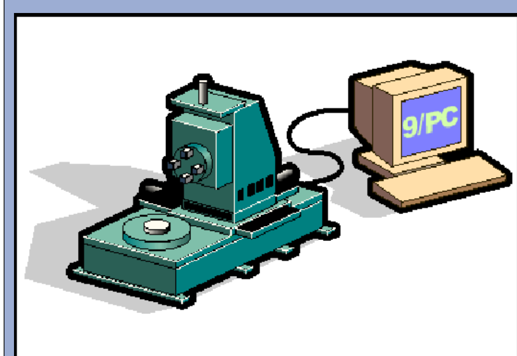
Allen-Bradley

9/PC CNC Mill

Operation and Programming Manual



Important User Information

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

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

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

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.

- Capability of using two 1394 drives
- Addition of 1394-related 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.

Table of Contents

Chapter 1
Using this Manual

Chapter 2
Basic Control Operation

Chapter 3
Offset Tables and Setup

Chapter 4
Manual/MDI Operation Modes

Chapter 5
Editing Programs Online

Chapter 6
Editing Programs Offline

Chapter 7
Running a Program

Chapter 8
Data Display

Chapter 9
Introduction to Programming

Chapter 10
Basic Control Operation

Chapter 11
Overtravels and Programmable Zones

Chapter 12
Coordinate Control

Chapter 13
Axis Motion

Chapter 14
Using QuickPath Plust

Chapter 15
Chamfering and Corner Radius

Chapter 16
Spindles

Chapter 17
Programming Feedrates

Chapter 18
Dual Axis Operation

Chapter 19
Tool Control Functions

Chapter 20
Cutter Diameter Compensation (G40, G41, G42)

Chapter 21
Using Pocket Milling Cycles

Chapter 22
Using Post Milling Cycles

Chapter 23
Using Hemisphere Milling Cycles

Chapter 24
Irregular Pocket Milling Cycles

Chapter 25
Milling Fixed Cycles

Chapter 26
Skip and Gauge Probing Cycles

Chapter 27
Paramacros

Chapter 28
Program Interrupt

Appendix A
Softkey Tree

Appendix B
Error and System Messages

Appendix C
G–code Tables

Using this Manual

Chapter Overview

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

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

Audience

We created this manual for Allen-Bradley 9/PC CNC programmers and/or operators. We assume that you are familiar with the following:

- operating and programming a CNC
- 9/PC standard front panel
- operating a personal computer
- AMP programming
- the Offline Development System (ODS)

Manual Design

We divided the manual this way:

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

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

Table 1.A provides a summary of each chapter.

Table 1.A
Manual Organization

Chapter	Title	Summary
1	Manual Overview	Manual overview, intended audience, definition of key terms, how to proceed.
2	Basic Control Operation