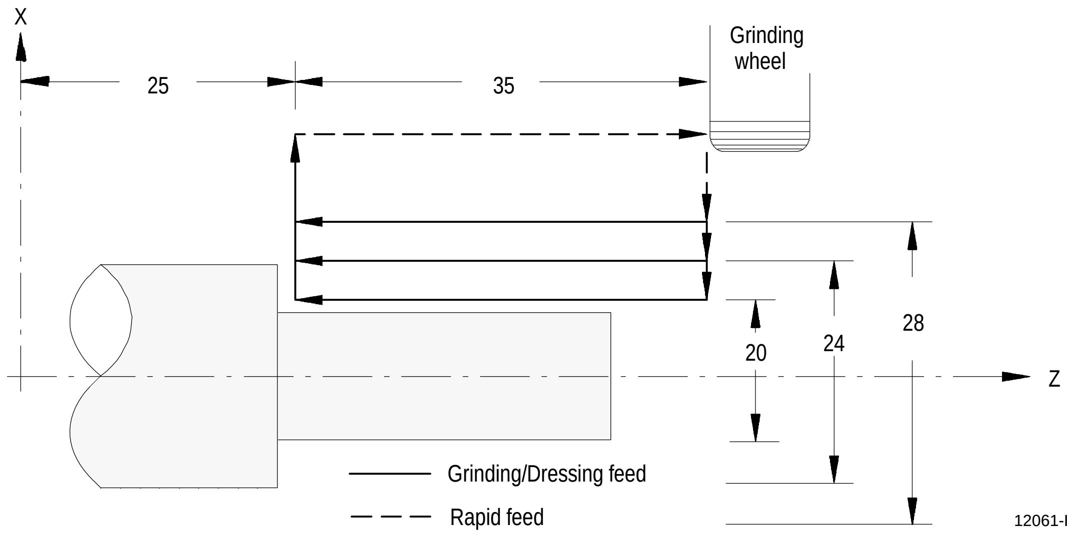
Where :	Is :
X__	the grinding/dressing depth for the X-axis. In incremental mode, specify the amount of infeed. In absolute mode, specify the coordinate position at the desired grinding/dressing depth. X can be programmed as either a diameter or radius value.
Z__	the length of grind along the Z-axis. In incremental, specify the amount of feed across the part. In absolute, specify the coordinate position of the end point of the grinding pass.

After the control executes the G20 block, the control re-executes the cycle for any following blocks that command axis motion (until you cancel the cycle). The values of the axis words in the following block replace the values of the parameters specified in the original G20 block and the cycle is re-executed using these new values.

Example 18.1
Straight Grinding Cycle

```
G08G90G00X40.Z60.;
G20X28.Z25.F10.;
X24.;
X20.;
```

Figure 18.2
Results of Example 18.1



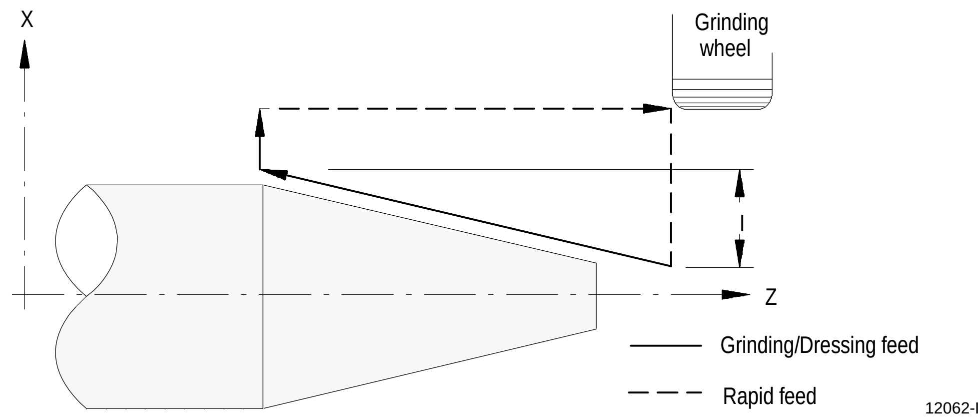
G20 Taper O.D. and I.D. Grinding/Dressing

A G20 block that includes an I word generates a grinding/dressing pass that produces a taper. The format to grind/dress a taper is:

```
G20X__Z__I__;
```

Where :	Is :
X__	the grinding/dressing depth for the X-axis at the end point of the grinding move into the part or the dressing move into the wheel. In incremental mode specify the amount of infeed, in absolute mode specify the coordinate position at the desired grinding/dressing depth. X can be programmed as either a diameter or radius value.
Z__	the length of grind along the Z-axis. In incremental mode specify the amount of feed across the part, in absolute mode specify the coordinate position of the end point of the grinding stroke.
I__	the amount of change in the grinding/dressing depth for the X-axis. I is always an incremental radius value. Figure 18.4 shows the relationship between the sign of the I value and the way that the cycle is performed. The control adds the value of I to the X depth to determine the start point of the grinding pass.

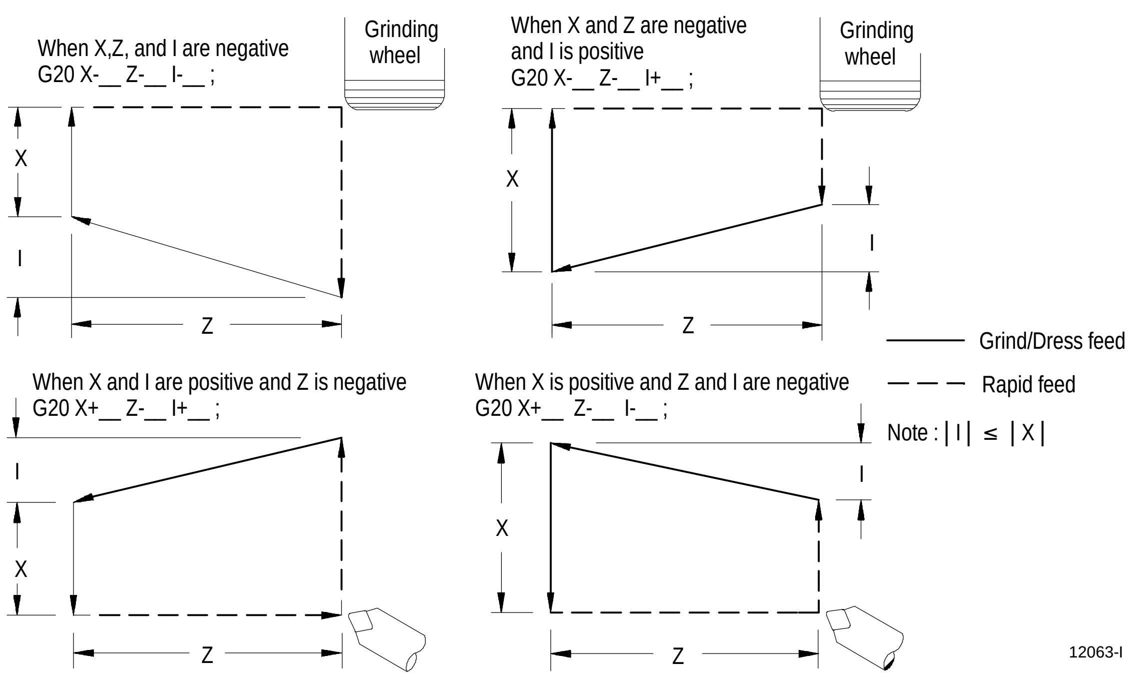
Figure 18.3
G20 Taper Grinding Cycle



After the control executes the G20 block, it re-executes the cycle for any following block that command axis motion (until the cycle is canceled). The values of the axis words in the following block replace the values of the parameters specified in the original G20 block.

Figure 18.4 applies only if you program X and Z as incremental values. If you program X and Z as absolute values, the depth of taper I is added or subtracted (depending on its sign) to the absolute X-axis position.

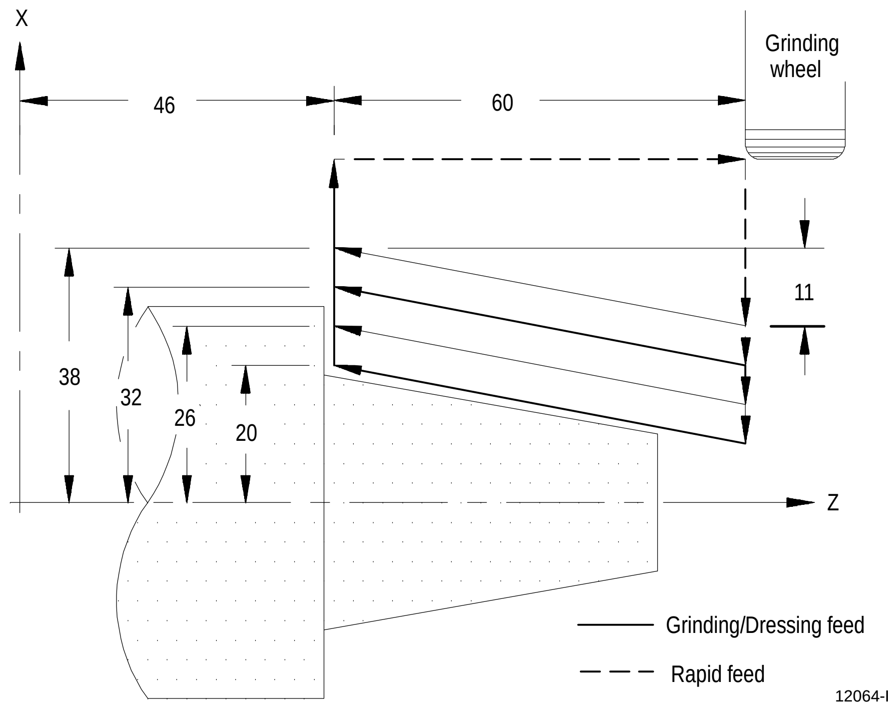
Figure 18.4
Effect of Positive and Negative Parameters in a G20 Block



Example 18.2
Taper Grinding

```
G90G00X50.Z106.;
G20X38.Z46.I-11.F.5;
X32.;
X26.;
X20.;
```

Figure 18.5
Results of Example 18.2

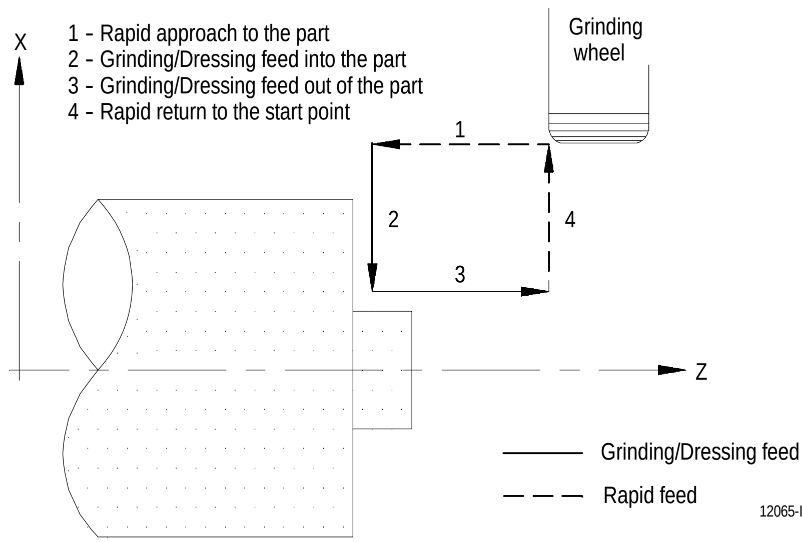


18.1.2 Single Pass Rough Facing Cycle (G24)

G24 calls either a straight or a tapered facing cycle. This cycle is a single pass cycle (makes only one grinding pass over the workpiece or one dressing pass over the wheel each time it is called).

Use the G24 cycle to grind along the face of a workpiece (in this manual that means it grinds along the X-axis). The G24 cycle consists of the moves shown in Figure 18.6.

Figure 18.6
G24 Straight Facing Cycle



ATTENTION: When programming the single pass cycle, the first move to the grinding/dressing depth is a rapid move. Make sure that the grinding wheel does not contact the part on this initial move.

The single pass cycle uses the currently active programmed grinding/dressing feedrate. You can specify a different grinding/dressing feedrate in the single pass cycle block.

Use the rapid feedrate (for the axis in motion as assigned in AMP) for the approach to the part and the return to start point.

G24 Straight Facing

The format for the G24 straight facing cycle is:

```
G24X__ Z__;
```

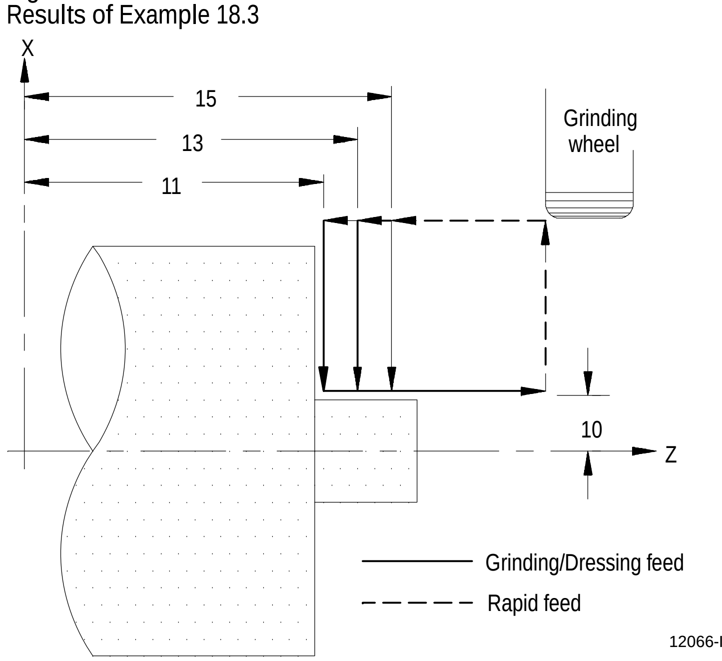
Where :	Is :
X__	the length of grind along the X-axis. In incremental mode, specify the amount of feed across the part. In absolute mode, specify the coordinate position of the end point of the grinding/dressing stroke. X can be programmed as a diameter or radius value.
Z__	the grinding/dressing depth for the Z-axis. In incremental mode, specify the amount of infeed. In absolute mode, specify the coordinate position at the desired grinding/dressing depth.

After the control executes the G24 block, the control re-executes the cycle for any following block that commands axis motion until the cycle is canceled. The values of the axis words in the following block replace the values of the parameters specified in the original G24 block and the cycle is re-executed using these new values.

Example 18.3
Straight Facing Cycle

```
G90G00X30.Z22.;
G24X10.Z15.F10.
Z13.;
Z11.;
G00;
```

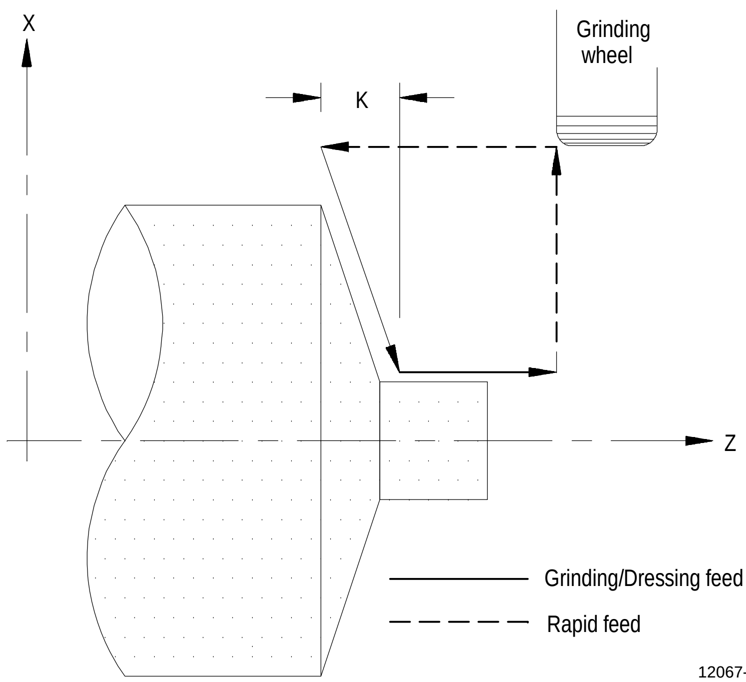
Figure 18.7
Results of Example 18.3



G24 Tapered Facing

A G24 block that includes a K word generates a facing pass that produces a taper.

Figure 18.8
G24 Face Taper Grinding/Dressing Cycle



The format for the G24 single pass cycle to grind a taper on a face is:

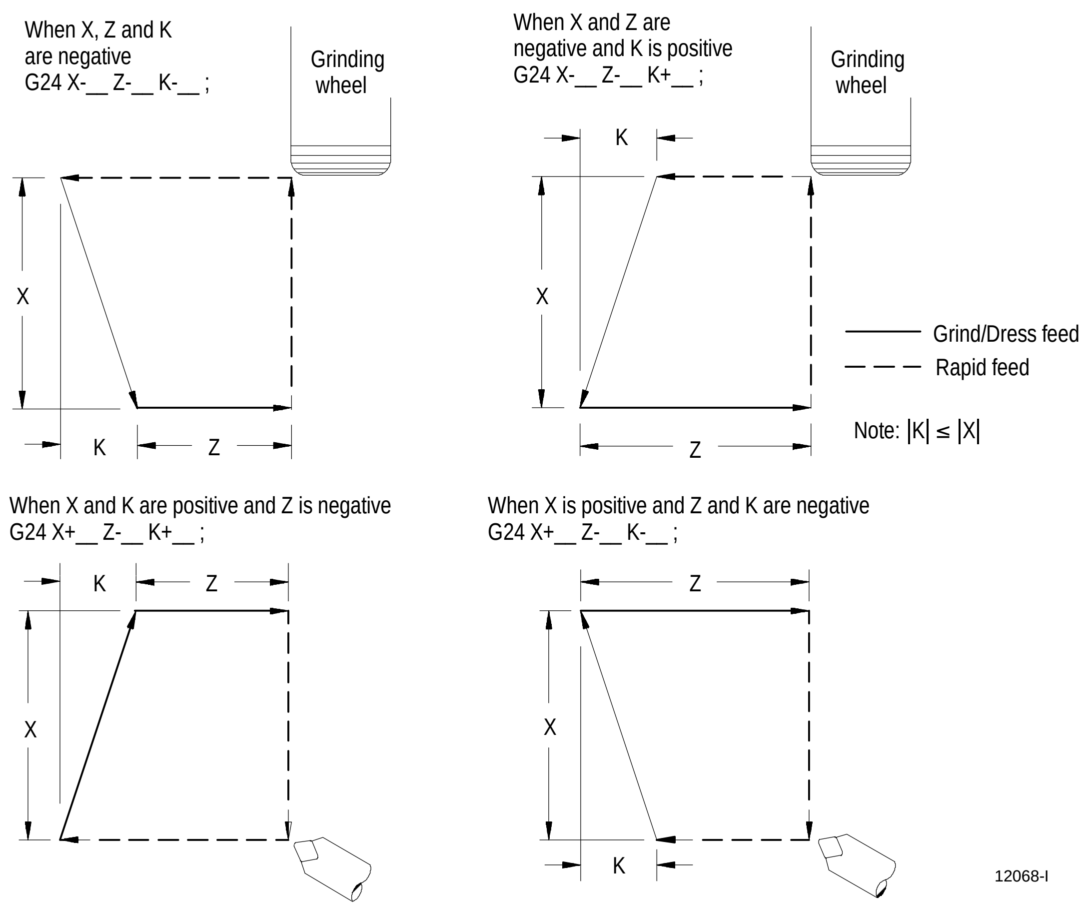
```
G24X__Z__K__;
```

Where :	Is :
X__	the length of grind along the X-axis. In incremental mode specify the amount of feed across the part, in absolute mode specify the coordinate position of the end point of the grinding/dressing stroke. X can be programmed as either a diameter or radius value.
Z__	the grinding/dressing depth for the Z-axis at the end point of the grinding/dressing move into the part. In incremental mode specify the amount of infeed, in absolute mode specify the coordinate position at the desired grinding/dressing depth.
K__	the amount of change in the grinding/dressing depth for the Z-axis. K is always an incremental value. Figure 18.9 shows the relationship between the sign of the K value and the way that the cycle is performed. The control adds the value of K to the Z depth to determine the start point of the grinding/dressing pass.

After the control executes the G24 block, the control re-executes the cycle for any following block that commands axis motion (until the cycle is canceled). The values of the axis words in the following block replace the values of the parameters specified in the original G24 block and the cycle is re-executed using these new values.

Figure 18.9 applies only if programming X and Z as incremental values. If programming X and Z as absolute values, the depth of taper K is added or subtracted (depending on its sign) to the absolute Z-axis position.

Figure 18.9
Effect of Positive and Negative Parameter Values in a G24 Block
(incremental X and X only)

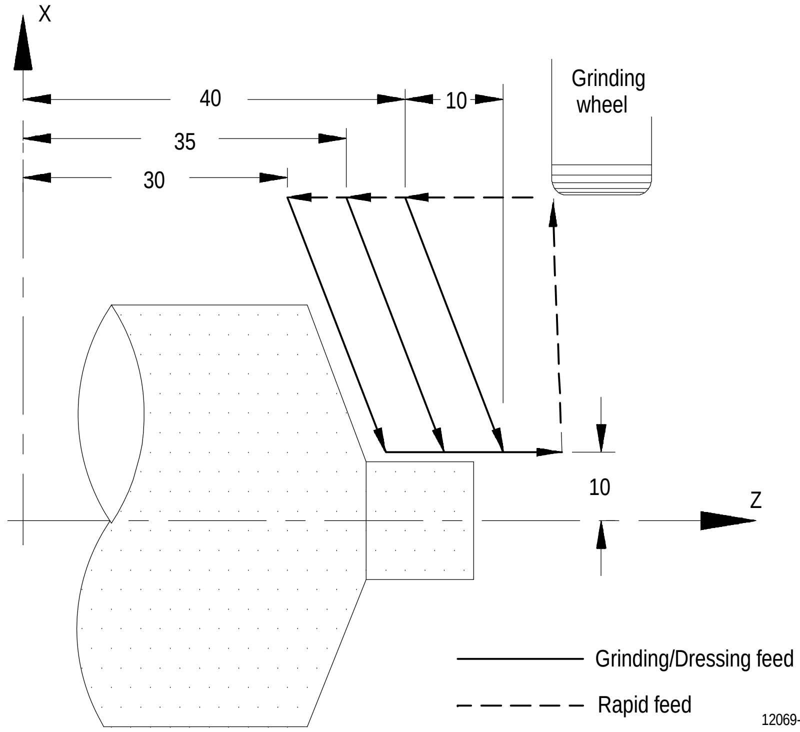


After the control executes this G24 block, the control re-executes the cycle for any following block that contains an axis word (until the cycle is canceled). The values of the axis words in the following block replace the values of the parameters specified in the original G24 block and the cycle is re-executed using these new values.

Example 18.4
Tapered Face Grinding

```
G90G00X43.Z55.;
G24X10.Z50.K-10.F10.;
Z45.;
Z40.;
G00;
```

Figure 18.10
Results of Example 18.4



18.2 Single Pass Thread Grinding

The control provides turning operations for single pass thread grinding. The single pass thread grinding operating modes are enabled by programming a G33 or a G34. G33 mode grinds straight, tapered, face, multi-start, and multi-block threads. G34 mode grinds thread passes of increasing or decreasing leads.

18.2.1 Considerations for Thread Grinding

Consider these issues when performing threading operations:

- Emergency Stop - Pressing the emergency stop during threading causes all axes to come to a rapid stop. This can cause damage to the part or grinding wheel and resumption of the threading moves is not possible.
- <CYCLE STOP> (cycle suspend) - A cycle stop does not occur if this button is pressed during a threading pass; instead axis motion continues and cycle stop is ignored.
- Overrides - During the execution of any threading pass, all feedrate overrides are fixed at 100%.
- Single Block - When performing single pass threading (G33 or G34) motion stops at the end of the threading block, typically resulting in a ringing of the thread.
- Dry Run - Whether or not the “dry run” and spindle speed override functions are operable during threading is determined by the system installer’s PAL program.
- Radius/Diameter Mode - The control performs threading in either radius or diameter modes. Radius/Diameter mode only affects the controls interpretation of the X parameter.
- Start point - Due to axis acceleration and other machine dynamics, threading passes should be programmed such that the axes have room to attain speed prior to contacting the workpiece. Failure to do so can result in the initial thread lead being incorrect.

- Axis feedrates - When threading, the speed of the grinding axis is determined by the spindle speed and the thread lead through the following equation:

```
axis feedrate = (S) / (F threads per inch)

              = (S) / (E threads per inch)

              = (S)(E inches per thread)
```

Where :	Is :
S	the actual speed of the spindle (programmed spindle speed times the spindle speed override switch setting in percent).
F	threads per inch or millimeter depending on the current active mode
E	threads per inch or inches per revolution as determined in AMP by your system installer.

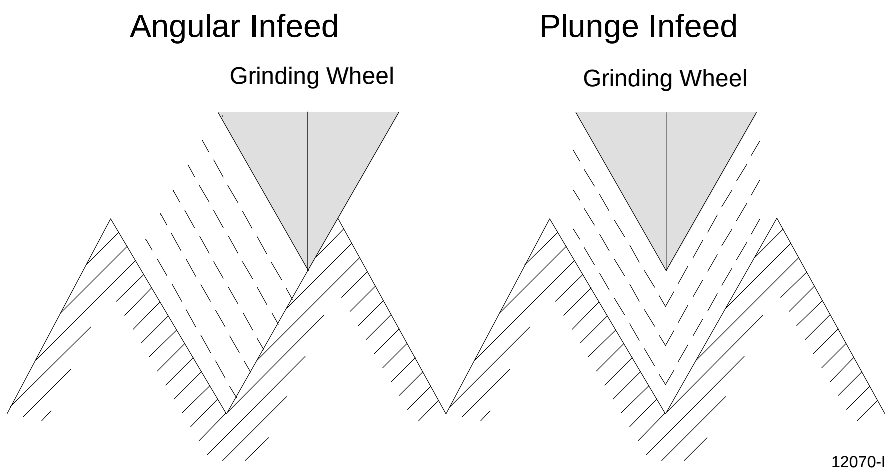
The programmer should use this equation to verify that the feedrate resulting from the thread parameters does not exceed the maximum allowable feedrate for the grinding axis. Otherwise an error results and axis motion is stopped. This equation can also be applied to face threads and tapered threads.

- Pullout angles - During threading passes, the control synchronizes the moves and speeds of the X, Z, and spindle axes. This occasionally forces the X-axis to move quite rapidly in order to produce the desired thread taper or pullout angles at the rates dictated by the active spindle speed. Compounded with the fact that many machines have X-axis feedrate limits lower than those for the Z-axis, velocity limitations can result. This is best prevented by first executing a Feed Check prior to actually grinding the threads and then reducing spindle speed or changing the pullout angle where necessary.

- Tapered Thread Lead - When grinding a tapered thread, the thread lead (E or F word) is applied to the axis that travels the greatest distance from the start to the end of the threading pass.

- Infeed - Plunge infeed relies on a grinding wheel dressed at the exact thread angle so that two sides of the grinding wheel are grinding as it is fed perpendicular to the work on successive passes. On larger threads, this type of infeed can cause vibration. In that case, angular infeed can be preferred. This results in the grinding wheel being fed along the thread flank with each successive pass, meaning only one side of the grinding wheel grinds. Figure 18.11 illustrates plunge and angular infeed.

Figure 18.11
Angular versus Plunge Infeed

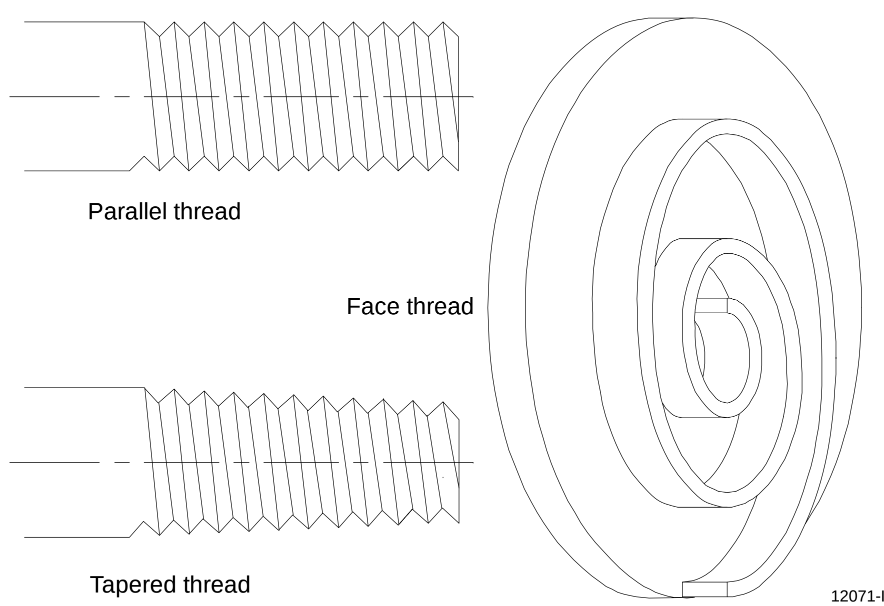


When threading, you must program a small Z move to generate an angular infeed.

18.2.2 Single Pass Threading Mode (G33)

The G33 single pass thread grinding mode grinds straight, tapered, face, and multi-start threads that have constant thread leads (use G34 to grind threads that do not have a constant lead). The G33 thread grinding mode is a mode, not a cycle and does not generate any extra motion blocks. This mode synchronizes the thread grinding motion with the spindle to allow programming multiple passes over the same threads.

Figure 18.12
Constant Lead Threads

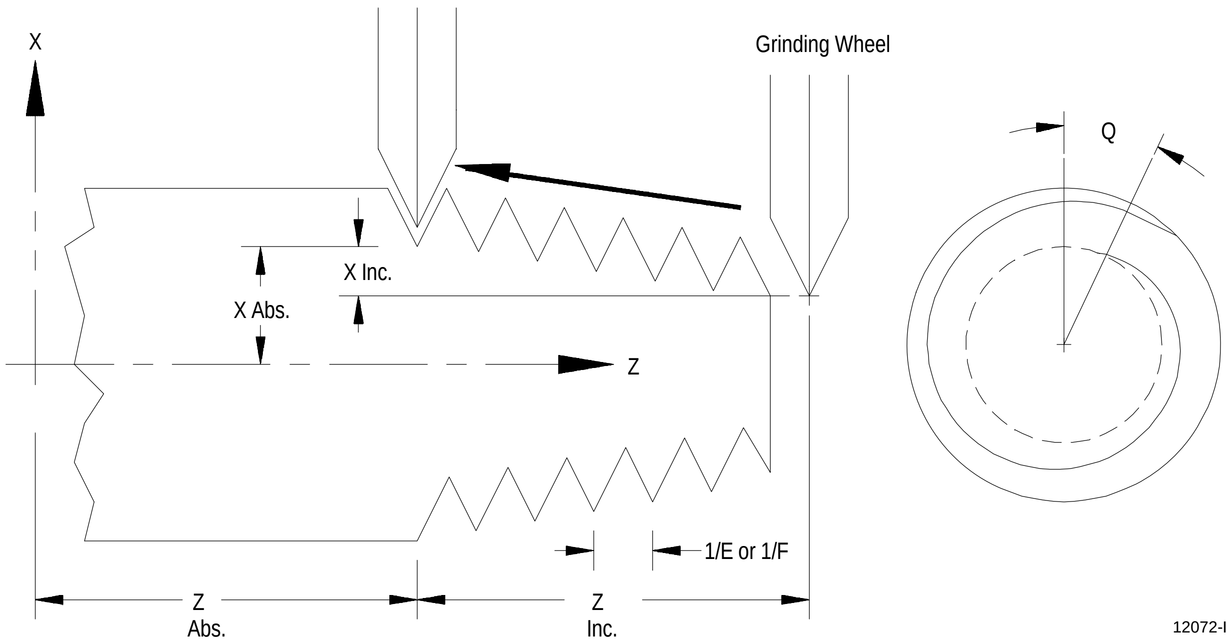


The format for the G33 thread grinding operation is:

```
Parallel thread  G33Z__ { F__ } Q__;
                        { E   }

Tapered thread   G33X__Z__ { F__ } Q__;
                           { E   }

Face thread      G33X__ { F__ } Q__;
                        { E   }
```

Where :	Is :
X	the end point of the thread grinding move in the X-axis. This parameter can be an incremental or absolute and radius or diameter value. If not present there must be a Z parameter. If an X parameter is present, it indicates either a face, tapered, or lead-in thread. When used in a G33 block without a Z parameter, a facing thread is made parallel to the X-axis at the Z-axis position prior to the G33 block. X values can be entered as a radius or a diameter value. X can also be programmed as an incremental or absolute value. The initial minor diameter of any straight or tapered thread is determined by the position of the X-axis prior to the G33 block.
Z	the end point of the thread grinding move in the Z-axis. This parameter can be an incremental or absolute value. If not present there must be an X parameter. When using a Z parameter in a G33 block without an X parameter the threading pass is made parallel to the Z-axis at whatever X position the grinding wheel edge was at prior to the G33 block. Z parameter is always entered as a radius values regardless of the current mode.
E F	This parameter may be entered by using either an E- or F-word. It represents the thread lead along the axis with the largest programmed distance to travel to make the thread cut. It is mandatory when cutting any threads. If the E-word is programmed, its value (sign ignored) is equal to the number of threads per inch or inches per thread (determined in AMP) regardless of whether inch or metric mode is active at the time. If the F-word is programmed, its value (sign ignored) is the thread lead in inches per revolution or millimeters per revolution, depending on the mode in which the control is operating.
Q	an optional parameter that provides a relative value for the start offset angle of the thread. Its primary use is in grinding multi-start threads. For example, if a threading pass were made with a value of zero here, and then followed by another pass with a value of 180 then the second pass is started 180 degrees from the first resulting in a two start thread. If two more passes are then made, one with a parameter value of 90 and one with a value of 270, the result would be a four start thread.

Figure 18.13
G33 Block Parameters

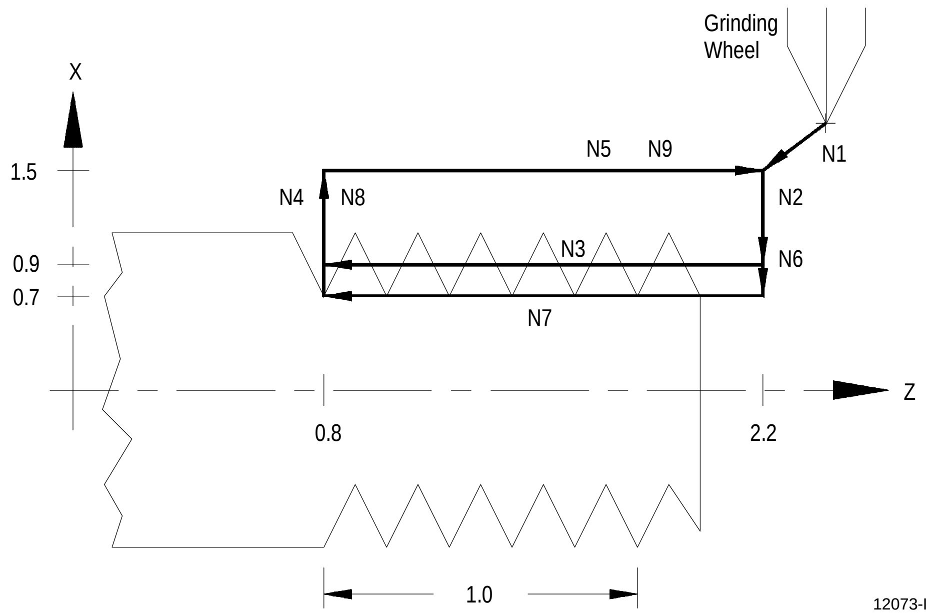


Example 18.5
Parallel Thread Grinding

```
Thread lead:    5 threads/inch (.20 inch pitch)

Depth of grind:       .1 pitch

Number of grinding passes:  2
```

```
N1G00X1.5Z2.2;
N2X.9;
N3G33Z.8E5.;
N4G00X1.5;
N5Z2.2;
N6X.7;
N7G33Z.8E5.;
N8G00X1.5;
N9Z2.2;
```

Figure 18.14
Results of Parallel Thread Grinding Example 18.5

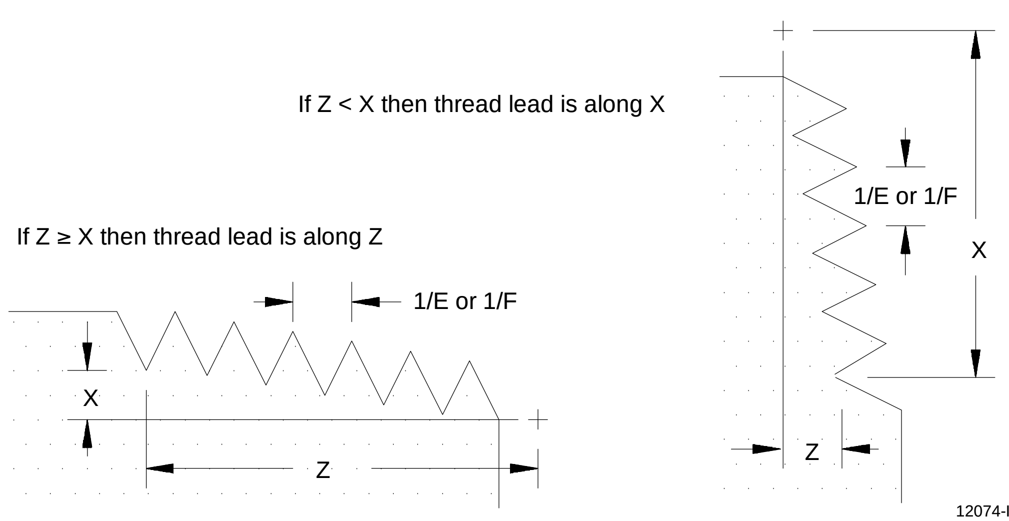


If you program both E and F in the same block, the right-most parameter takes effect for that block.

The programmed lead remains in effect until another thread lead value is programmed, the control is reset, or an M02 or M30 end of program block is executed.

For tapered threads, the thread lead (determined by the F or E word) is applied along the axis that travels the greatest distance when grinding the thread. See Figure 18.15.

Figure 18.15
Lead Designation for Tapered Thread



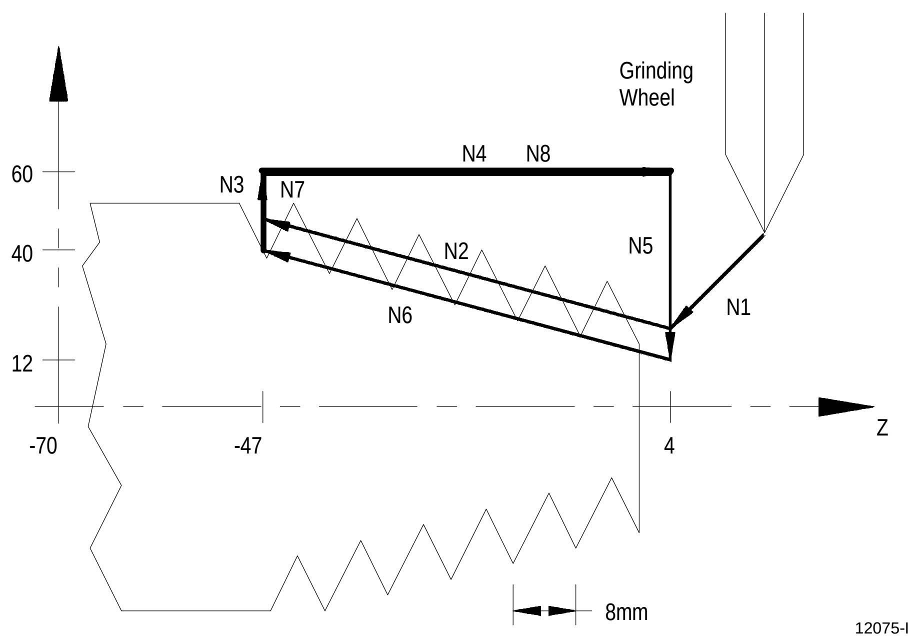
When using the X-axis as the thread lead axis for E or F, program thread leads as radial values.

Example 18.6
Tapered Thread Grinding

```
Thread lead: .125 threads/mm (8 mm pitch)

Depth of grind:  1 mm (X direction)

Number of grinding passes:  2
```

```
N1G77G00X20.Z4.;
N2G33X48.Z-47.F8;
N3G00X60.;
N4Z4.;
N5X12.;
(second pass)
N6G33X40.Z-47.;
N7G00X60.;
N8Z4.;
```

Figure 18.16
Results of Tapered Thread Grinding Example 18.6

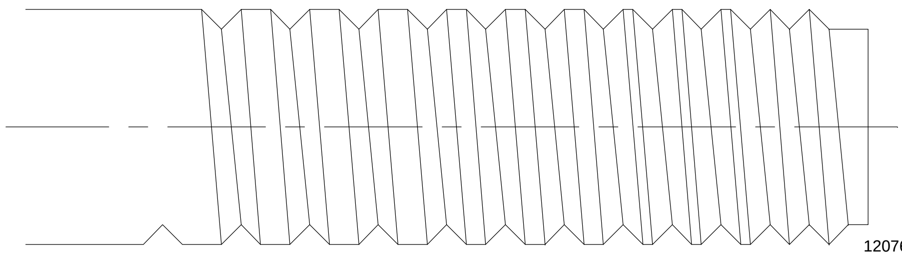


You can program multiple-thread grinding by assigning a thread grinding start shift angle using a Q word. Omission of a Q word indicates a shift angle of 0.

18.2.3 Single Pass Variable Lead Threading Mode (G34)

The G34 single pass variable lead thread grinding mode grinds straight, tapered, face, and multi-start threads that do not have a constant thread lead. It is programmed almost identically to the G33 thread grinding mode with the addition of a K word used to program the amount of lead variation per revolution.

Figure 18.17
Variable Lead Thread



The format for the G34 threading mode is:

```
Parallel thread  G34Z__ { F__ } Q__K__;
                        { E   }

Tapered thread   G34X__Z__ { F__ } Q__K__;
                           { E   }

Face thread      G34X__ { F__ } Q__K__;
                        { E   }
```

Where :	Is :
X	the end point of the thread grinding move in the X-axis. This parameter can be an incremental or absolute and radius or diameter value. If not present there must be a Z parameter. If an X parameter is present, it indicates either a face, tapered, or lead-in thread. When used in a G34 block without a Z parameter, a facing thread is made parallel to the X-axis at the Z-axis position prior to the G34 block. The initial minor diameter of any straight or tapered thread is determined by the position of the X-axis prior to the G34 block.
Z	the end point of the thread grinding move in the Z-axis. This parameter can be an incremental or absolute value. If not present there must be an X parameter. When using a Z parameter in a G34 block without an X parameter the threading pass is made parallel to the Z-axis at whatever X position the grinding wheel edge was at prior to the G34 block. Z parameters are always entered as a radius values regardless of the current mode.

Where :	Is :
E F	This parameter may be entered by using either an E- or F-word. It represents the thread lead along the axis with the largest programmed distance to travel to make the thread cut. It is mandatory when cutting any threads. If the E-word is programmed, its value (sign ignored) is equal to the number of threads per inch or inches per thread (determined in AMP) regardless of whether inch or metric mode is active at the time. If the F-word is programmed, its value (sign ignored) is the thread lead in inches per revolution or millimeters per revolution, depending on the mode in which the control is operating. In a G34 block, E or F indicates the initial thread lead used at the start of the threading pass.
Q	an optional parameter that provides a relative value for the start offset angle of the thread. Its primary use is in grinding multi-start threads. For example, if a threading pass were made with a value of zero here, and then followed by another pass with a value of 180 then the second pass would start 180 degrees from the first resulting in a two start thread. If two more passes are then made, one with an a parameter value of 90 and one with a value of 270, the result would be a four start thread.
K	the difference in the thread lead per spindle revolution (inch/rev/rev or mm/rev/rev). The amount of K is added to the thread lead (E or F) after each thread is ground. K can be programmed as a positive (increasing thread lead) or a negative (decreasing thread lead) value.

The actions of the G34 variable lead threading operation is identical to the G33 threading operation with the exception of the variable thread lead. See the G33 threading section, beginning on page 18-14, for details and examples of single pass threading blocks that grind parallel, tapered, or face threads.

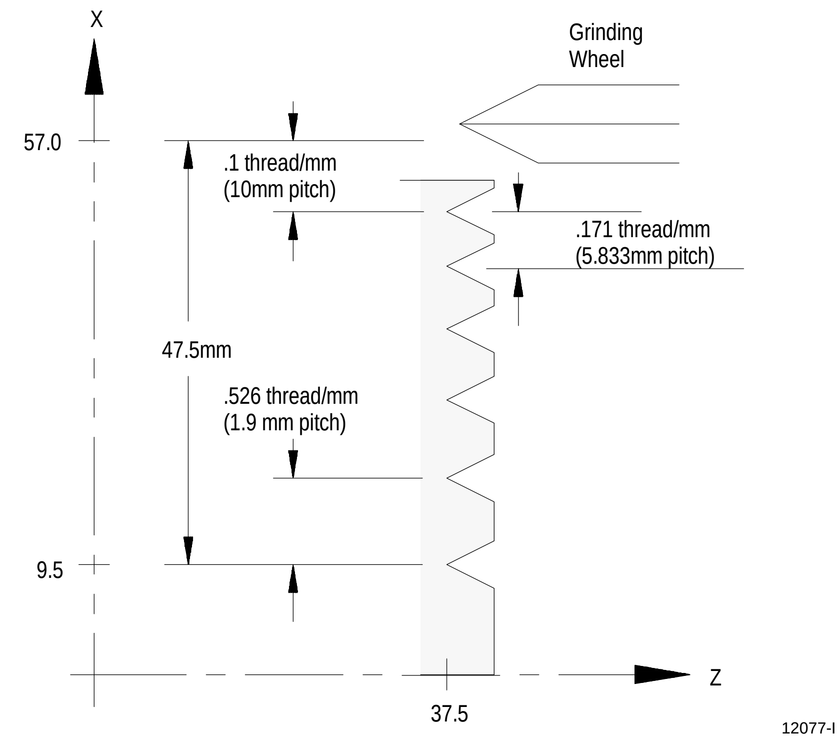
Metric and inch lead variation limits are indicated below:

```
+/- 0.0001  to  +/- 100.0000 mm/rev

+/- 0.000001 to  +/- 1.000000 inch/rev
```

Example 18.7
Variable Lead Face Threading Using G34

```
N1G00G07X57.Z37.5F100;
N2G91;
N3G34X-47.5F.1K.071;
N4G00Z10.;
N5X47.5;
```

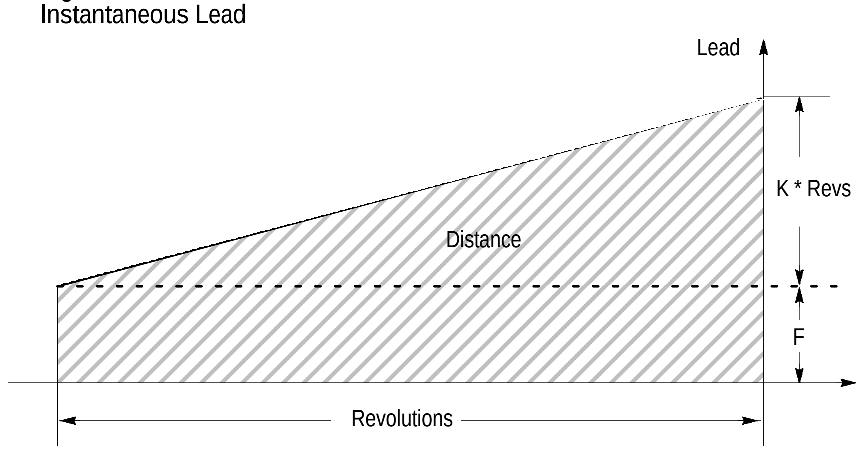
Figure 18.18
Results of Variable Lead Face Threading Example 18.7



The lead changes continuously during the move. At any point during the move, you can calculate the lead with this formula:

```
instantaneous lead = F + (K * number of revs since the start)
```

Figure 18.19
Instantaneous Lead



END OF CHAPTER

Chapter 19

Skip and Gauge Probing Cycles

19.0 Chapter Overview

External skip functions are motion-generating G-code blocks that can be aborted when the control receives an external signal through the PAL program. Gauging functions are similar to the external skip functions except that you can use the axis coordinates (at the time the external signal is received) to modify the wheel offset table. The major topics covered in this chapter include:

Important: You can also enable the G04 dwell feature as an external skip or wheel gauging command. For details on programming a G04, see chapter 12.

The control provides several means of triggering an external skip or gauging block:

- discrete inputs through the I/O ring
- any one of the four available "High Speed Inputs"
- a "Probe" input that directly latches the feedback counters

You can use these different inputs, each with different degrees of precision, to signal the control to store the current axes positions. See documentation prepared by your system installer for details on your specific machine.

These conditions must be satisfied when you execute an external skip or gauging block:

- you must disable dresser/wheel radius compensation (G40 mode) when the control executes the block
- the block that contains the external skip G code (excluding G04 as external skip) must be a linear block

ATTENTION: We do not recommend using a skip block from any fixed cycle block (such as multi-pass face grinding or a turning). 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.

19.1 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. Your system installer determines what signal (such as a touch probe, manual switch, etc.) corresponds to each G31 code in PAL. Your 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, you can also use a G04 dwell in seconds as an external skip function.

The format for G31 external skip blocks is:

```
G31 X__ 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, Z	the endpoint of the move if no external skip signal is received. These also determine the direction that the wheel travels in.
F	the external skip function feedrate. If no value is entered here, the external skip function executes 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 produces 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 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.

Your system installer defines probe tip radius in AMP. You can also change this value through the paramacro system parameter #5096. See chapter 20 for details on paramacro parameters.

Skip Function Application Example

A typical application for these G codes would be to mount the probe as if it were a wheel. When the probe contacts the part and triggers, coordinate data would be available in the paramacros for use in the remainder of the part program.

The probe tip radius would be significant for this application.

19.2 Wheel Gauging External Skip Functions (G37 Codes)

Wheel gauging functions are similar to external skip functions. The difference is that the wheel gauging functions use the actual wheel position (when the external skip signal is received) to enter values in the wheel offset table for the currently active offset.

Use wheel 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. Your system installer determines what signal (such as a touch probe, manual switch, etc.) corresponds to each G37 code in PAL. Your 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.

The format for any G37 skip block is:

```
G37 Z__ F__;
```

Where :	Is :
G37	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, Z	the axis on which the length offset measurement is to be taken, specified here as either X or Z. Only one axis can 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.
F	the wheel gauging external skip function feedrate. If no value is entered here, the external skip function executes 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.

Important: You cannot use the G37 series G codes to modify the wheel radius offset values. You can only modify wheel length offset values.

The target offset value for these gauging functions is determined by the currently active wheel offset number.

ATTENTION: If modifying a wheel length offset, the offset value generated with this gauging function is immediately loaded into the offset table. Since this offset must be the currently active offset, it becomes effective immediately when the next block is executed or is delayed until the next block that contains motion on the wheel length axis is executed (when an offset is activated is determined in AMP by your system installer).

The G37 series of G codes always produces 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).

Your 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 wheel enters or after the wheel exits the position tolerance range, a PROBE ERROR occurs. This error appears on the screen as a warning but does not place the control in E-STOP. Instead, the control aborts the G37 block and proceeds program execution to the next block. No modification of the wheel offset table is performed.

Important: The move that immediately follows a G37 series skip block cannot be a circular move.

Your system installer determines in AMP if the new value is added to or replaces the old value in the table. Your system installer also determines in AMP which gauge functions alter which wheel geometry or radius offset tables.

The control automatically compensates for probe radius and length when calculating wheel offset changes if these probe parameters have been entered.

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 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.

Your system installer defines probe tip radius and probe length in AMP. You can also change these values through the paramacro system parameters #5096 (for radius) and #5095 (for length). See chapter 20 for details on paramacro parameters.

Wheel Gauging Application Example

A typical application for these G codes in determining wheel length offsets would execute as follows:

1. When the control executes the G37 block, the wheel is moved towards the triggering device 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 wheel length offset table for the currently active wheel offset number.

Figure 19.1
Typical Wheel Gauging Configurations

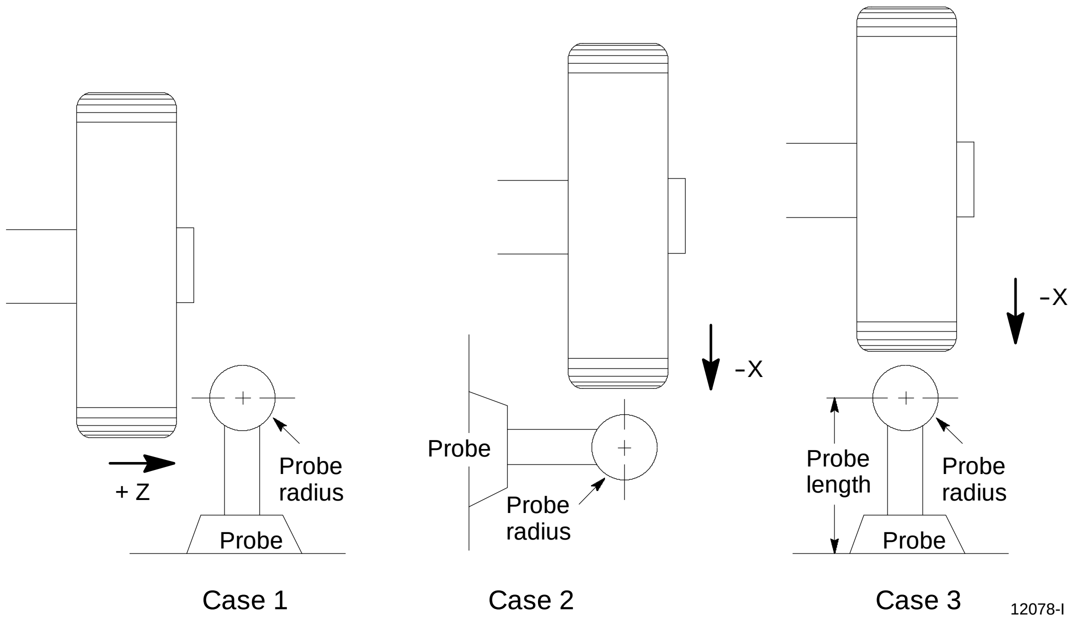


Figure 19.1 illustrates 3 typical wheel gauging configurations. All 3 cases assume that the probe is at a known fixed point on the machine.

In case 1, the Z-axis wheel length offset is being gauged, while in case 2, the X-axis wheel length offset is being gauged. In both cases:

- only the probe tip radius is significant to the control in calculating the offset adjustment
- the reference position is the center of the probe ball

In case 3, the X-axis wheel length offset is being gauged and both the probe radius and the probe length are significant to the control's offset adjustment calculations. In this case, the reference position is the bottom of the probe.

Important: We do not recommend the wheel gauging configuration case 3 depicted in Figure 19.1 due to the risk of probe damage.

END OF CHAPTER

Chapter 20

Paramacros

20.0 Chapter Overview

Paramacros are similar to subprograms, with many added features. Use paramacros to create custom cycles that may require complex mathematical calculations, access to wheel offset, work coordinates, wheel position data, and the ability to alter normal program execution. The major topics covered in this chapter include:

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 stated otherwise. The only restriction is that no other program commands, except other paramacro commands, can exist in a block that contains paramacro commands. Macro and non-macro commands cannot exist in the same program block.

20.1 Parametric Expressions

This section contains the following subsections:

Parametric expressions are mathematical or logical expressions that are evaluated in a paramacro. The following subsections cover 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.

20.1.1 Basic Mathematical Operators

This subsection covers the basic mathematical operators that are available on the control. Use these operators to accomplish mathematical operations necessary to evaluate basic mathematical equations, such as addition, multiplication, etc. Table 20.A lists the basic mathematical operators and their meanings.

Table 20.A
Mathematical Operators

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
[ ]	Brackets
OR	Logical OR
XOR	Logical Exclusive OR
AND	Logical AND
MOD	Modulus

The order of execution for the above operations is:

1. The control evaluates any part of the expression that is between the brackets [].
2. The control evaluates multiplication, division, and MOD.
3. The control evaluates all other operations.

If you perform the same level of evaluation, the left-most operation takes priority. Example 20.1 illustrates the order of execution for the operators found in Table 20.A.

Example 20.1
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 parameter with a value assigned to it
- B cannot be negative or an error occurs
- 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 20.2 illustrates the proper format for the logical operators.

Example 20.2
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

20.1.2 Mathematical Function Commands

This subsection covers the basic mathematical functions that are available on the control and their usage. Use these functions to accomplish mathematical operations necessary to evaluate trigonometric and other complex mathematical equations, such as rounding off, square root, logarithms, exponent, etc. Table 20.B lists the basic mathematical functions that are available and their meanings.

Table 20.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 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 20.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 20.3 illustrates the format for the mathematical functions found in Table 20.B.

Example 20.3
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 20.4 illustrates the results obtained when using the functions in Table 20.B.

Example 20.4
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[12.49]	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: Take precautions when performing calculations within the brackets [] following a mathematical function. The operations within the bracket are performed first, then the mathematical function is performed on the result of the operation within the brackets. See Example 20.5.

Example 20.5
Precaution for Order 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

The values composing parameter #3 are added together first and then rounded, not rounded and then added together.

20.1.3 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.

20.2 Transfer of Control Commands

This section contains the following subsections:

Use transfer of control commands to alter the normal flow of program execution. Normally the control executes program blocks sequentially. 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 access 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 uses the first block it encounters with the correct N number. The control searches for an N number starting at its current position in the program and continuing to the end of the program. If the N number is not found, the control returns to the top of the program and continues the search.

Two types of transfer of control commands are available:

- Conditional – The control executes a jump or a loop depending on whether a mathematical condition is true
- Non-Conditional – The control executes a jump or loop when the control executes that block

20.2.1 Conditional Operators

This subsection covers 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. The use of conditional operators in "IF" or "WHILE" commands is covered on pages 20-9 and 20-11.

The true or false condition determines whether the "IF" or "WHILE" blocks are executed. Table 20.C lists the conditional operators available for paramacro programming.

Table 20.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

A condition is programmed between the [and] brackets in the following format:

[A EQ B]

where:

- A and B represent some numerical value
- the values for A and B are in the form of some mathematical equation or in the form of a paramacro parameter

Example 20.6 illustrates the use of the conditional operators found in Table 20.C.

Example 20.6
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, see page 20-9 for details on “IF” statements and page 20-11 for details on “WHILE” statements. For details on the use of paramacro parameters, see page 20-12.

20.2.2 GOTO and IF-GOTO Commands

Unconditional GOTO

Use the unconditional GOTO command to automatically transfer control any time that the GOTO block is executed.

The format for the GOTO command is:

```
GOTO n;
```

Where:	Specifies:
n	the sequence number of the block to which execution is transferred when the GOTO block is executed.

Example 20.7
Un-conditional GOTO

```
N1...;
N2...;
N3GOTO5;
N4...;
N5...;
N6...;
/N7GOTO1;
```

In Example 20.7, execution continues sequentially until block N3 is read. Execution then transfers to block N5 and again resumes sequential execution to block N6. If optional block skip 1 is off, block N7 transfers execution back to block N1.

Conditional IF-GOTO

The conditional IF-GOTO command is dependent on whether a mathematical condition is true. If the condition is true, execution transfers to the block specified.

Format for the IF-GOTO command is:

```
IF [(condition)] GOTO n;
```

Where:	Specifies:
(condition)	some conditional expression (see page 20-7for details). This condition is tested by the control to determine if it is true or false.
n	the block to which execution is transferred if the condition is tested as true.

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 20.8 illustrates the use of the conditional IF-GOTO command.

Example 20.8
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, blocks N3 and N4 are skipped and execution continues on from block N5. If the comparison is false, execution continues on to block N3. When block N6 is read, parameter #4 is compared to the value 3. If the comparison is true, execution is transferred to block N1. If false, execution continues on to block N7.

20.2.3 DO-END and WHILE-DO-END Commands

Use the DO-END and WHILE-DO-END commands to set up complex looping structures in your part program. The format for these commands is shown below.

Unconditional DO-END

The unconditional DO-END command is rarely used. The lack of a condition here causes the control to loop indefinitely, until you press <CYCLE STOP> or <E-STOP> or until some other transfer of control command forces execution out of the loop.

The format for the UNCONDITIONAL DO-END command is:

```
DO m;
:
:
:
END m;
```

Where :	Is :
m	an identifier used to relate a DO block with an END block. The value of m must be the same for the DO and 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 transferred to some block out of the loop or stopped by some external operation such as pressing <CYCLE STOP> or <E-STOP>.

Conditional WHILE-DO-END

The conditional WHILE-DO-END command is dependent on whether a mathematical condition is true. If this condition is false, execution transfers to the block immediately following the END statement block.

The format for the WHILE-DO-END command is:

```
WHILE [ (condition) ] DO m;
;
;
;
END m;
```

Where :	Is :
(condition)	some condition expression (see page 20-7for details). 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 and 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 transferred to the block immediately following the END statement block.

Example 20.9 illustrates the use of the WHILE-DO-END block.

Example 20.9
WHILE-DO-END Program Segment

```
N1 #1=1;
N2WHILE[#1LT10]DO1
N3#1=[#1+1];
N4...;
N5...;
N6END1;
N7...;
```

In Example 20.9, 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. We defined nesting 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 20.10 illustrates the use of nested WHILE-DO-END commands.

Example 20.10
Nested WHILE DO Commands

```
N1#1=1;
N2WHILE[#1LT10]DO1;
N3#1=[#1+1];
N4WHILE[#1EQ2]DO2;
N5...;
N6END2;
N7END1;
N8...;
```

In Example 20.10, blocks N2 through N7 are repeated until the condition in block N2 becomes false. Within DO loop 1, DO loop 2 is repeated until the condition in block N4 becomes false.

20.3 Parameter Assignments

This section contains the following subsections:

Topic:	On page:
Local Parameters Assignment	20-13
Common Parameters	20-15
System Parameters	20-16
PAL Parameters	20-37

The following subsections cover assigning different parameter values and how to use these parameters in a paramacro. Paramacros use parameters in place of numeric values. Parameters are used as variables.

There are 4 types of parameters that you can call 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

The following subsections cover these different types of parameters independently. This does not mean that they are not interchangeable in the same macro program. Mixing the different types of parameters in the same paramacro is acceptable.

20.3.1 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 20-39.

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 M98. 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 (M99, M02, or M30).

Considerations for local parameters

You must consider the following when assigning values to local parameters:

- All local variable assignments are reset to zero any time the control reads an M02, or M30 in a part 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 20.H (B) for the parameter assignment

Example 20.11 through Example 20.13 illustrates local parameter assignments.

Example 20.11
Assigning Using More Than One I, J, K Set

```
G65P1001K1I2J3J4J5;
```

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 20.12
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 value 3.1, assigned to parameter #18 by the 1st R word, is replaced by the value -0.5, assigned to the 2nd R word.

Example 20.13
Assigning The Same Parameter Twice Using I, J, and K

```
G65P1001R2I3.4D5I-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 value 5, assigned to parameter #7 by the D word, is replaced by the value -0.6, assigned to the second I word.

20.3.2 Common Parameters

The common parameters refer to parameters 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. They are assigned through the use of a common parameter table as described on page 20-44.

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 into two types:

- Unsaved common parameters – 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
- Saved common parameters – 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.

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 PAL reference manual.

20.3.3 System Parameters

You can use system parameters in any part program, including paramacros and subprograms. All of these parameters can be used as data or can be changed by assignment (read and write) unless indicated differently in Table 20.D. System parameters are generated by the control and can be modified by operation or programming.

Table 20.D
System Parameters

Parameter #	System Parameter	Page
2001 to 2732	Dresser/Wheel Offset Tables	20-18
3000	[2] Program Stop With Message (PAL)	20-18
3001	System Timer (PAL)	20-19
3002	System Clock	20-19
3003	[2] Block Execution Control 1	20-19
3004	[2] Block Execution Control 2	20-20
3006	[2] Program Stop With Message	20-21
3007	[1] Mirror Image	20-22
4001 to 4120	[1] Modal Information	20-22
5001 to 5012	[1] Coordinates of End Point	20-23
5021 to 5032	[1] Coordinates of Commanded Position	20-24
5041 to 5052	[1] Machine Coordinate Position	20-24
5061 to 5069 or 5541 to 5552	[1] Skip Signal Position (Work Coordinate)	20-25
5071 to 5079 or 5561 to 5572	[1] Skip Signal Position (Machine Coordinates)	20-26
5081 to 5089 or 5581 to 5592	[1] Active Tool Length Offsets	20-27
5095 to 5096	[1] Probe Length and Radius	20-28
5101 to 5112	[1] Current Following Error	20-29
5201 to 5212	External Offset amount	20-29
5221 to 5232	G54 Work Coordinate Table Value	20-30
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	

5500 to 5509	In-Process Dresser Parameters	20-32
5600 to 5625	Part Program Block Create through PAL Display Pages	20-33

Table 20.D
System Parameters (continued)

Parameter #	System Parameter	Page
5630	[1] S-Curve Time per Block	20-33
5661 to 5642	[1] Acceleration Ramps for Linear Acc/Dec Mode	20-34
5651 to 5662	[1] Deceleration Ramps for Linear Acc/Dec Mode	20-34
5671 to 5682	[1] Acceleration Ramps for S-Curve Acc/Dec Mode	20-35
5691 to 5702	[1] Deceleration Ramps for S-Curve Acc/Dec Mode	20-35
5711 to 5722	[1] Jerk	20-36
5731 to 5743	Home Marker Distance	20-36
5751 to 5763	Home Marker Tolerance	20-37

[1] These parameters may only have their value received (read-only)
[2] These parameters may only have their value changed (write-only)

Pages 20-18 through 20-33 contain descriptions of the system variables.

Important: Angled wheel grinders that have a virtual axis can access system parameters for the virtual axis. The virtual axis is the non-physical axis created by the control that is perpendicular to the part spindle centerline. This virtual axis is assigned system paramacro parameters as the last axis number after the last real axis (excluding non-positioning spindles). For example if your angled wheel grinder has two linear axes, a rotary axis that controls the wheel angle, a non positioning part spindle, and a wheel spindle, then system parameters for axis four represent the status of the virtual axis.

#2001 to 2732
Dresser/Wheel Offset Tables

Use these parameters to enter dresser/wheel offset values into the dresser/wheel offset tables for geometry and radius (as covered in chapter 3). They can be changed or simply read through programming. Table 20.E lists the parameter numbers associated with each table value.

Table 20.E
Dresser/Wheel Offset Table Parameters

Offset Number		Parameter # for Dresser/Wheel Table
Axis 1	1 to 32	#2001 to 2032
Axis 2	1 to 32	#2101 to 2132
Axis 3	1 to 32	#2201 to 2232
Axis 4	1 to 32	#2301 to 2332
Axis 5	1 to 32	#2401 to 2432
Axis 6	1 to 32	#2501 to 2532
Radius	1 to 32	#2601 to 2632
Orientation	1 to 32	#2701 to 2732

#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. It is a write-only parameter. Any block that assigns any non-zero value to parameter #3000 results in a cycle stop. The actual value assigned to parameter #3000 is not used.

ATTENTION: Once axis reciprocation begins, it continues until stopped by a G80 or an end of program (M02, M30, M99). This means that executing paramacro #3000 in a part program does not necessarily stop the reciprocating axis.

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 that an important comment is in the program block that assigns a value to parameter #3000 (see page 10-10 for details on comment blocks).

For example, programming:

```
#3000=.1 (WHEEL NUMBER 6 IS WORN);
```

causes program execution to stop at the beginning of this block and display a message telling the operator to read the comment in the block. A block reset must be performed before a cycle start can resume normal program execution.

When this block is executed, it also sets the paramacro alarm PAL flag ($MCALRM) true. See your system installer's documentation for details on the effect of this PAL flag.

#3001
System Timer (PAL)

Use this parameter to enter or monitor the system timer value. It is a read-write parameter. Every 20ms a value of 20 is added to the value of this parameter.

The value of this parameter is stored in a PAL flag ($PM20MS) and can be modified or set by your PAL program. See the documentation prepared by your system installer for details on the use of this timer. The maximum value of this parameter is 32768ms. Any value greater than 32768 causes this parameter to "rollover" to zero and restart counting. The value of this parameter is reset to zero every time power is lost.

#3002
System Clock

Use this parameter to enter or monitor the system clock value. The system clock parameter references an internal hour counter. It is a read-write parameter.

The maximum value for this parameter is 1 year (8760 hours). The value is maintained when power is lost. It is incremented by .000005556 every 20ms. Negative value assignments are illegal.

#3003
Block Execution Control 1

Use this parameter to determine whether the control ignores single block mode and to determine when M-codes are executed in a block. The value of this parameter ranges from 0 to 3. It is a write-only parameter.

Value of Parameter	Result
0	Single block mode can be activated and M-codes are executed at the beginning of the program block's execution.
1	Single block mode requests are ignored and M-codes are executed at the beginning of the program block's execution.
2	Single block mode can be activated and M-codes are executed after the complete execution of the other commands in the block.
3	Single block mode requests are ignored and M-codes are executed after the complete execution of the other commands in the block.

#3004
Block Execution Control 2

Use this parameter to determine whether a cycle stop request is 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 20.F shows the results of the different values for parameter #3004. If cycle stop, feedrate override, or exact stop mode is ignored, the control does not allow the feature to be used. If they are recognized, the feature can be activated in the normal manner.

Table 20.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. It is a write-only parameter. Any block that assigns a new value to parameter #3006 results in a cycle stop. Any value can be assigned to this parameter. The actual value is not used.

ATTENTION: Once axis reciprocation begins, it continues until stopped by a G80, or an end of program (M02, M30, M99). This means executing paramacro #3006 in a part program does not necessarily stop the reciprocating axis.

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 that an important comment is in a program block (see page 10-10 for details on comment blocks).

For example, programming:

```
#3006=.1 (Install Wheel Number 6);
```

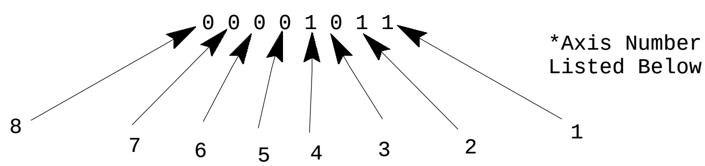
causes program execution to stop at the beginning of this block and the message display to tell the operator to read the comment in the block.

#3007
Mirror Image

Use this parameter to monitor which axes are mirrored. It is a read-only parameter. This parameter is an integer that represents, in binary, which 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 parameter #3007 is the integer 11 (binary 00001011) as shown below, it indicates axes one, two, and four are mirrored and axes three, five, and six are not mirrored:



where:

- 0 indicates axis is not mirrored
- 1 indicates axis is mirrored

#4001 to 4120
Modal Information

Use these parameters to monitor the value of a modal program word. They are read-only parameters. Table 20.G shows the modal program word that applies to the given parameter number.

Table 20.G
Modal Data Parameters

Parameter Number	Modal Data Value
#4001 to 4021	G-code Groups 1-21 (see page 20-54) and list what G-code from which 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 are as follows:

- G02 is in G-code group 1, so parameter #4001 is set to 02
- the programmed feedrate is F100, so parameter #4109 is set to 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.

#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 5089 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.

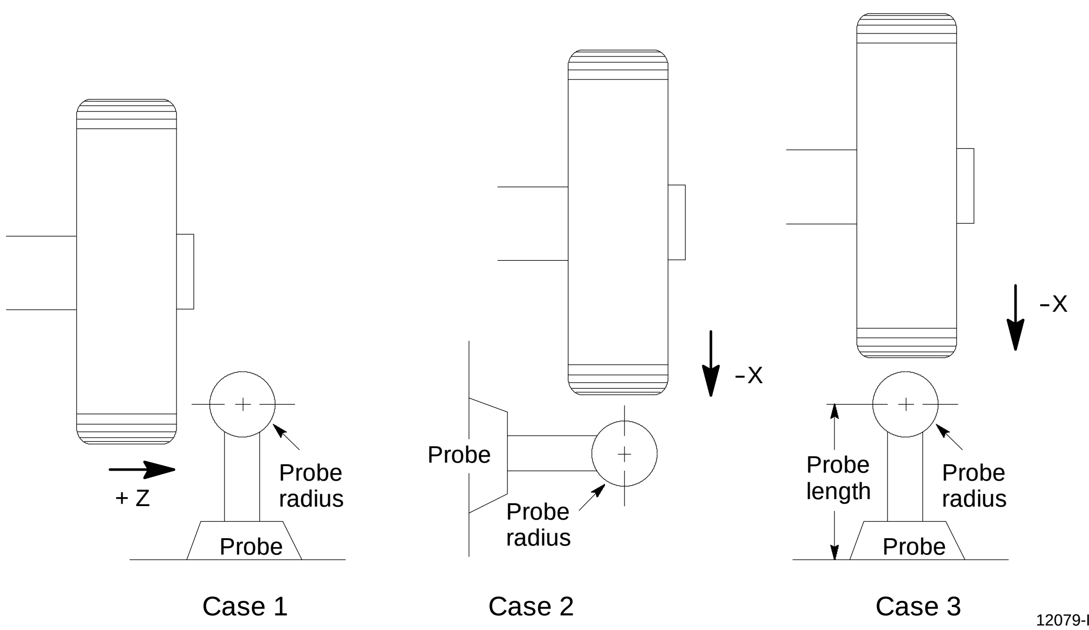
#5095 and 5096
Probe Length and Radius

Probe tip radius and probe length are defined by your system installer in AMP. These values can also be changed by using these paramacro system parameters:

5095	Length of Probe - used primarily for G37 operations. This distance is measured along the axis being probed. For example, if an X-axis probing operation is being performed (as shown in the following figure) then the value entered here is the length of the probe on the X-axis. The length is measured from the tool gauge point to the center of the probe radius.
5096	Radius of Probe - used for either G31 or G37 operations. This parameter defines the radius of the probe tip. If the documentation for your probe lists a tip diameter, enter half of that value.

Figure 20.1 shows how these parameters are typically used.

Figure 20.1
Typical Tool Gauging Configurations



See chapter 19 for details on probing operations.

#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

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

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

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

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

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

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

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

The system installer determines in AMP the name (or word) that is used to define the axis.

#5500 to 5508
In-Process Dresser Parameters

Use these parameters to assign values to the in-process dresser operation. These parameters are read/write. Details on what these parameters represent to the in-process dresser are covered in chapter 21. The in-process dresser parameters accessible through paramacros are:

*5500	New Wheel Diameter
*5501	Warning Wheel Diameter
*5502	Minimum Wheel Diameter
*5503	Maximum Wheel Speed (RPM)
*5504	Dresser Roll Diameter
*5505	Dresser Amount Per Rev
**5506	Dresser Retract Distance
*5507	Dresser Surface Speed Ratio
**5508	Current Wheel Diameter
5509	Wheel Width

* These parameters take effect immediately when changed.
** These parameters do not take effect until the next time the in-process dresser is turned on.

#5600 to 5625
Part Program Block Create Through PAL Display Pages

Use these parameters to assign numeric values to their corresponding block letter codes during part program block creation through PAL display pages. They are read or write parameters. Your system installer must create PAL display pages through which these numeric values must be entered. When the part program block create through PAL display pages feature is activated, the control combines the block letter codes and their assigned numeric values to form a part program block.

5600	A
5601	B
5602	C
5603	D
5604	E
5605	F
5606	G
5607	H
5608	I
5609	J
5610	K
5611	L
5612	M

5613	N
5614	O
5615	P
5616	Q
5617	R
5618	S
5619	T
5620	U
5621	V
5622	W
5623	X
5624	Y
5625	Z

#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

20.3.4 PAL Parameters

Use the paramacro parameters of the control to allow communication of 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 use the output flags to transfer data from PAL to the part program as needed

Input Flags:

There are four-integer or three-integer and 32-bit pattern input parameters available. The part program can only read the values assigned to these parameters; it cannot write values to them. The paramacro input parameters available to the part programmer are:

- #1000 - #1031 and #1040 - #1071.

  Use these paramacro PAL parameters 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 these PAL parameters as <u>integer</u> values regardless of how they are assigned in PAL (as an integer or on a per bit basis). #1032 is the only parameter that can also be interpreted by the control on a per-bit basis using parameters #1000 - #1031. PAL can always interpret these values on either a per-bit basis or as integer values.

See the documentation prepared by your system installer for a detailed description of the use and operation of these input flags.

Output Flags:

Output flags function almost identically to input flags with one key difference: where input flags can only be read by the part program, output flags can be both read and written to by the part program. Use these flags 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 can 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 to or read the values of these flags as integers with the exception of parameter #1132 which can be assigned as an integer or as a bit pattern. The paramacro output 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, PAL interprets it 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 documentation prepared by your system installer for a detailed description of the use and operation of these input flags.

20.4 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 can be used only to assign local parameter values. System, common, and PAL variables cannot 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) cannot be assigned using arguments. The second through fifth sets can be assigned by their association to given words in an argument statement located in a paramacro calling block. Table 20.H lists the arguments and their corresponding parameter numbers.

These arguments assign values to the local parameters associated with the paramacro called in the same block.

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 the specific formats for calling the paramacro), 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

Arguments can be specified as any valid parametric expression. For example:

```
G246A#100B[#500+10.0]C[SIN[#101]];
```

Table 20.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 20.H (B) for the parameter assignment.

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 can also take on the form of:

```
#parameter expression = parameter expression
```

Example 20.14
Legal Parameter Numbers

```
#6=1;
#144=1;
#[14/2]=1;
#[#6]=1;
```

In Example 20.14, all of the parameter numbers are legal. Any time you use a different parameter between the [] symbols, the <u>current</u> value of that parameter is evaluated. 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 20.15 illustrates the assignment of values to the parameters.

Example 20.15
Assigning Values to 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 using multiple assignments in the same block, remember the following:

- 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;
  ```

  causes the error message, “PARAMETER ASSIGNMENT SYNTAX ERROR” to appear
- Only assign the same parameter a value once in each block. For example:

  ```
  #1=5,#2=4,#1=6;
  ```

  causes 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	PRGRAM CHECK	SYSTEM SUPORT	→

	FRONT PANEL	ERROR MESAGE	PASS- WORD	SWITCH LANG		→

The macro parameters are separated into 4 tables that are accessed through softkeys. Table 20.I lists these softkeys, the parameters accessed through them, and additional information on the parameters.

Table 20.I
Macro Parameter Tables

Softkey:	Used to:
{LOCAL PARAM}	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}	View or edit the common parameters (available to any program, subprogram, or paramacro program) numbered 100 - 199. These common parameters lose their value and are reset to zero when power to the control is turned off.
{COM-2A PARAM}	View or edit the common parameters (available to any program, subprogram, or paramacro program) numbered 500 - 519. These common parameters are battery backed up on the 9/240; on the 9/260 and 9/290, they are backed up by the supercapacitor. Their value is retained even when power to the control is cycled off. In addition to being battery backed up, these parameters allow an alphanumeric name to be assigned to them. This alphanumeric name is only for the purpose of easy identification. It cannot be used to call a common parameter in a program.
{COM-2B PARAM}	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. 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.

2. Press the appropriate softkey for the table to be viewed or edited.

(softkey level 2)

↑	LOCAL PARAM	COM-1 PARAM	COM-2A PARAM	COM-2B PARAM		→

↑	BACKUP COM1	BACKUP COM2A	BACKUP COM2B		BACKUP ALL	→

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.

3. Select the parameter to change by moving the cursor to the desired parameter number. The selected parameter appears in reverse video.

Move the cursor an entire page by pressing the up or down cursor key while holding down the [SHIFT] key.

You can also perform a rapid search for the desired parameter number. To do so, press the {SEARCH NUMBER} softkey, key in the parameter number you want, and then press the [TRANSMIT] key. The parameter you want appears in reverse video when it is found.

If you press the {COM-1 PARAM} or {COM-2B PARAM} softkey (in step 2), these options are available to alter the parameter value:

- 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 value 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):" is 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

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 are not displayed. By pressing the {REFRSH SCREEN} softkey, any parameters that have been changed by the program are updated to their current values.

4. Select the softkey to alter the common parameter values.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ZERO VALUE	0 ALL VALUES	REFRSH SCREEN	

If you press the {COM-2A PARAM} softkey (in step 2), these options are also available to alter the parameter name:

- 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 can 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 number 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 is deleted
- To clear all of the parameter names that are listed on the {COMMON 2A} screen, press the {CLEAR ALL NM} softkey. The prompt "OK TO CLEAR ALL NAMES? (Y/N):" appears. Press Y followed by the [TRANSMIT] key to delete all parameter names. Press N followed by the [TRANSMIT] key to abort the clear-all-name operation

Parameter names are only for display purposes. They have no real function other than to permanently label a parameter value. Parameter names and parameter values are retained for parameters even after power is turned off.

5. Select the softkey to alter the common parameter names.

(softkey level 3)

↑	SEARCH NUMBER	REPLCE VALUE	ZERO VALUE	REPLCE NAME	CLEAR NAME	→

↑	CLEAR ALL NM			0 ALL VALUES	REFRSH SCREEN	→

Addressing Assigned Parameters

Once you assign a parameter you can address it in a program:

Example 20.16
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 20.17
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.
```

20.5 Backing Up Parameter Values

You can back up the contents of COM1, COM2A, COM2B, or SHARED 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	SHARED PARAM	→

↑	BACKUP COM1	BACKUP COM2A	BACKUP COM2B	BACKUP SHARED	BACKUP ALL	→

To back up parameters:	Press this softkey:
#100 - 199	{BACKUP COM1}
#500 - 519	{BACKUP COM2A}
#520 - 999	{BACKUP COM2B}
#7100 - 7199 [1]	{BACKUP SHARED} [1]
all of the above	{BACKUP ALL}

[1] Available only on a dual-processing system.

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 the specified part program. You can restore these parameters by selecting and executing that part program.

   Parameters 100 – 199 and 500 – 999 are per-process values; only the values for the ‘active’ process are backed up.

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.

The backup file will also stop executing if the number of digits in the parameter assignment exceeds the maximum allowed by the software. This limit is no more than 10 total digits and no more than 6 digits to the right of the decimal point.

20.6 Macro Call Commands

This section contains these subsections:

When a paramacro is called, execution of the currently active part program is halted, and execution is transferred to the macro program. Paramacros can be called in the following ways:

- programming G65 in a part program
- programming G66 or G66.1 in a part program
- setting the proper AMP data (calls 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 page 10-8 for details 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 executes more than one time.

The control first searches the main program directory for the program called by the paramacro. If the control does not find the program in the main program directory, it then checks the protectable program directory for the program. If programs in the main and protectable directories have the same name, the control uses the program in the main directory.

Important: The M99 code can 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 macro has been executed 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.

ATTENTION: Any edits that are made to a subprogram, or to a paramacro program that has already been called for automatic execution, are ignored until the calling program is disabled and reactivated (see page 7-7). Subprograms and paramacros are called for automatic execution the instant that the calling program is selected as active (see page 7-8).

20.6.1 Non-Modal Paramacro Call (G65)

Use this format when calling a paramacro using the G65 command:

```
G65 P_ L_ A_ B_;
```

Where:	Determines:
P	the program number of the called macro. P ranges from 1 - 99999.
L	the number of times the macro is executed. L ranges from 1 - 9999, and can 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. 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.

This macro is executed 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 using any valid parametric expression. For example:

```
G65 P1002 L[#1+1] A[12*6] B[SIN[#101]];
```

20.6.2 Modal Paramacro Call (G66)

Use this format when calling a paramacro using the G66 command:

```
G66 P_ L_ A_ B_;
```

Where:	Determines:
P	the program number of the called macro. P ranges from 1 - 99999.
L	the number of times the macro is executed after each motion block that follows the G66. L ranges from 1 - 9999, and can 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. 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 calls the macro specified by the G66 block.

This macro is executed 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 executes 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 re-programming 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 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 canceled 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 20.18
Modal Macro Call

```
N0100G66P1001;
N0200G65P1002;
```

In Example 20.18, after the complete execution of the macro 1002, the macro 1001 is called. Any motion blocks in macro 1002 causes macro 1001 to be executed.

Example 20.19 illustrates modal macro operation.

Example 20.19
Modal Macro Operation

(MAIN);	
O1000;	
NO10G90;	
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.

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.

20.6.3 Modal Paramacro Call (G66.1)

Use this format when calling a paramacro using the G66.1 command:

```
G66.1 P_ L_ A_ B_;
```

Where:	Determines:
P	the program number of the called macro. P ranges from 1 - 99999.
L	the number of times the macro is executed. L ranges from 1 - 9999, and can 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. 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 <u>after</u> the G66.1 block, are used as argument assignments:

  N: When programmed after a word other than N or O, it is used as assignment #14.

  G: The last G-code programmed in a block is 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 20.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 20.20 illustrates G66.1 macro operation.

Example 20.20
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 re-program 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. See Example 20.21.

Example 20.21
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]];
```

20.6.4 AMP-Defined G-Code Macro Call

Use this format when calling an AMP-defined macro:

```
G_ A_ B_;
```

Where:	Determines:
G_	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. 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 your 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 can range from 1 to 255.9. Your system installer can 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. See your system installer’s documentation for details.

Important: Your system installer can disable the use of AMP-defined G and M-code macro calls when in MDI mode. See your 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 executed using either G66 modality (see page 20-50 for details) or G66.1 modality (see page 20-53 for details). This modality type for AMP-defined G-codes is also determined by your system installer in AMP.

Optional argument statements following an AMP-defined G-code can 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 is 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 other AMP-defined macro calls (see page 20-57 for details).

20.6.5 AMP-Defined M-Code Macro Call

Use this format when calling an AMP-defined M-code macro:

```
M255 A_B_
```

Where :	Programs :
M255	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. 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 your system installer. Your system installer can define M-codes that call paramacro programs with program names ranging from 9001 to 9009. See your system installer's documentation to determine which M-codes call which 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 can range from -1 to 999. Your system installer can define a maximum of 9 AMP-defined M-codes to call specific paramacro programs.

Important: Your system installer can optionally disable the use of AMP-defined G and M-code macro calls when in MDI mode. See your system installer's documentation to determine if this feature is functional in MDI.

Important: Certain AMP-defined M-code macro calls cannot be called by other AMP-defined macro calls (see page 20-57 for details).

20.6.6 AMP-Defined T-, S-, and B-Code Macro Call

Use this format when calling an AMP-defined T-, S-, or B-code macro:

```
T t ;  S s ;  or  B b ;
```

Where:	Is:
t:	is equal to the value assigned to parameter #149
s	is equal to the value assigned to parameter #147
b	is equal to the value assigned to parameter #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 these 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:

- 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.

- An AMP flag for that specific word must be turned on by your system installer to allow that word to call a macro.
- 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.

Important: Certain AMP-defined T-, S-, or B-code macro calls cannot be called by other AMP-defined macro calls (see page 20-57 for details on nesting macros).

20.6.7 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 can be nested up to a maximum of four levels. However, if the maximum number of nested paramacros (four) is combined with up to four subprograms that end with M98, a maximum of eight 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.

Take precautions when attempting to nest AMP assigned macro calls since many combinations of these calls cannot be valid. Your system installer determines in AMP the functionality of the AMP-defined macro call when nested.

The following two options are available:

- 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 20.J.

Table 20.J
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 can be called from the macro type down the left column.

No – the macro type across the top row cannot be called from the macro type down the left column. When this nesting is attempted, the control executes any other operation that would normally be performed by that G, M, T, S, or B code (as defined by the 9/SERIES 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 20.K.

Table 20.K
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 can be called from the macro type down the left column.

No -- the macro type across the top row cannot be called from the macro type down the left column. When this nesting is attempted, the control executes any other operation that would normally be performed by that G, M, T, S, or B code (as defined by the 9/SERIES 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 executes the programmed code as some other function (as defined by the system 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 20.K is that an AMP-assigned macro cannot 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.

20.7 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 the following 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 can be output to an RS-232 compatible device). The port used to output information from the control is selected in AMP by your system installer.

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;
```

PCLOS

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 has been output. This command only executes provided that the corresponding POPEN command has previously been executed.

The format for a PCLOS 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 is not 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 can 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 20.22
Sample of a BPRNT Block

```
BPRNT[INSTALL*WHEEL*NUM-
BER*1#123[4]*PRESS*CYCLE*STOP**#234[2]];
```

Example 20.22 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 20.23 gives an example of a BPRNT program.

Example 20.23
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 20.23 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 is not executed if the POPEN command has not been issued.

The format for the DPRNT block is:

```
DPRNT [ s #p[id]...];
```

Where :	Is :
s	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	any valid parameter number. Note that the parameter number must have the # sign before it.
i	the number of digits before the decimal point to be output.
d	the number of digits after the decimal point to be output.

Important: The sum of i + d cannot exceed 8.

There can be as many s and #p in a block as you want provided that the length of the block does not exceed the maximum block size.

Example 20.24
Sample of a DPRNT block

```
DPRNT[INSTALL*WHEEL*#123[53]*PRESS*CYCLE*STOP**#234[20]];
```

Example 20.24 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 can be either ASCII/ISO or EIA format as determined in AMP by your system installer. Leading zero suppression, and/or a leading plus sign can also be set by AMP parameters.

Example 20.25 gives an example of a DPRNT program.

Example 20.25
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 the above program would be:

```
COMMENT HERE X23.456 Y-98.700 T35
```

END OF CHAPTER

Chapter 21

In-process Dresser

21.0 Chapter Overview

This chapter describes these topics:

In this chapter, we cover general information about dressing with the in-process dresser (IPD) feature. The in-process dresser lets the control perform a continuous dressing operation while the control is "in process" of grinding a part.

While this dressing is taking place, the grinding wheel shrinks in diameter. The in-process dresser takes into consideration this change in wheel diameter and offsets the wheel position accordingly. This prevents the grinding wheel from losing contact with the part it is grinding as its diameter shrinks.

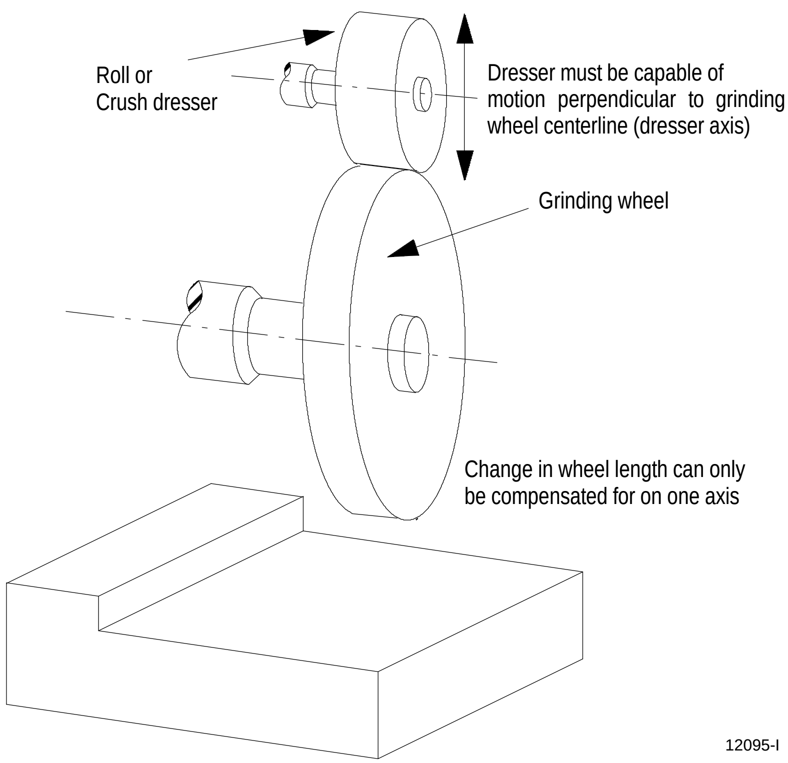
Important: Because of this feature's close interaction with the wheel length offset and dresser/wheel radius compensation features, we recommend that you familiarize yourself with the operation of these features before proceeding with this section. See chapters 3 and 13 for details on wheel length offsets, and chapters 3 and 15 for details on dresser/wheel radius compensation.

Machine Requirements

Use this feature with a roll or crush dresser. The dressing tool must be able to move independently of wheel motion and in a direction perpendicular to the grinding wheel center line. The dressing axis positions the dresser perpendicular to the grinding wheel center line. Control over the rotation of the roll or crush dresser is determined by your system installer through PAL programming. See the documentation provided by your system installer for details.

Important: The in-process dresser feature requires that your control be configured such that the S word controls the grinding speed. For cylindrical grinders, this means you must be capable of performing CSS on the grinding wheel, not the part spindle. See spindle speed control, page 12-34 for details on CSS and spindle configurations.

Figure 21.1
Roll Dresser Motion Perpendicular to Grinding Wheel Centerline



21.1 Offset Generation While Dressing

While the in-process dresser is active and dressing the grinding wheel, the wheel is shrinking in diameter. The control uses the value entered into the in-process dresser table to calculate the amount of wheel shrinkage per wheel revolution. See the description on page 21-10 for details on the “Dressing Amount Per Rev” parameter.

The control uses this amount of wheel shrinkage to generate an in-process dresser offset. This offset is similar to a wheel length offset. It compensates changing wheel diameter by creating additional length offsets. The axis that this dresser offset is applied to is determined by the current plane selection.

In-process dresser offsets cannot be modified, activated, or deactivated by the programmer. As long as the in-process dresser remains active, any generated dresser offset also remains active. Once the in-process dresser is deactivated, the control cancels any currently active in-process dresser offset. If the in-process dresser offset is to be maintained, you must take steps to make sure the correct wheel length or radius offset numbers are activated as described on page 21-6.

In-process offset axis motion is automatically generated as the wheel is positioned about the part and the in-process dresser changes the wheel diameter. This offset is updated every 20ms as the wheel is dressed.

We recommend the use of dresser/wheel radius compensation (G41 or G42) during grinding operations when the in-process dresser is active. This prevents drastic changes to wheel path at corner intersections by generating a circular transition around the corner. This circular transition forces the generated length compensation move to be tangent to the generated circular transition.

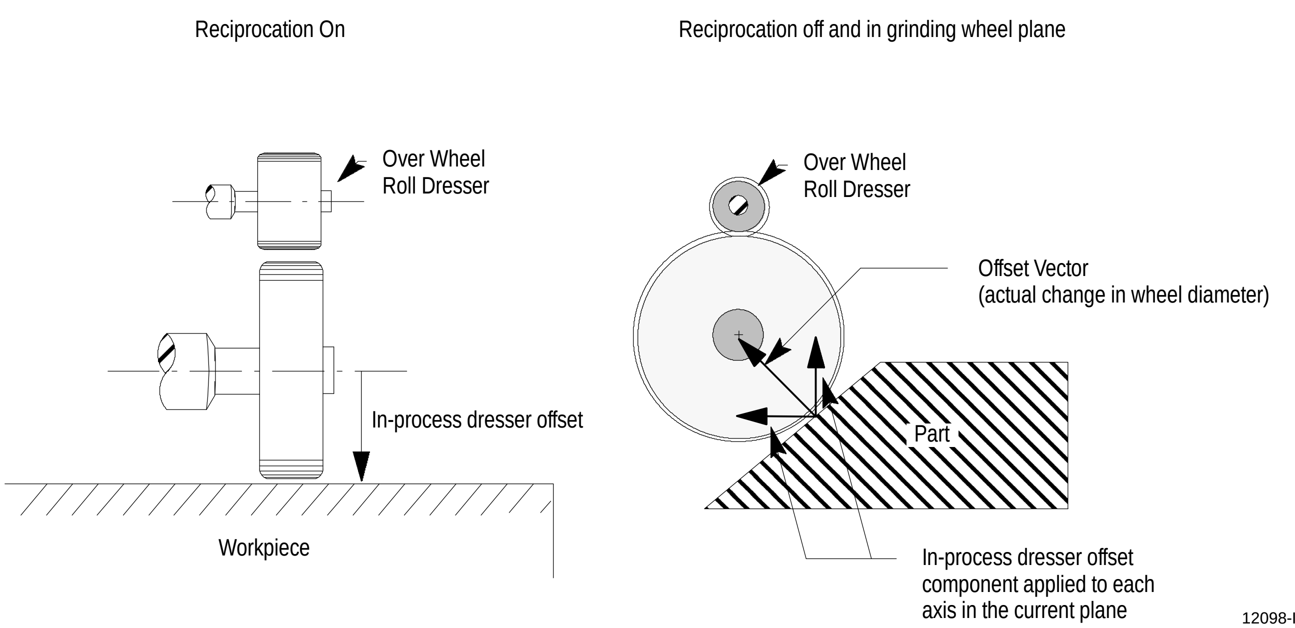
The in-process dresser feature does not have any effect on the grinding wheel radius if dresser/wheel radius compensation is active. In-process dresser offsets are only length type offsets. This can cause some slight inaccuracies on more complex grinding contours. If changes in wheel radius must be considered by the in-process dresser, you should use radius offset 33. This offset is updated with one-half the new wheel diameter when the in-process dresser is deactivated. The more frequently the in-process dresser is deactivated, the more accurate the value of radius offset number 33. See “Maintaining Dresser Offsets” on page 21-6 for details on radius offset number 33.

21.1.1 Plane Selection for the In-process Dresser Offset

As the grinding wheel is dressed, the in-process dresser generates an offset similar to a length offset. The axis (or axes) that this dresser offset is applied to is directly dependent on the currently active plane (G17, G18, or G19) and the grinding wheel plane (defined as the plane made up of the horizontal and vertical compensation axes selected in AMP for the in-process dresser). The following table applies provided the control is not in the end of program state.

Reciprocation is:	the active plane is:	the offset affects:
On	any plane	only the axis defined in AMP as the vertical compensation offset axis. See Figure 21.2.
Off	the same as the plane of the grinding wheel (defined in AMP as the horizontal and vertical compensation axes)	both axes in the plane. The offset vector is broken down into its two components in that plane and applied to the appropriate axis on a per move basis. See Figure 21.2.
Off	not the same as the plane of the grinding wheel	the axis that is in both the grinding wheel plane (either the horizontal or vertical compensation axis defined in AMP) and the currently active plane. The entire in-process dresser offset is applied to this axis.

Figure 21.2
Offset Cases as Dependent on Active Plane

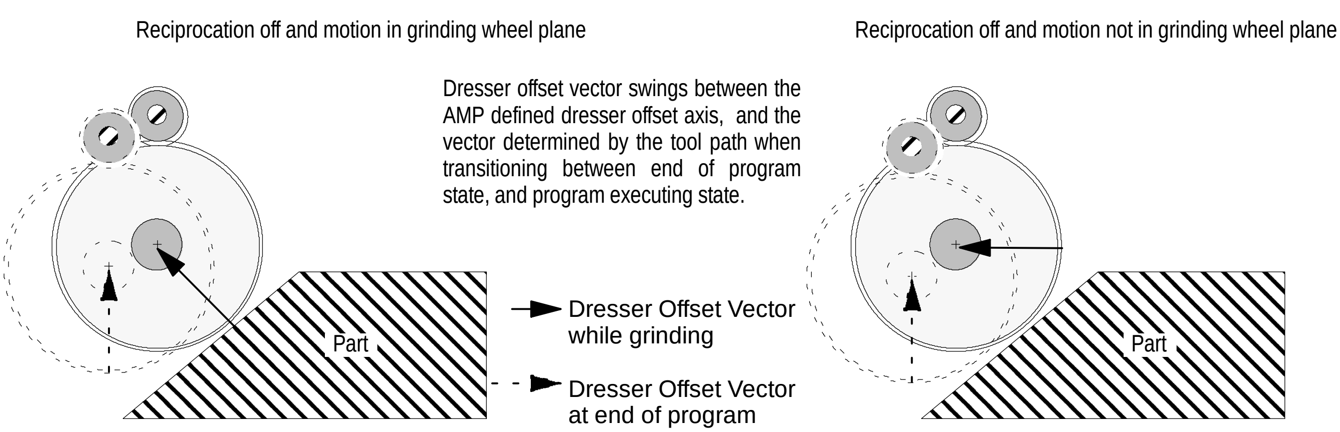


The following discussion only applies if your system installer has configured you system to leave the in-process dresser on at the end of program state. If the control is in the end of program state and the in-process dresser is on, the in-process dresser offset is always applied to the AMP defined vertical compensation offset axis (as though reciprocation was on). The control enters the end of program state after executing an M02 or M30 in either a part program or MDI program. An M99 however will not cause the control to enter the end of program state.

An end of program state also occurs for MDI blocks when they are completed and no part program is selected as active. For example:

MDI mode is:	and a part program is:	in-process dresser offset is applied:
executing	active or inactive	perpendicular to the current wheel path
completed	active	perpendicular to the last wheel path
completed	inactive (none selected)	to the AMP defined vertical dressing axis

Important: When the control transfers the in-process dresser offset that is perpendicular to the wheel path, to the vertical axis or vice versa, wheel motion relative to the part occurs. The direction and amount of this relative motion is dependant on the surface you were grinding and the amount of dressing that has occurred. If reciprocation is on the in-process dresser offset always remains on the AMP defined dressing axis.



ATTENTION: Damage to the part or machine can occur if care is not taken as to the direction of the in-process dresser offset. For example if you are at the end of program state and jog the cutting tool against the part while the in-process dresser is on, when you execute a program block the grinding wheel can swing into, or away from the part when the offset swings off of the AMP defined dressing axis. We recommend shutting the in-process dresser off at the end program state.

21.1.2 Maintaining Dresser Offsets

When the in-process dresser is deactivated, the control will remove any offsets that have been generated by the in-process dresser. This can cause the wheel to lose contact with the part since the wheel diameter has changed and the offset compensating for this change has been canceled.

To solve this problem the in-process dresser has been designed to modify select length and radius offsets in the offset tables that correspond to specific offset numbers. These offsets are automatically updated to reflect the new wheel diameter when the in-process dresser is deactivated.

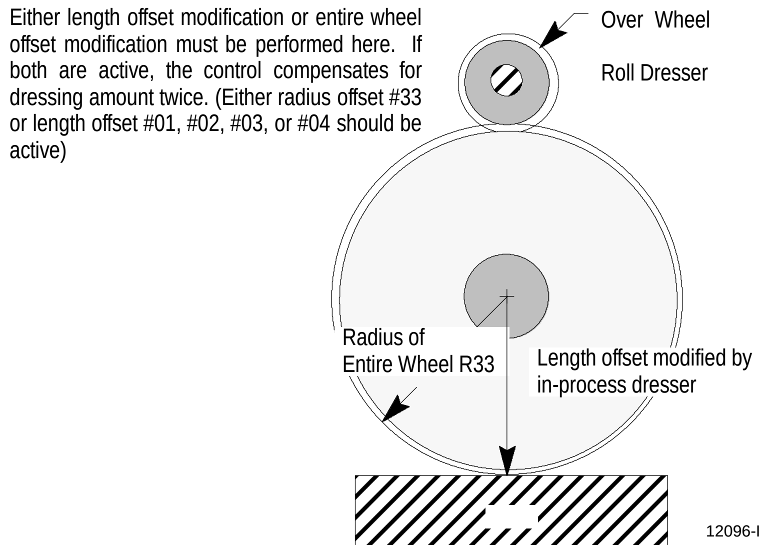
- Length offsets 01, 02, 03, and 04
  Wheel length offset 01 to 04 are the length offsets that the in-process dresser modifies when deactivated. When the in-process dresser is deactivated, the control replaces the value of the length offset with the “Vertical dressing axis” for each of these 4 length offsets. The “Vertical dressing axis” is configured in AMP by the system installer. The control replaces this length offset with a value equal to one-half the current wheel diameter. Make sure you are calling one of these length offsets with the correct T word to properly maintain a length offset.

- Radius offset 33
  Radius offset 33 is the radius offset that the in-process dresser modifies when canceled. When the in-process dresser is deactivated the control replaces the value of radius offset 33 with one-half the value of the current wheel diameter. Radius offset number 33 cannot be modified and does not appear on the radius offset table screen. This radius offset number is reserved exclusively for the in-process dresser.

Length offsets other than 01 – 04 or radius offsets other than 33 can be used; however, it is the programmer’s responsibility (or system installer’s PAL program responsibility) to make sure that these offsets get updated when the in-process dresser is deactivated.

Keep in mind you should only have either the length offset or the radius offset active. If both are active you will be compensating for the change in wheel diameter twice as shown in Figure 21.3.

Figure 21.3
In-process dresser should compensate for either length or radius change (not both)



21.2 Activating the In-process Dresser

PAL controls the rotation of the actual in-process dresser and the enabling/disabling of the in-process dresser feature. See the documentation prepared by your system installer for details. This section assumes you have already calibrated your dresser axis as described on page 21-12. You must calibrate your dresser axis before the in-process dresser can be activated.

Dresser Enable

When the in-process dresser is enabled, the control automatically performs a retract return towards the grinding wheel on the dresser axis. The retract return amount is the same distance as the “Dresser Retract Distance” configured on the in-process dresser status page (see page 21-10). This retract return is only performed if the dresser is currently in a retracted position (i.e., the dresser does not return when enabled immediately after a dresser calibration operation).

The feedrate at which the in-process dresser is returned from retract towards the grinding wheel is configured in AMP as the retract feedrate. Once the dresser reaches the return position, the control continues to infeed the dresser in the same direction (towards the wheel) at the configured dressing rate.

While the in-process dresser is active the dressing axis cannot be positioned by any other normal means (i.e., cannot be programmed, jogged, or positioned by the PAL axis mover).

Dresser Disable

When the in-process dresser is disabled, the control automatically retracts the roll away from the grinding wheel using the dresser axis. The amount the dresser is retracted is configured on the in-process dresser status page. See page 21-10 for information on the “Dresser Retract Distance” parameter.

The feedrate that the dressing tool is retracted away from the grinding wheel is configured in AMP as the retract feedrate. Once the dresser reaches the retract position, the control updates the wheel length offset values for length offsets 01, 02, 03, and 04, and updates the radius offset number 33.

ATTENTION: When the control enters E-STOP while the in-process dresser is active, the dresser axis can perform a dresser retract operation when E-STOP reset is performed. This is because an E-STOP reset will typically deactivate the in-process dresser. It is the system installer’s responsibility to make sure that the in-process dresser is not disabled in PAL at E-STOP reset. This will make certain that this motion does not occur.

When the in-process dresser is disabled, the dressing axis can be positioned as any other positioning axis (i.e., it accepts normal programming commands and can be jogged or positioned through the PAL axis mover).

Important: The in-process dresser feature cannot be disabled if dresser/wheel radius compensation (G41 or G42) is active. If an attempt is made to cancel the in-process dresser feature when compensation is active, the control postpones the deactivation of the in-process dresser until compensation is turned off (G40).

21.3 On-line In-process Dresser Parameters

This section covers the parameters used to control how the in-process dresser operates. This section assumes that the in-process dresser has already been configured correctly in AMP. See your AMP reference manual. Only operating parameters can be changed on line (at the control). Basic configuration parameters must be modified in AMP. Table 21.A lists the in-process dresser parameters found on the in-process dresser status screen. See page 21-11.

Important: IPD works only with spindle 1. If your system is configured for two spindles, IPD looks at spindle 1 and ignores spindle 2.

Table 21.A
Dresser Parameters on the In-process Dresser Screen

This Parameter: (paramacro system parameter)	Indicates:	Range:
DRESSER/ACTIVE/INACTIVE	if the in-process dresser is currently active (turned on) or inactive (turned off). This is controlled by PAL.	Active (in-process dresser on) Inactive (in-process dresser off)
DRESSER HOLD/NOT HOLD	if the in-process dresser is currently in the hold state. The hold state stops the wheel dressing operation without turning off the in-process dresser. Although both dress retract and dress retract return are available, programmed and jog moves on the dressing axis are still not permitted because the in-process dresser is still active. This is controlled by PAL.	Hold (dressing infeed stopped) Not Hold (dressing in-process)
INITIAL WHEEL DIAMETER	the starting wheel diameter that was active the last time the in-process dresser was calibrated. When the wheel is calibrated, the control writes the value of the new wheel diameter into this parameter. The difference between this parameter and the “Current Wheel Diameter” parameter is equal to the amount the wheel has been dressed since it was last calibrated. The value of this parameter only changes when the wheel is calibrated or the in-process dresser is deactivated.	0 to 9999.9999 [inch] 0 to 99999.999 [mm]
CURRENT WHEEL DIAMETER (paramacro #5508)	the value of the current wheel diameter. When the wheel is calibrated, the control copies the “New Wheel Diameter” value into this parameter. Once the in-process dresser is enabled, this value is reduced by the amount of wheel shrinkage as the wheel is dressed. When the in-process dresser is disabled, the control copies the current wheel diameter value into the “Initial Wheel Diameter” parameter.	0 to 9999.9999 [inch] 0 to 99999.999 [mm]
CURRENT DRESSER RPM	the current speed of the rotating roll dresser in RPM.	System Defined
* NEW WHEEL DIAMETER (paramacro #5500)	the value of the wheel diameter before the in-process dresser is calibrated. Once calibrated the value of this parameter is copied into the “Initial Wheel Diameter” parameter.	0 to 9999.9999 [inch] 0 to 99999.999 [mm]
* WARNING WHEEL DIAMETER (paramacro #5501)	a wheel diameter greater than the “Minimum Wheel Diameter”. When the wheel is dressed down to this diameter, a warning message is displayed. This value should be large enough so that the operator has sufficient wheel left to finish current operations and take corrective action before the wheel is dressed down to the “Minimum Wheel Diameter”. Your system installer determines in PAL the control’s response to a wheel that has been dressed down to the warning wheel diameter.	0 to 9999.9999 [inch] 0 to 99999.999 [mm]
* MINIMUM WHEEL DIAMETER (paramacro #5502)	the minimum acceptable wheel diameter. When the wheel is dressed down to this diameter, a warning message is displayed. The system installer determines in PAL the control’s response to a wheel that has been dressed down to the minimum wheel diameter.	0 to 9999.9999 [inch] 0 to 99999.999 [mm]
* MAXIMUM WHEEL SPEED(RPM) (paramacro #5503)	the maximum speed of the grinding wheel. This is the maximum speed the grinding wheel is allowed to reach when constant surface speed is turned on. On cylindrical grinders keep in mind that only one spindle is controlled by CSS.	0 to 999999. (RPM)
* WHEEL WIDTH (paramacro #5509)	this parameter is present only on this screen for operator convenience in entering this parameter for later use with paramacros.	0 to 9999.9999 [inch] 0 to 99999.999 [mm]
* DRESSER ROLL DIAMETER (paramacro #5504)	the diameter of the roll dressing tool.	0 to 9999.9999 [inch] 0 to 99999.999 [mm]

* This parameter can be configured by the operator on this screen. Other parameters are assigned by the control or PAL during calibration or dressing.

This Parameter: (paramacro system parameter)	Indicates:	Range:
* DRESSER AMOUNT PER REV (paramacro #5505)	the amount the dresser is to feed into the grinding wheel for each revolution of the grinding wheel. When the in-process dresser is active, this amount of infeed is subtracted from the "Current Wheel Diameter" at each wheel revolution. This is actually the dresser axis feedrate (in feed per rev) when it is dressing the wheel.	0 to 9.9999999 [inch] 0 to 0.000000 [mm]
* DRESSER RETRACT DISTANCE (paramacro #5506)	the distance the dresser roll should retract away from the grinding wheel when the in-process dresser is deactivated. When in-process dressing is re-activated, the dresser roll automatically performs a retract return equal to the amount of the dresser retract distance. If the dresser is not currently retracted when the in-process dresser is enabled, the control does not perform the retract return (as is the case when after a dresser calibration operation). The dresser retract or retract return can also be performed on request when in the dresser hold state. Both of these moves are performed on the dresser axis.	0 to 999.9999 [inch] 0 to 99999.999 [mm]
* DRESSER SURF SPEED RATIO (paramacro #5507)	the ratio of the wheel's surface speed to the roll dresser surface speed. A ratio of 1.00 would rotate the dresser and the wheel at the same relative surface speed. A ratio of 0.50 would rotate the dresser at one half the surface speed of the wheel. A ratio of 1.50 would rotate the dresser at one and one-half times the surface speed of the wheel. Surface speed of the wheel is controlled through your part program and CSS. Surface speed of the roll dresser is controlled through PAL and calculated from the "Dresser Roll Diameter", "Dresser Surface Speed Ratio", and the current surface speed of the grinding wheel. The sign of this parameter indicates the direction that the dresser will rotate relative to the grinding wheel: "+" equals the same direction and "-" equals the opposite direction.	0 to 99999.00

* This parameter can be configured by the operator on this screen. Other parameters are assigned by the control or PAL during calibration or dressing.

Entering In-process Dresser Parameters

This section covers entering parameter values into the in-process dresser table manually. You can also enter values in this table through programming with paramacro system parameters (#5500-#5509). See chapter 20 for details on assigning values to system parameters.

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

```
(softkey level 1)

     | PRGRAM | OFFSET | MACRO | PRGRAM | SYSTEM | -->
     | MANAGE |        | PARAM | CHECK  | SUPORT |

     | FRONT  | ERROR  | PASS- | SWITCH |        | -->
     | PANEL  | MESAGE | WORD  | LANG   |        |
```

2. Press the {DRESSR TABLE} softkey to display the in-process dresser status screen.

(softkey level 2)

↑	WORK CO-ORD	WHEEL GEOMET	RADIUS TABLE	DRESSR TABLE	SCALNG	→

↑	COORD ROTATE	BACKUP OFFSET				→

The in-process dresser status screen appears:

```
ENTER NEW WHEEL DIAMETER:

IN-PROCESS DRESSER STATUS                  [INCH]
 DRESSER ACTIVE/INACTIVE  :              INACTIVE
 DRESSER HOLD/NOT HOLD    :              NOT HOLD
 INITIAL WHEEL DIAMETER   :                0.0000
 CURRENT WHEEL DIAMETER   :                0.0000
 CURRENT DRESSER RPM      :                    0.

*NEW  WHEEL DIAMETER      :               20.0000
*WARNING WHEEL DIAMETER   :               19.0000
*MINIMUM WHEEL DIAMETER   :               18.7500
*MAXIMUM WHEEL SPEED(RPM) :                90000.
*WHEEL WIDTH              :                0.0000
*DRESSER ROLL DIAMETER    :                2.1000
*DRESSER AMOUNT PER REV   :             0.0000140
*DRESSER RETRACT DISTANCE :                0.2000
*DRESSER SURF SPEED RATIO :                  0.81
```

↑	REPLCE VALUE	ADD TO VALUE	INCH/ METRIC			

* only these parameter values can be altered by the operator (asterisk does not appear on screen)

3. Use the up and down cursor keys to place the cursor on the parameter you want to modify. The selected parameter appears in reverse video.

   Table 21.A describes these parameters.

4. Select the unit system to be used (inch or metric) by pressing the {INCH/METRIC} softkey.

5. Replace the current value of the parameter by pressing the {REPLCE VALUE} softkey or add an amount to the current value by pressing the {ADD TO VALUE} softkey.

(softkey level 3)

↑	REPLCE VALUE	ADD TO VALUE	INCH/ METRIC			→

6. Key in the value to replace or add to the current value and press the [TRANSMIT] key. The control updates the parameter value selected.

Typically, if the in-process dresser is active when the parameter values are modified, the new values for these parameters take effect immediately. The exception to this is the "Dresser Retract Distance" that takes effect the next time the control performs either a dresser retract or dresser retract return motion, and the "New Wheel Diameter" parameter that is used only when the in-process dresser is calibrated.

21.4 Calibrating the In-process Dresser

In-process dresser calibration is a measurement feature that tells the control the exact diameter and location of the grinding wheel relative to the dressing axis. You cannot activate the in-process dresser until you perform a calibration operation. If you do not calibrate the in-process dresser before you activate it, the control will display an error.

Calibrate the in-process dresser when:

- a new wheel is mounted
- any dressing or grinding operations that change wheel diameter take place while the in-process dresser is disabled or in a "Hold" state
- power to the control is turned off

  Any time power to the control is turned off, PAL must re-initialize the in-process dresser by performing a calibration operation.

The calibration operation is very PAL dependent. See the documentation prepared by your system installer for details on wheel calibration. A typical wheel calibration routine would consist of these steps:

1. Access the in-process dresser status screen and enter a new wheel diameter value. If this is a new grinding wheel or a wheel that has had a diameter change due to some non-in-process dresser operation, you must manually measure the wheel diameter. If power to the control was shut off and no change in wheel diameter has occurred since the last in-process dresser operation, you can enter the value from the "Current Wheel Diameter" parameter as the "New Wheel Diameter." This can also be performed automatically through PAL.

2. Position the dresser axis until it physically touches the grinding wheel. This can be performed by jogging, programming, using PAL axis mover, or some other method devised by your system installer to position the dresser axis.

3. Indicate to PAL that it should initialize the in-process dresser at the current dresser axis location. See the documentation prepared by your system installer.

END OF CHAPTER

Chapter 22

Program Interrupts and Dressing Interrupts

22.0 Chapter Overview

This chapter describes these topics:

Use the program interrupt or dressing interrupt features if you want to execute a subprogram or paramacro program while suspending operation of some other currently executing program, subprogram, or paramacro.

Two styles of interrupts are available:

Program Interrupts (Type 1 and Type 2) See page 22-1	Use these interrupts to interrupt any currently executing program block so that an operation defined in a subprogram or paramacro program can be performed. This style of interrupt is not typically used to interrupt program execution during reciprocation as finish results can be undesirable.
Dressing Interrupts (Type 2 only) See page 22-10	Use this interrupt exclusively to interrupt reciprocation or grinding cycle operation so that an operation defined in a subprogram or paramacro program can be performed. This style of interrupt is typically used for dressing.

22.1 Program Interrupts

Use program interrupts to temporarily halt the currently executing program and transfer execution to a subprogram or paramacro program. During reciprocation, a program interrupt request can cause finish problems as the grinding wheel can be decelerated across the part during the reciprocation pass.

We recommend that you use dressing interrupts if you want to interrupt reciprocation. We describe the actual operation of program interrupts during reciprocation in the section, “Immediate vs. Delayed Interrupts,” on page 22-7.

22.1.1 Enabling/Disabling Program Interrupts (M96, M97)

Enable or disable program interrupts on the control by using two modal M codes. Your system installer determines in AMP these M codes. This manual assumes these values for these M codes (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 causes an interrupt program to be executed. When program interrupts are disabled (M97), the control ignores the interrupt signal.

The format for these M codes is:

```
M96L__P__;

M97L__;
```

Where:	Is:
L	the type of interrupt and the signal that calls the interrupt L ranges from 0 - 3.
P	the interrupt program P is followed by a 5-digit, numeric, non-decimal program name.

An error appears if you program anything other than an N, P, or L word, a block delete /, or a comment character in the M96 or M97 block.

You can also program interrupt M code M96 or M97 within a subprogram or paramacro program called by the program interrupt. If this is the case, the new program interrupt does not become enabled/disabled until the currently executing subprogram or paramacro program is completed and execution is returned to the main program.

A program interrupt cannot be executed while another program interrupt or dressing interrupt is in progress.

Important: Interrupt programs should normally be disabled during thread grinding. The execution of an interrupt program during a threading pass can cause undesired results.

Selecting the Type of Program Interrupt (L word)

There are two types of interrupt programs that you can use to enable or disable these M codes. You can use up to 4 signals from PAL (switches) to call interrupt programs. Your system installer determines what switch corresponds to which type of interrupt in PAL. See documentation prepared by your system installer for details on the types available and the switches used to control them for a specific machine application.

The M code determines the interrupt type and which switch signal is enabled or disabled by programming an L word with the M96 or M97 code. There are 4 L words:

This L word:	determines this interrupt type:	and this switch signal is enabled:
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

You program these L words in each M96 or M97 block. Not programming the L word causes the control to assume that an L word of L0 has already been programmed. If you program more than one L word in a block, the control uses the right-most L word for that block and ignores other L words in that block.

The difference between a type 1 and type 2 interrupt is the path that is taken when the return from the interrupt is made (as programmed with an M99 in the interrupt subprogram or macro program).

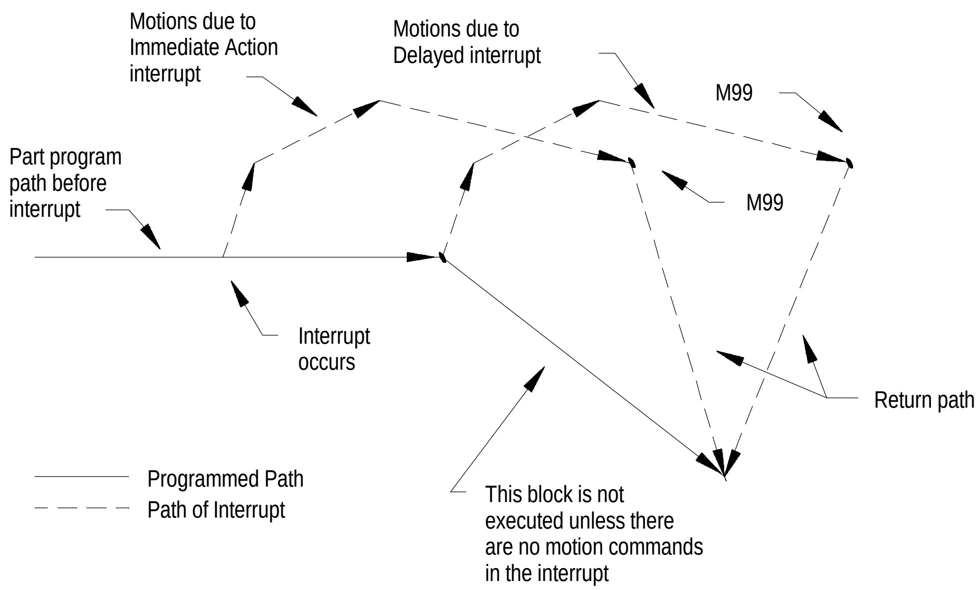
Important: Your system installer can determine in AMP if a signal to execute a program interrupt is delayed until the end of a currently executing block (delayed interrupt), or if the interrupt is executed immediately when the signal is received (immediate action interrupt). This selection dramatically affects the execution of your interrupt program, as shown in Figure 22.1 and Figure 22.2.

Type 1 Program Interrupts

If no axis motion is generated by the subprogram or macro program called by the type 1 program interrupt, the control halts program execution. The control then executes the subprogram or macro program called, returns to the interrupted part program, and continues execution as normal.

If axis motion is generated by the subprogram or macro program called by the type 1 program interrupt, when the interrupt is completed the control executes the next fully unexecuted program block in the part program and continues executing the part program from this point (see Figure 22.1).

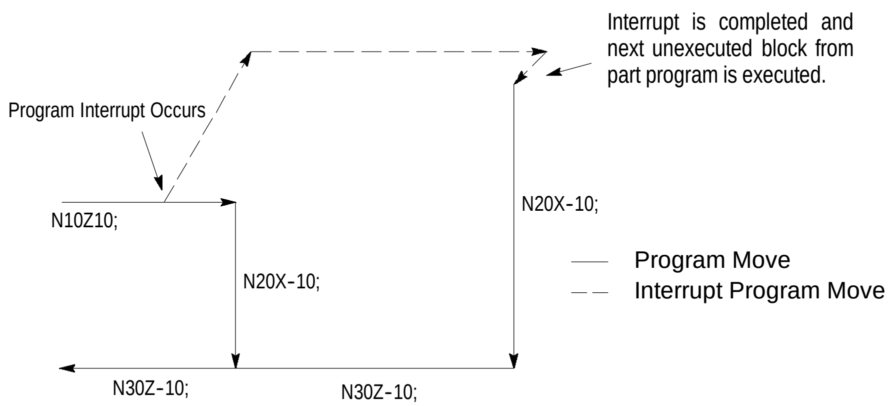
Figure 22.1
Type 1 Interrupt



ATTENTION: If the interrupt is executed in the middle of a block’s execution, and there is axis motion in the interrupt program, then the control never reaches the end-point of the interrupted block; execution transfers to the end-point of the following block. This applies to type 1 interrupts only.

If the next un-executed part program block after the interrupt does not contain an endpoint for all axes moved during the interrupt, the endpoint of the move is not the same endpoint had the interrupt not occurred. For example:

Figure 22.1
Type 1 Interrupt with Single Axis Moves

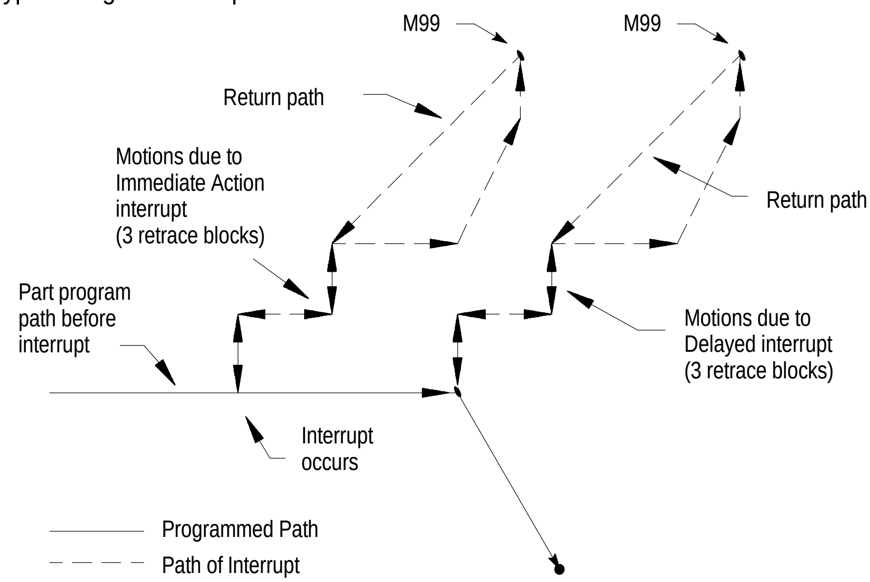


Type 2 Program Interrupts

When type 2 program interrupts are complete, the control returns the wheel to the same point in the calling program that it was at when the interrupt was performed.

Normally, the first 4 linear moves (G00 or G01) in the interrupt program are remembered and retraced when returning to the calling program. This can be altered by programming a specific M code, as described later. If a non-linear (G02 or G03) circular move is performed as one of the first 4 blocks of the interrupt, the control remembers only the moves up to that block. If a circular block is the first block in the interrupt, then the start-point of that block is all that is remembered. In this case, the control returns the wheel to the part in a linear move to that start point.

Figure 22.2
Type 2 Program Interrupts



Number of retrace blocks for type 2 interrupts

You can alter the number of blocks that the control retraces when returning to the start-position of type 2 interrupts. The default number of retraced blocks is 4; however, you can alter that number by programming one of these M codes:

Programming an:	Retraces:
M900	zero blocks
M901	one block
M902	two blocks
M903	three blocks
M904	four blocks

You can program these M codes in any block in the main program before the control executes the interrupt program. You can also program them within an interrupt, though the M code does not go into effect until execution is returned to the main program. If not programmed, the default is four blocks. The default is automatically reset at the end-of-program commands (M02 or M30), E-STOP reset, or control reset.

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 takes on the number of retrace blocks as programmed with this M code.

When the return from interrupt is executed (M99 in the interrupt program), the control generates a linear move to the end-point of the last-remembered move for retrace. The moves are then retraced, returning the wheel back to the start-point of the interrupt. This may not be the same location in the calling program if a different wheel offset has been activated.

Immediate vs. Delayed Interrupts

Your system installer determines in AMP if an interrupt occurs immediately upon request, or if it is delayed until the end of the currently executing program block. When immediate is selected as the interrupt type, the control decelerates the axes to a stop when the interrupt request is made and promptly executes the interrupt program. When delayed is selected, the control decelerates the axis at the end of the currently executing block. Once in position at the end of this block, the control executes the interrupt program.

During reciprocation, including reciprocation caused by execution of a grinding cycle:

- immediate interrupts are executed immediately regardless of the current status of the grinding operation

  All axis motions are decelerated to a stop, including reciprocation motion, and the interrupt program is executed.

- delay interrupts are executed at the end of the currently executing motion

  The current reciprocation move, and any motions normally generated by the cycle at this reversal point (any pick, plunge, or dither motions), are decelerated to their endpoints. The endpoint of reciprocation is defined as either the primary or secondary reversal points. Once at the endpoint with axis motion stopped, the interrupt program is executed.

When the interrupt program is completed, control is returned to the calling program and program execution (and reciprocation) resumes.

Important: If you execute an interrupt during reciprocation, including reciprocation caused by execution of a grinding cycle, you should use the dressing interrupts. Both immediate and delay type program interrupts can cause major finish problems on the workpiece if reciprocation is decelerated across the part. See page 22-10 for details on dressing interrupts.

Selecting an Interrupt Program (P word)

You can select any legal subprogram or paramacro as a program interrupt program (see chapter 10 on subprograms or chapter 20 on paramacros). To use a program as an interrupt program, it must have a program name of 5 numeric characters following an O address (see chapter 11 for program names). This interrupt program must contain an M99 block as its last block. The M99 block has all the same restrictions as an M99 block for a subprogram as described in chapter 10.

An M02 or M30 end of program command can also be used at the end of the interrupt program. If an end of program command is executed in an interrupt program, the control will assume it has completed program execution of both the interrupt and the main program. Any remaining blocks in the interrupt or main calling program are not executed. If executing a program from memory, the main program is reset to the first block.

Program a P word in the M block that enables the interrupt (M96 in this manual) to assign subprograms or paramacro programs to a particular type of interrupt. When selecting a program with a P word, only the numeric value of the program name is entered; the letter O is omitted. For example, programming

```
M96L0P11111;
```

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 generates an error when the switch that activates the program is enabled. If the P word is not programmed in the M96 block, the control defaults to the last programmed P word. The P word is not programmed in the disable M code block (M97 in this manual).

Example 22.1
Enabling and Disabling the Interrupt Features

`N1M96L0P11111;`	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. 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.
`N5M96L3;`	Re-enables interrupt program called by switch 3. Since no P word is programmed in this block, the last programmed P word (enabling program O11111) is active.
`N5M96L0P22222;`	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.

22.2 Dressing Interrupts

Use dressing interrupts to temporarily halt reciprocation or a grinding cycle and execute a subprogram or paramacro call. This feature allows the operator/programmer to interrupt a reciprocating grinding operation or grinding cycle with a wheel dressing operation or some other function.

Three methods of requesting a dressing interrupt are available:

- On operator request (through PAL interface usually a push button or switch)
- Auto-Dressing through programming of a grinding cycle (executed at pre-determined passes in the reciprocating cycle as requested with a D word in the cycle block)
- Through programming a pre-dress operation before execution of the grinding cycle with a G8n.1

Important: Reciprocation or any cylindrical or surface grinding cycle (G8n) must be active before a dressing interrupt can be requested. Dressing interrupts can not be performed when the control is in G80 mode.

22.2.1 Operator Request for Dressing Interrupt

The operator can request a dressing interrupt at any time through PAL (provided a surface/cylindrical grinding cycle or reciprocation is currently active). For details on how to perform this request, see your system installer's documentation. The program called by this dressing interrupt must have previously been defined using a P word in either a reciprocation block (G81) or any grinding cycle block (cylindrical or surface). Once you program a P word, it remains the active P word until another P word is programmed or an end of program code is executed (M02, M30, or M99).

22.2.2 Auto-Dressing Request during Grinding Cycle (D word)

During grinding cycles (for both cylindrical and surface grinders), the number of dressing operations to be performed during the course of the cycle's execution is defined using a D word. Each of these dressing operations actually calls a dressing interrupt. The program called for by this interrupt is also defined in the cycle block with the P word. See chapters 16 and 17 for details on grinding cycle block format.

The number of dressing interrupts programmed with the D word are evenly distributed over the course of the cycle's execution. The dressing interrupt request is automatically made just after the plunge moves.

22.2.3 Dressing Interrupt through Pre-Dress Request

In addition to auto-dress as described on page 22-10, grinding cycles for both cylindrical and surface grinders can be defined to have a pre-dress feature (identified by the decimal point following the cycle G code, i.e. G8n.1). Grinding cycles with pre-dress automatically perform a dressing interrupt before the cycle motions begin, including reciprocation.

The program called for by this pre-dress interrupt is also defined in the cycle block with the P word. See chapters 16 and 17 for details on which grinding cycles have pre-dress and their block formats.

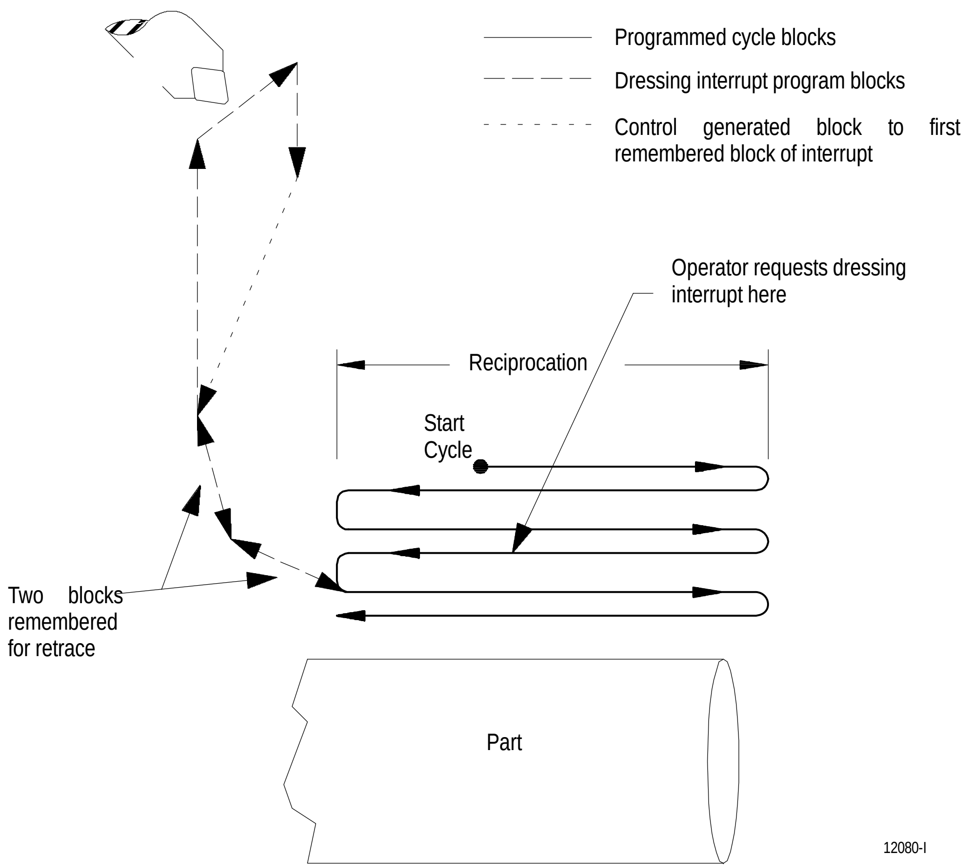
22.2.4 Dressing Interrupt Execution

During reciprocation or any grinding cycle, dressing interrupts are executed at the end of the currently executing motion. The current reciprocation move, and any motions normally generated by the cycle at this reversal point (any pick, plunge, or dither motions), are decelerated to their endpoints. The endpoint of reciprocation is either the primary or secondary reversal point. Once at the endpoint with axis motion stopped, the interrupt program is executed.

When dressing interrupts are complete, the control returns the wheel to the same point in the calling program that it was at when the interrupt was performed (same as type 2 program interrupts).

Normally, the first 4 linear moves (G00 or G01) in the interrupt program are remembered and retraced when returning to the calling program. This can 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 four blocks of the interrupt, the control remembers only the moves up to that block. If a circular block is the first block in the interrupt, then the start-point of that block is all that is remembered. In this case, the control returns the wheel to the part in a linear move to that start point.

Figure 22.3
Dressing Interrupts



Upon returning from the dressing interrupt, axis cycle or reciprocation resumes as does program execution from the point of interruption.

Important: When dressing occurs, the in-process dresser feature generates offsets based on the newly dressed wheel size. These offsets are activated as necessary upon the wheel's return to the calling program. For details on the in-process dresser feature, see chapter 21.

Number of retrace blocks for dressing interrupts

You can alter the number of blocks that the control retraces when returning to the start position of the interrupt. The default number of retraced blocks is 4. Alter the number of retrace blocks by programming one of these M codes:

Programming an:	retraces:
M900	zero blocks
M901	one block
M902	two blocks
M903	three blocks
M904	four blocks

You can program these M codes in any block in the calling program before the control executes the dressing interrupt program. You can also program them within an interrupt, though the M code does not go into effect until execution is returned to the calling program. If not programmed, the default is four blocks. The default is automatically reset at the end-of-program commands (M02 or M30), E-STOP reset, or control reset.

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 takes 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 generates a linear move to the end-point of the last-remembered move for retrace. The moves are then retraced, returning the tool back to the start-point of the interrupt. This may not be the same location in the main program if a different tool offset has been activated.

22.3 The Interrupt Program (P word)

Remember when you use a program as an interrupt program for both types of interrupts (program interrupts as well as dressing interrupts):

- Any modal data (G codes, feedrates, spindle speeds, coordinate system offsets, etc.) contained in the calling program are carried into the interrupt program. Any changes made to this modal data within the interrupt are effective only in the interrupt program; changes are not carried back into the calling program when the interrupt is completed. This does not include wheel or wheel offset data that is changed in the interrupt. Any wheel or wheel offset changes are carried back into the suspended calling part program.

- Your system installer can determine if an interrupt program is to be called as a paramacro or a subprogram when executed. If you call it as a paramacro, remember that this assigns a new set of local parameters for the interrupt. If you call it as a subprogram, the same set of local parameters that apply to the interrupted program applies to the subprogram.

- If you choose an interrupt as a macro program, it cannot be a macro that requires the assignment of local variables in the calling block. It cannot require an argument.

- Always call macro type interrupts as the G65 non-modal type. G66 and G66.1 modal types cannot be called. See page 20-50 for details on 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 results in an error. The control executes other programming commands to the left of the M99 code in the block. The control ignores any characters to the right of the M99 code or any blocks following the M99 code.

  An M02 or M30 end of program command can also be used at the end of the interrupt program. If an end of program command is executed in an interrupt program, the control will assume it has completed program execution of both the interrupt and the main program. Any remaining blocks in the interrupt or main calling program are not executed. If executing a program from memory the main program is reset to the first block.

- The interrupt program cannot contain any grinding cycles (G8n.n). However, turning cycles (G20 and G24) and threading operations (G33 and G34) are permitted in an interrupt program.

- Coordinate system offsets are illegal in an interrupt program. This includes G52, G92, G92.1, and G92.2.

- Any inherent modality from the main program, such as an active modal paramacro, turning routine, or grinding cycle, is temporarily canceled during the execution of an interrupt.

- Only one interrupt can be executed at a time. All can be active at once, but only one can be executed. This means that an interrupt cannot be executed during the execution of another interrupt, nor can a dressing interrupt be executed during a program interrupt.

22.4 Interrupt Request Considerations

Consider this list when programming and executing program or dressing interrupts:

- Your 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 (assuming the interrupt is already enabled). This is to prevent the accidental execution of an interrupt program if a switch is inadvertently left on when a program begins execution. This applies to program interrupts and operator requested dressing interrupts
- Interrupt programs can only be executed when the control is in the automatic mode. Interrupt requests that occur during MDI or MANUAL modes are ignored
- Dresser/wheel radius compensation
  - If compensation is active (G41 or G42) when an interrupt request is made, it remains active throughout execution of the interrupt unless canceled with a G40. Different tool radii (selected with a T word) and different compensation directions (G41 or G42) can be programmed in the interrupt program. However, it is the programmer's responsibility to make sure the correct compensation direction (G41, or G42) is active and the correct length and radius offset is active (as programmed with the T word) <u>before</u> the M99 return from interrupt command is executed.
  - If compensation is not active when the interrupt request is made, it can be activated by programming either a G41 or G42 in the interrupt program. Compensation, however, must be turned off with a G40 in the interrupt program before the M99 return block is executed.
- Interrupts that are requested when the control is in E-STOP are ignored regardless of whether the interrupt is enabled or not.
- You can execute interrupts only when the control is in the cycle start state. If you request for an interrupt when the control is in cycle stop or cycle suspend, the control recognizes the interrupt request. The control executes the interrupt program when a cycle start state becomes active again. Dressing interrupts can only be requested when the control is not in G80 mode (control must be reciprocating or performing a surface/cylindrical grinding cycle).

- If an interrupt occurs during a block retrace, the interrupt is performed. The block retrace is aborted at that point and no further retrace is allowed. Block retrace still returns any moves that have already been retraced before the interrupt occurred.
- When an interrupt request is performed, the control will decelerate the axis to a stop. If there is not enough distance remaining in the block to properly decelerate the axis, the control will perform any remaining deceleration in one 20ms step at the end of the current block (often causing overcutting or excessive machine loading). This condition is not typical since normal acc/dec limits feedrate so there is always enough block to properly decelerate the axis. In cases when the normal acc/dec feedrate check is disabled (G36.1 mode for very short blocks) this occurrence can be much more frequent.

END OF CHAPTER

Appendix A

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 MESAGE	PASS-WORD	SWITCH 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 MESAGE	PASS-WORD	SWITCH 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 MESAGE}
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
SPLIT ON/OFF

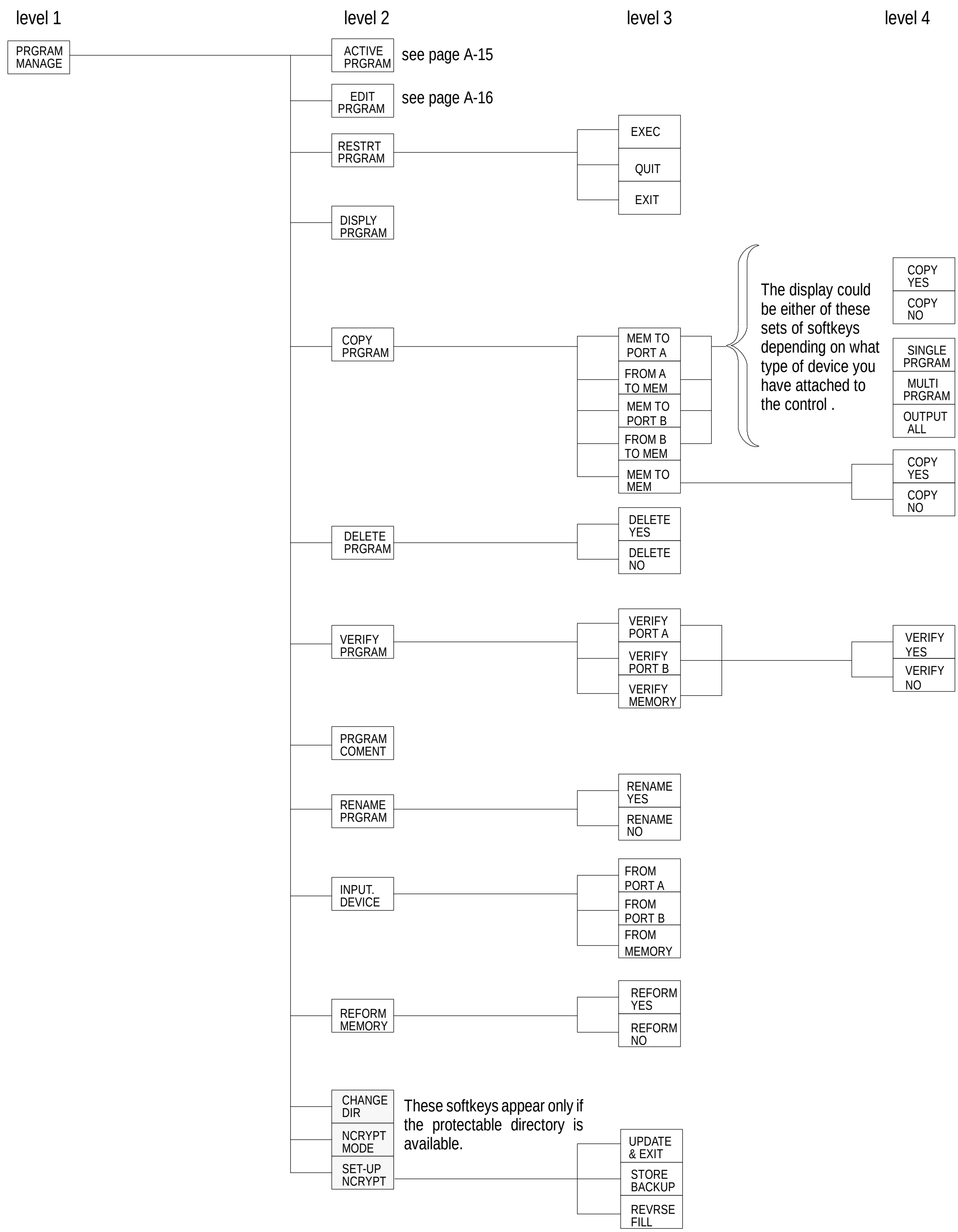
NOTE: The first 4 softkeys (from PRGRAM to DTG) toggle between small and large screen display.

THE FUNCTION SELECT SOFTKEYS LEVEL 1

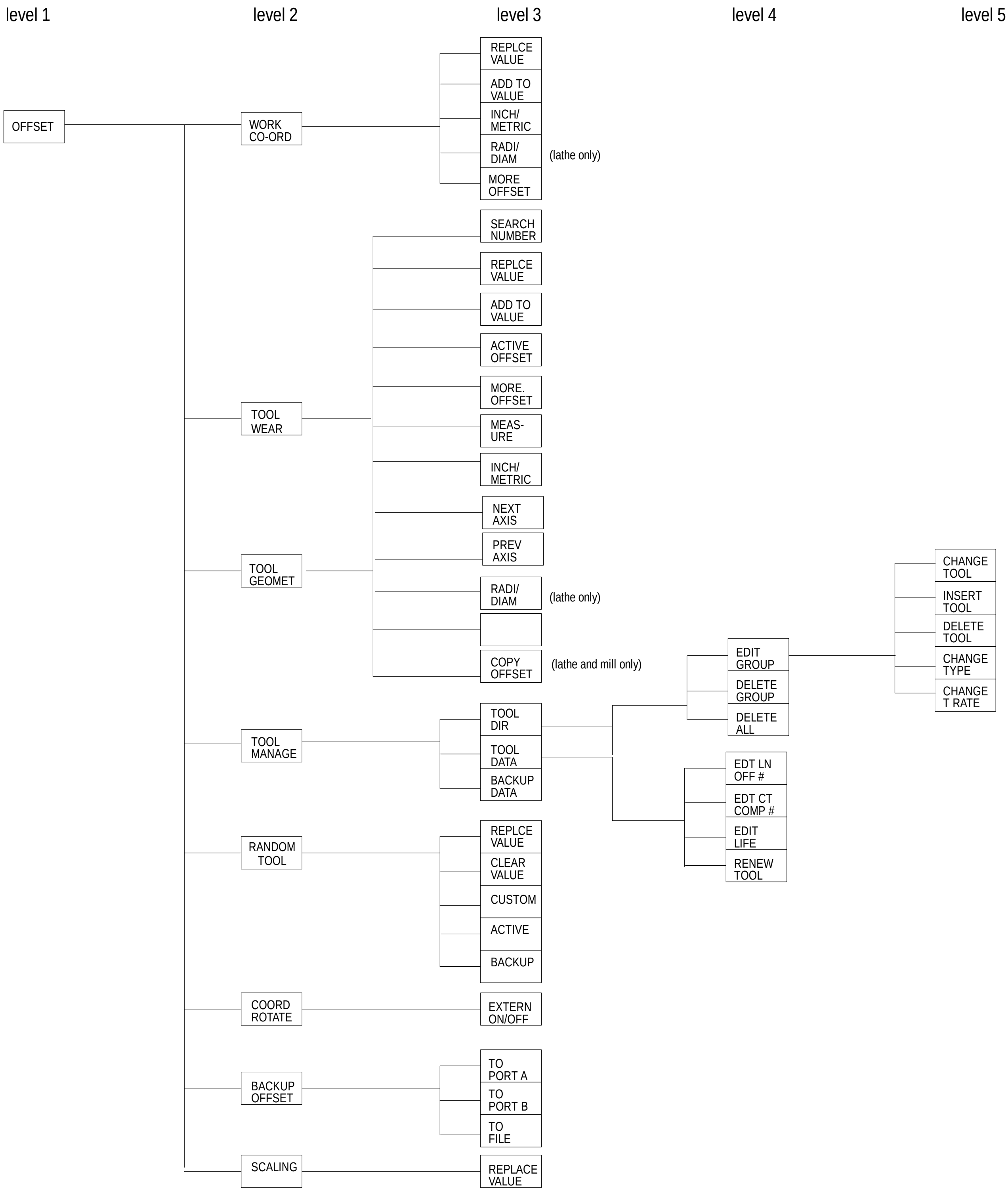
WITH POWER UP (AXIS POSITION) DISPLAY SCREEN

	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 MESAGE	see page A-13
	PASS-WORD	see page A-14
	SWITCH 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.		

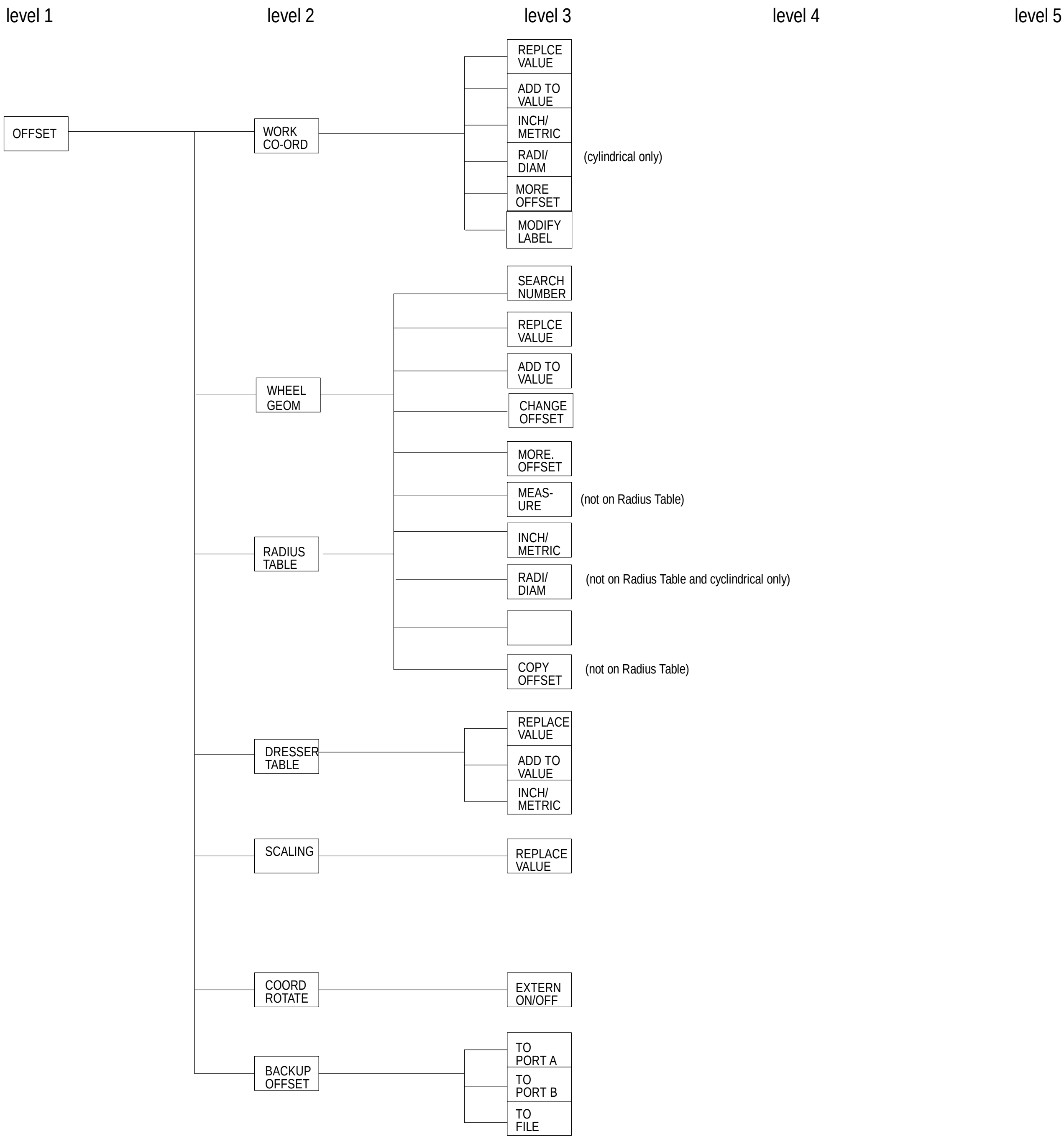
PRGRAM MANAGE



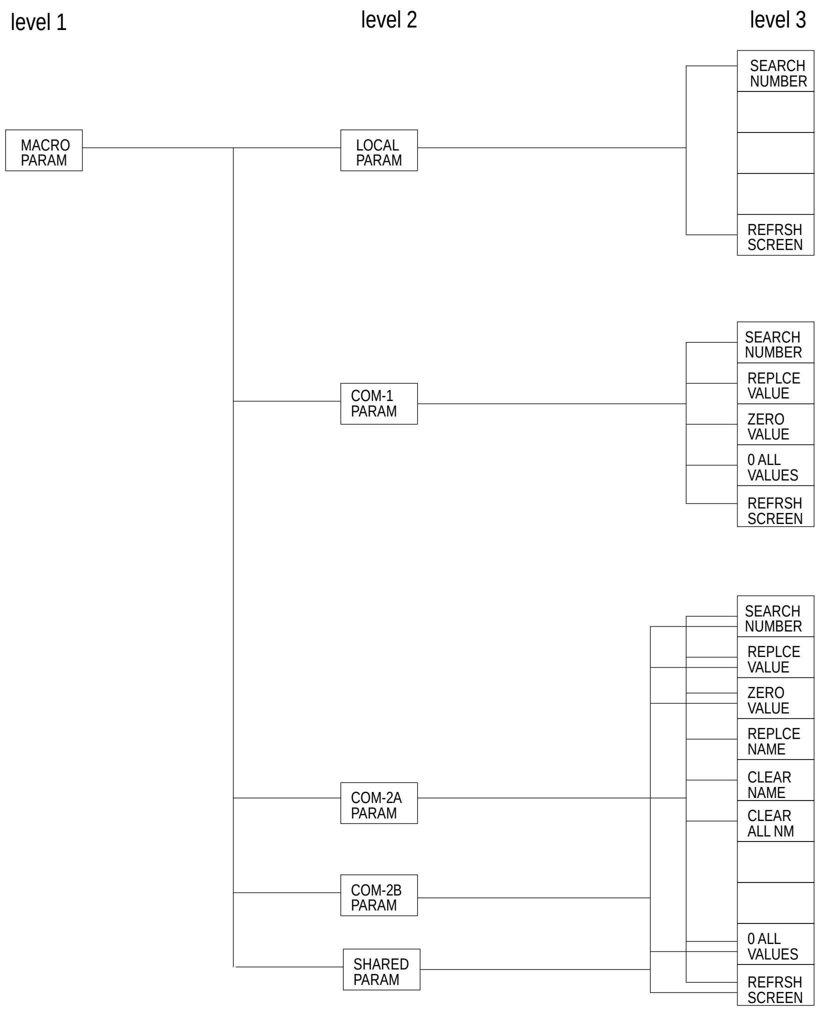
OFFSET (Lathe & Mill)



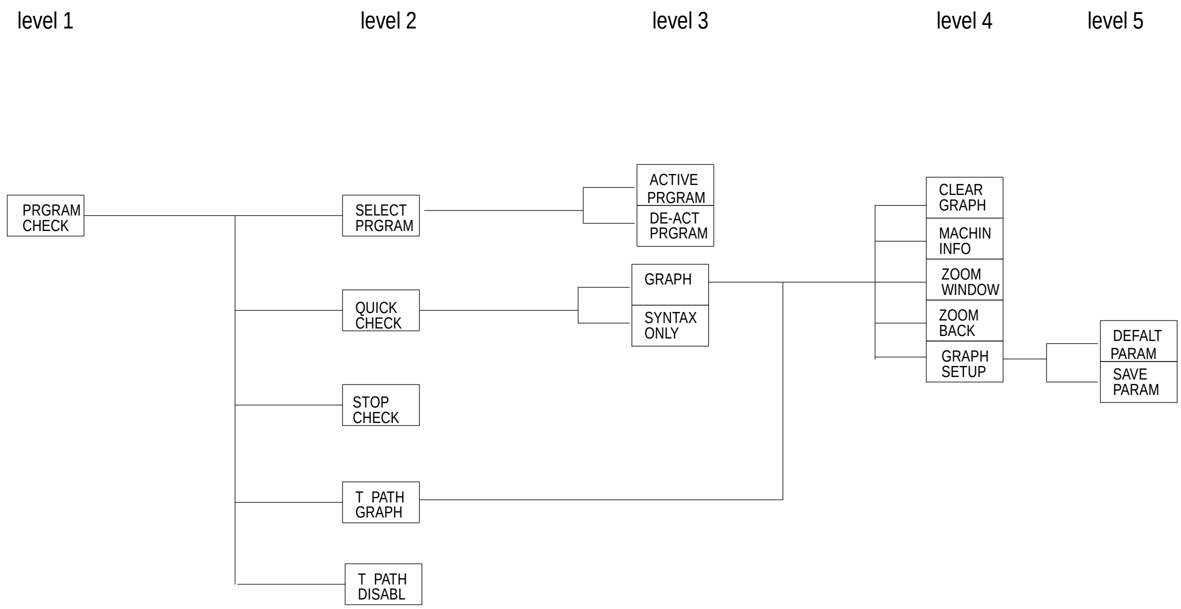
OFFSET (Grinder)



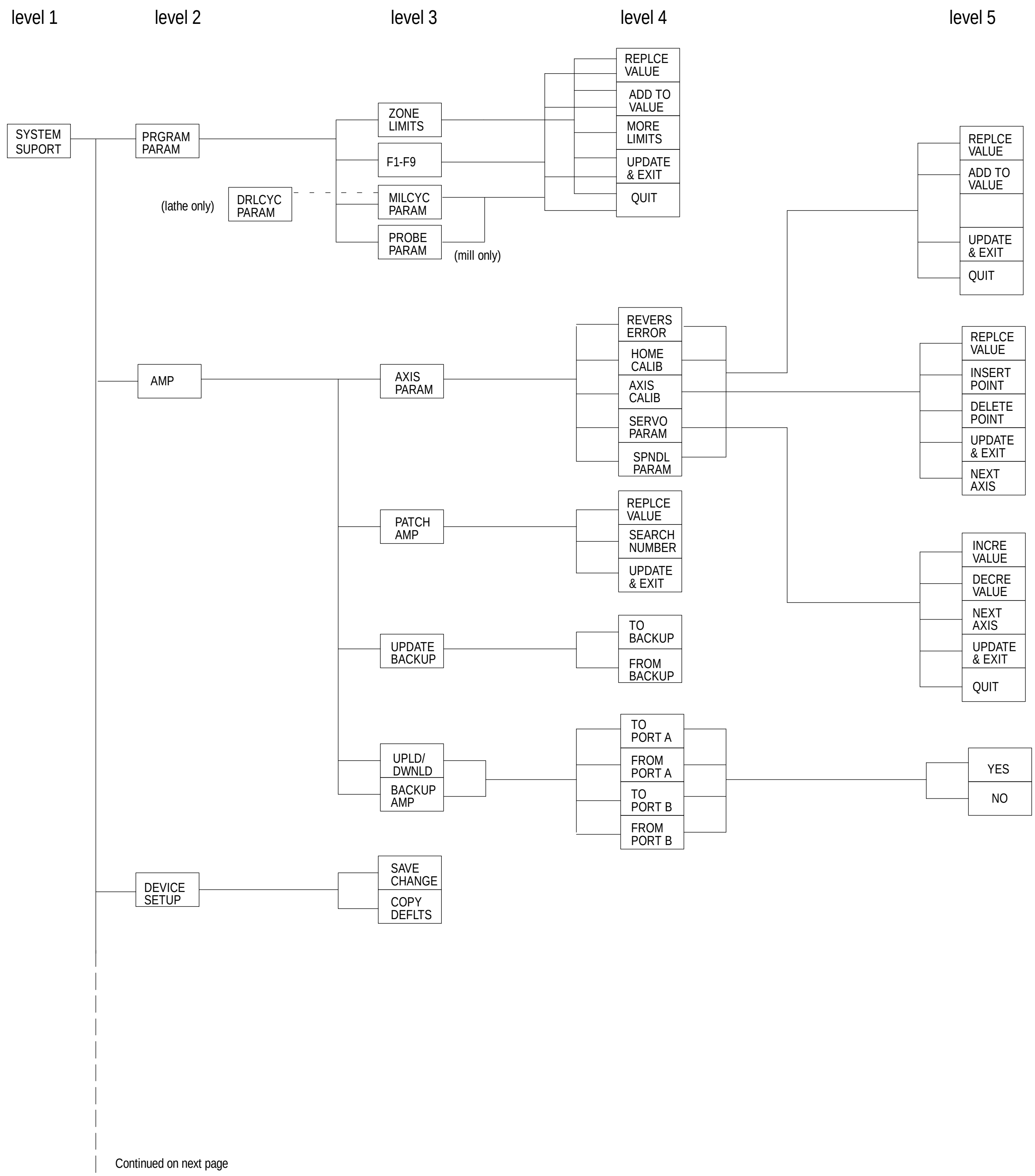
MACRO PARAM


level 1
level 2
level 3
MACRO PARAM
LOCAL PARAM
COM-1 PARAM
COM-2A PARAM
COM-2B PARAM
SHARED PARAM
SEARCH NUMBER
REFRSH SCREEN
SEARCH NUMBER
REPLCE VALUE
ZERO VALUE
0 ALL VALUES
REFRSH SCREEN
SEARCH NUMBER
REPLCE VALUE
ZERO VALUE
REPLCE NAME
CLEAR NAME
CLEAR ALL NM
0 ALL VALUES
REFRSH SCREEN

PRGRAM CHECK

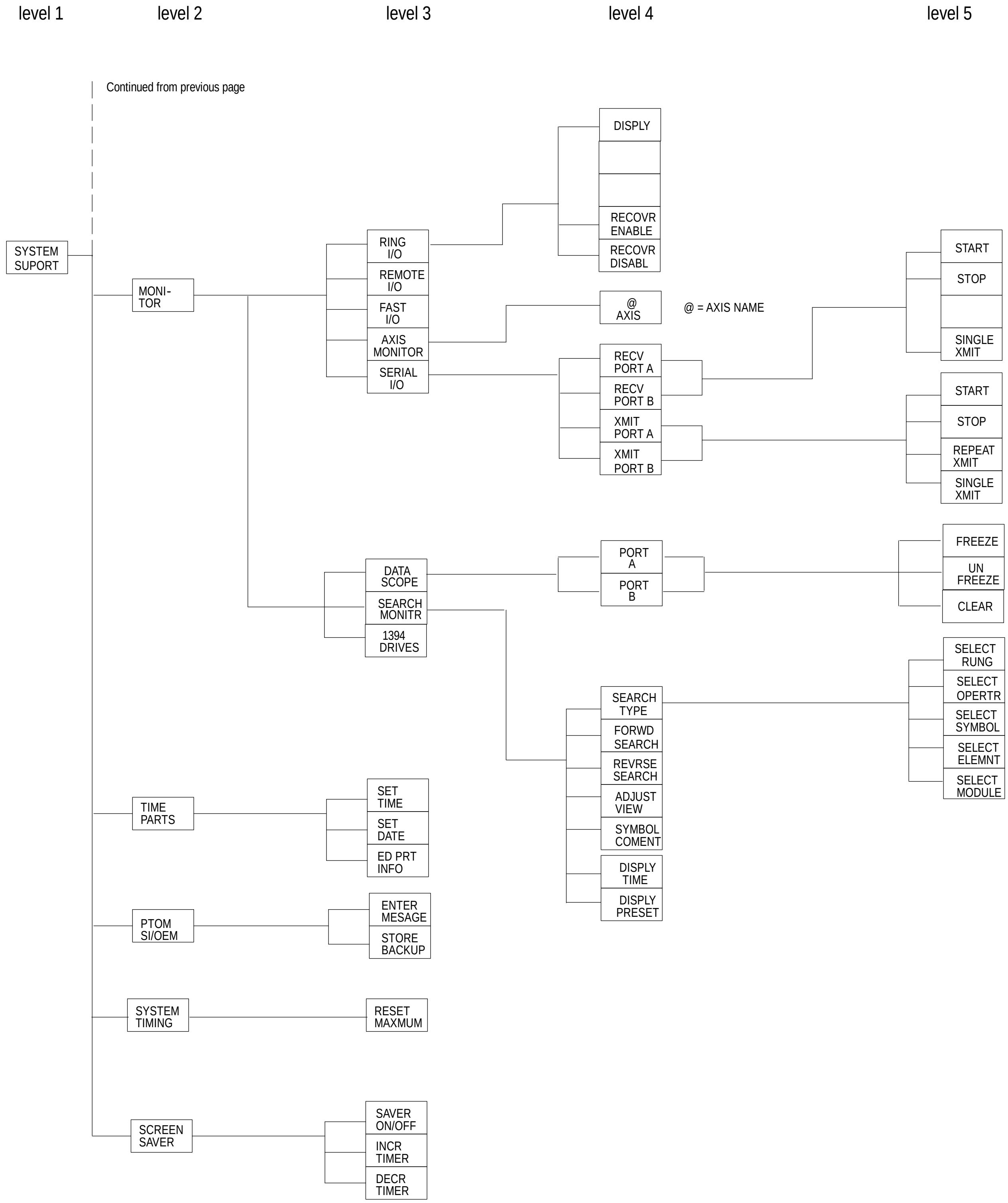


SYSTEM SUPPORT

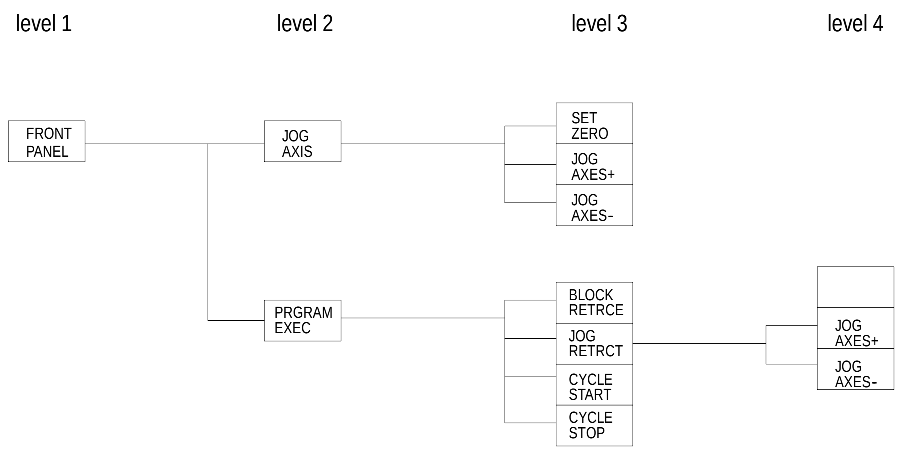


Continued on next page

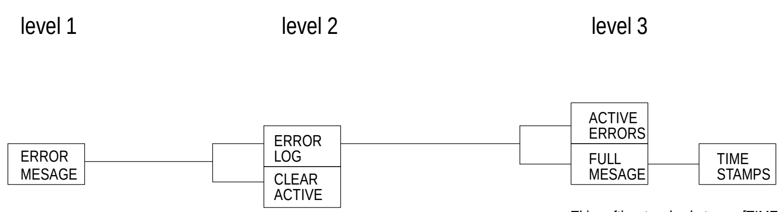
SYSTEM SUPPORT (continued)



FRONT PANEL

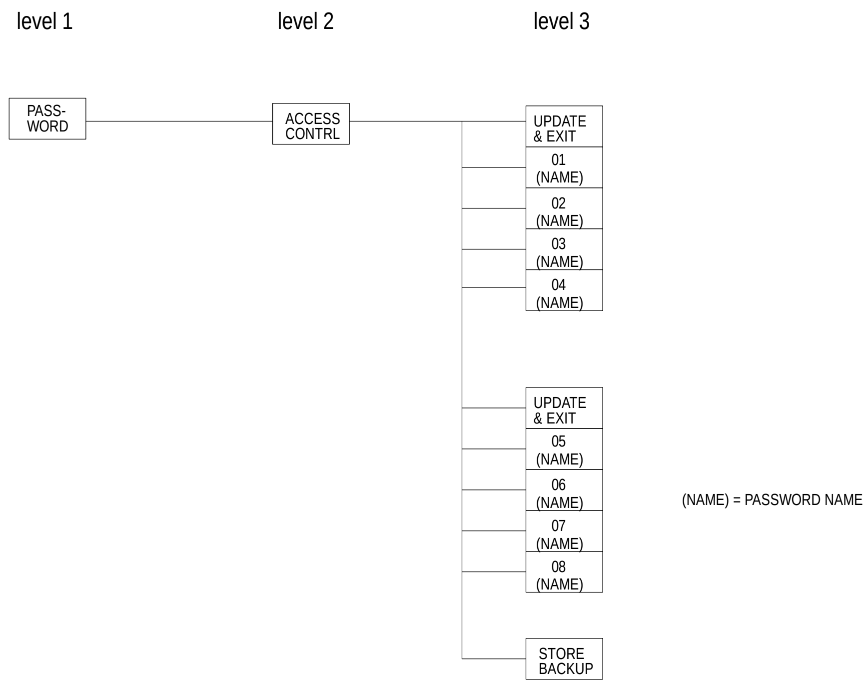


ERROR MESAGE

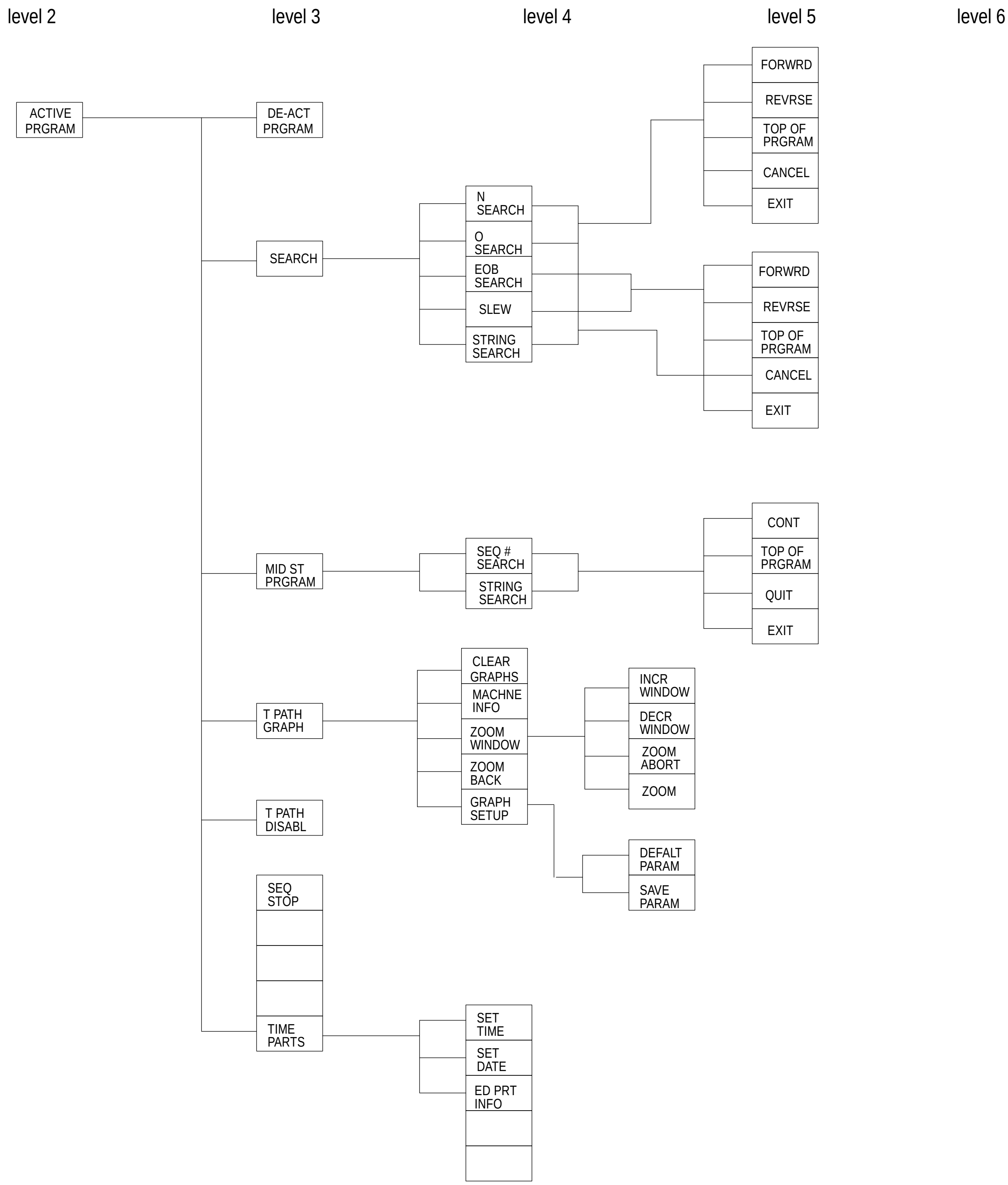


This softkey toggles between [TIME STAMPS] and [FULL MESAGE]

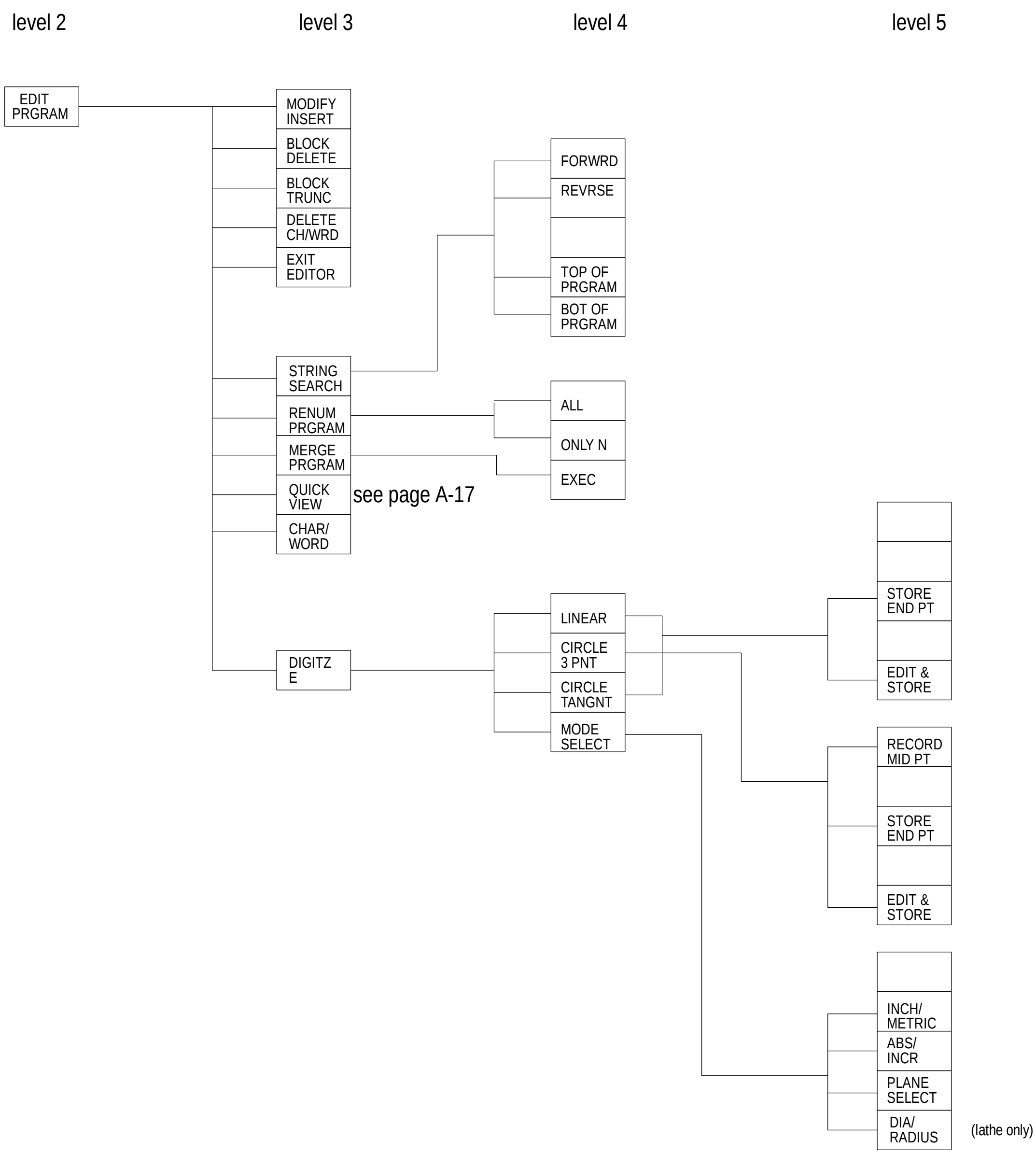
PASSWORD


level 1
level 2
level 3
PASS-
WORD
ACCESS
CONTRL
UPDATE
& EXIT
01
(NAME)
02
(NAME)
03
(NAME)
04
(NAME)
UPDATE
& EXIT
05
(NAME)
06
(NAME)
07
(NAME)
08
(NAME)
STORE
BACKUP
(NAME) = PASSWORD NAME

ACTIVE PRGRAM



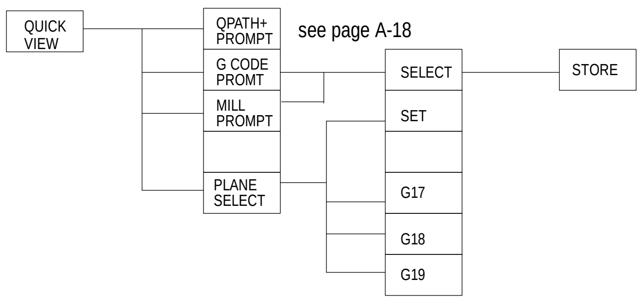
EDIT PRGRAM


level 2
level 3
level 4
level 5
EDIT PRGRAM
MODIFY INSERT
BLOCK DELETE
BLOCK TRUNC
DELETE CH/WRD
EXIT EDITOR
FORWRD
REVRSE
TOP OF PRGRAM
BOT OF PRGRAM
STRING SEARCH
RENUM PRGRAM
MERGE PRGRAM
QUICK VIEW
CHAR/ WORD
see page A-17
ALL
ONLY N
EXEC
DIGITZ E
LINEAR
CIRCLE 3 PNT
CIRCLE TANGNT
MODE SELECT
STORE END PT
EDIT & STORE
RECORD MID PT
STORE END PT
EDIT & STORE
INCH/ METRIC
ABS/ INCR
PLANE SELECT
DIA/ RADIUS
(lathe only)

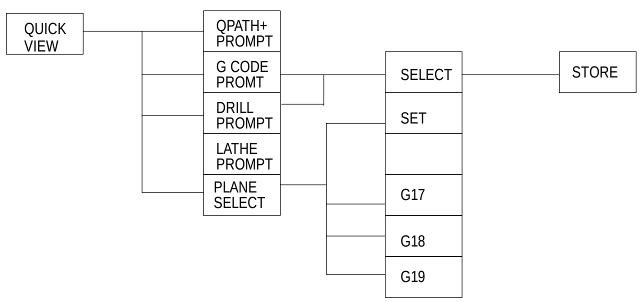
QUICK VIEW

level 3 level 4 level 5 level 6

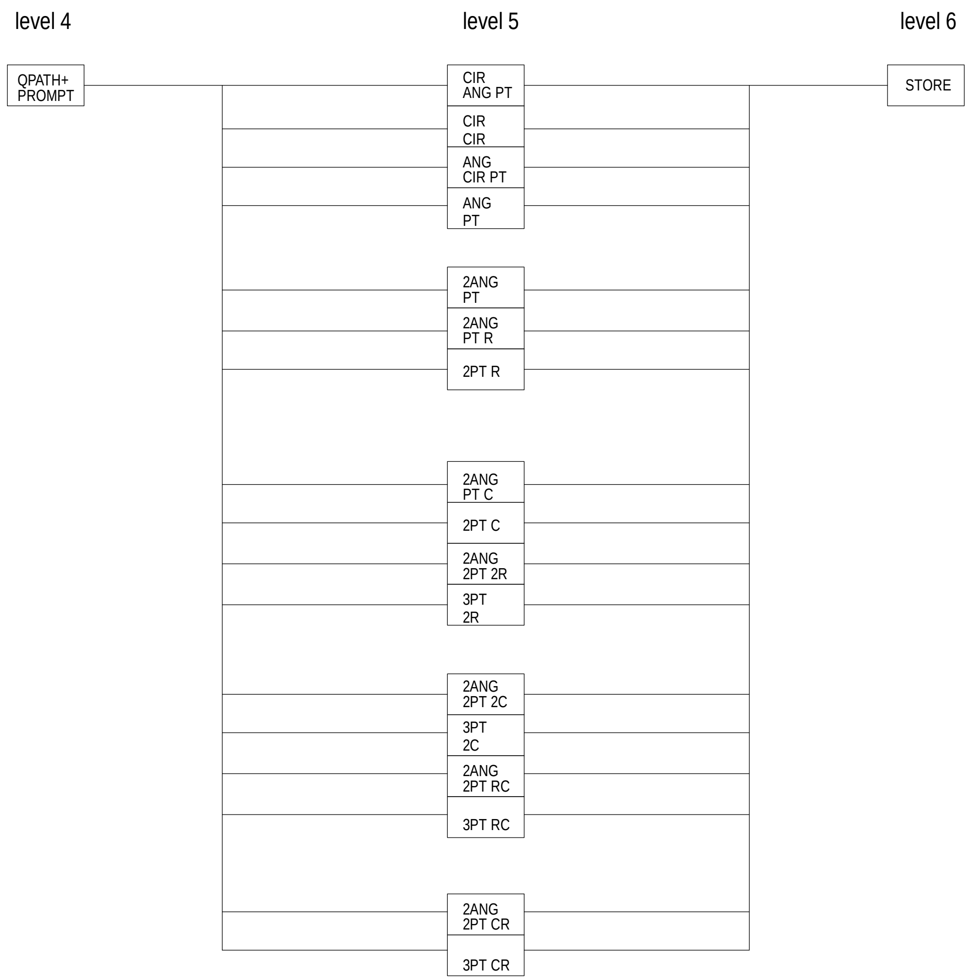
MILL



LATHE



QPATH+ PROMPT



END OF APPENDIX

Appendix B

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 MESAGE} 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 detercted 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 ± 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 then 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 "ok" 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 an 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 PRGRAM} 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 reseat 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 reseat 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 reseat 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.
FOREGROUND 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	
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 postion/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 configurated 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 INFEED (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 though 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 taping 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 a interrupt program and execute a MDI command. MDI commands cannot be executed during the execution of a 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 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-rogram 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 9/Series 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 dueled 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
O	
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 i s 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 date 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 SUPORT}, {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 it's 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 PRGRAM} 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 {PRGRAM COMENT} 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/ ≤ Motor speed at starting Id ≤ Motor speed at Id Break Point ≤ 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 their 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 temperture 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. • 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 CNC 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

Appendix C

G-Code Table

Overview

This appendix lists the G–codes for the 9/Series surface and cylindrical grinder. This table is presented numerically by G–code along with a brief description of their use. These G–codes are described in detail within this manual.

The group number given in the table refers to modality. Group 00 G–codes 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 that are not in group 1.

Table C.A
Grinder G-code Table

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
G00	G00	01	Rapid Positioning	Modal
G01	G01		Linear interpolation	
G02	G02		Circular / helical interpolation CW	
G03	G03		Circular / helical interpolation CCW	
G04	G04	00	Dwell	Non-modal
G05	G05		Send Command and Wait for Return Status (used with 9/Series Data Highway Plus Communication Module)	
G05.1–G05.4	G05.1–G05.4		Send Command without Waiting for Return Status (used with 9/Series Data Highway Plus Communication Module)	
--	G07	18	Programming using radius values	Modal
--	G08		Programming using diameter values	
G09	G09	00	Exact stop	Non-modal
G10L2	G10L2		Setup Work Coordinate Offset Table	
G10L10	G10L10		Setup Tool Offset Values Geometry Table	
G10.2L1	G10.2L1		Communication Configuration Table (used with 9/Series Data Highway Plus Communication Module)	
G10.2L2	G10.2L2		Output Command Table (used with 9/Series Data Highway Plus Communication Module)	
G10.2L3	G10.2L3		Download Configuration Information (used with 9/Series Data Highway Plus Communication Module)	
G12.1	G12.1	21	Spindle 1 Controlling	Modal
G12.2	G12.2		Spindle 2 Controlling	

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
G12.3	G12.3		Spindle 3 Controlling	
G13	G13	00	QuickPath Plus use first intersection	Non-modal
G13.1	G13.1		QuickPath Plus use second intersection	
G14	G14	19	Disable Scaling	Modal
G14.1	G14.1		Enable Scaling	
G17	G17	02	Axis plane select (XY)	
G18	G18		Axis plane select (ZX)	
G19	G19		Axis plane select (YZ)	
G20	G20	01	Single pass O.D. and I.D. roughing	
G22	G22	04	Programmable Zone 2 and 3 (On)	
G22.1	G22.1		Programmable Zone 3 (On)	
G23	G23		Programmable Zone 2 and 3 (Off)	
G23.1	G23.1		Programmable Zone	
G24	G24	01	Single pass rough facing cycle	
G27	G27	00	Machine home return check	Non-modal
G28	G28		Automatic return to machine home	
G29	G29		Automatic return from machine home	
G30	G30		Return to secondary home	
G31	G31		External skip function 1	
G31.1	G31.1		External skip function 1	
G31.2	G31.2		External skip function 2	
G31.3	G31.3		External skip function 3	
G31.4	G31.4		External skip function 4	
G33	G33	01	Constant lead thread grinding	Modal
G34	G34		Variable lead thread grinding	
G36	G36	22	Short block Acc/Dec enable	
G36.1	G36.1		Short block Acc/Dec disable	
G37	G37	00	Tool gauging skip function # 1	Non-modal
G37.1	G37.1		Tool gauging skip function # 1	
G37.2	G37.2		Tool gauging skip function # 2	
G37.3	G37.3		Tool gauging skip function # 3	
G37.4	G37.4		Tool gauging skip function # 4	
G39	G39	20	Dresser/Wheel compensation (linear generated blocks)	Modal
G39.1	G39.1		Dresser/Wheel radius compensation (circular gen. blocks)	
G40	G40	07	Dresser/Wheel radius compensation cancel	
G41	G41		Dresser/Wheel radius compensation, left	
G42	G42		Dresser/Wheel radius compensation, right	

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
G47	G47	24	Linear Acc/Dec in All Modes	Modal
G47.1	G47.1		S-Curve Acc/Dec for Positioning and Exact Stop Mode	
G47.9	G47.9		Infinite Acc/Dec (No Acc/Dec) (AMP-selectable only)	
G48	G48	00	Reset Acc/Dec to Default AMPed Values	Non-modal
G48.1	G48.1		Acceleration Ramp for Linear Acc/Dec Mode	
G48.2	G48.2		Deceleration Ramp for Linear Acc/Dec Mode	
G48.3	G48.3		Acceceleration Ramp for S-Curve Acc/Dec Mode	
G48.4	G48.4		Dececeleration Ramp for S-Curve Acc/Dec Mode	
G48.5	G48.5		Programmable Jerk Value	
G50.1	G50.1	11	Programmable mirror image cancel	
G51.1	G51.1		Programmable mirror image	
G52	G52	00	Offset coordinate system zero point	Non-modal
G53	G53		Motion in Machine coordinate system	
G54	G54	12	Preset Work Coordinate System 1	Modal
G55	G55		Preset Work Coordinate System 2	
G56	G56		Preset Work Coordinate System 3	
G57	G57		Preset Work Coordinate System 4	
G58	G58		Preset Work Coordinate System 5	
G59	G59		Preset Work Coordinate System 6	
G59.1	G59.1		Preset Work Coordinate System 7	
G59.2	G59.2		Preset Work Coordinate System 8	
G59.3	G59.3		Preset Work Coordinate System 9	
G61	G61	13	Exact stop mode	
G62	G62		Automatic corner override mode	
G64	G64		Cutting mode	
G65	G65	00	Paramacro Call	Non-modal
G66	G66	14	Modal paramacro call	Modal
G66.1	G66.1		Modal paramacro call	
G67	G67		Modal paramacro call cancel	
G68	G68	16	Part rotation	
G69	G69		Part rotation cancel	
G70	G70	06	Inch input	
G71	G71		Metric input	
G80	G80	09	Canned Cycle Cancel	
G81	G81		Reciprocation on	
G81.1	G81.1		Reciprocation with predress on	
G82	--		Plunge grind cycle	
--	G82		Incremental face grind cycle, axis 1 plunge	

Surface Grinder	Cylindrical Grinder	Group Number	Function	Modal or Non-modal
G82.1		09	Plunge grind with predress	Modal
--	G82.1		Incremental face grind with predress, axis 1 plunge	
G83	--		Incremental plane grind, axis 1 plunge	
--	G83		Incremental plunge grind, axis 2 plunge	
G83.1	--		Incremental plane grind, axis 1 plunge with predress	
--	G83.1		Incremental plunge grind, axis 2 plunge with predress	
G84	--		Incremental plane grind, axis 2 plunge	
--	G84		Multi-pass face cycle	
G84.1			Incremental plane grind, axis 2 plunge with predress	
--	G84.1		Multi-pass face cycle with predress	
G85	--		Continuous plane grind, axis 1 plunge	
--	G85		Multi-pass diameter plunge	
G85.1	--		Continuous plane grind, axis 1 plunge with predress	
--	G85.1		Multi-pass diameter plunge with predress	
G86	--		Continuous plane grind, axis 2 plunge	
--	G86		Shoulder grind cycle	
G86.1			Continuous plane grind, axis 2 plunge with predress	
--	G86.1		Shoulder grinder cycle with predress	
--	G87		Face plunge shoulder cycle	
--	G87.1		Face plunge shoulder cycle with predress	
--	G88		Diameter plunge shoulder cycle	
--	G88.1		Diameter plunge shoulder cycle with predress	
G90	G90	03	Absolute mode	
G91	G91		Incremental mode	
G92	G92	00	Maximum CSS spindle RPM, coordinate offset	Non-modal
G92.1	G92.1		Cancel offsets	
G92.2	G92.2		Cancel select offsets and G92 presets	
G93	G93	05	Inverse time feedrate mode	Modal
G94	G94		Feed per minute feedrate mode	
G95	G95		Feed per revolution feedrate mode	
G96	G96	17	Constant surface speed on	
G97	G97		RPM spindle speed mode (CSS off)	

END OF APPENDIX

Numbers

A

B

C

D

G

H

I

Q

R

S

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-UM514A-EN-P - October 2000
Supersedes Publication xxxx-1.1 – March 1992

PN 176955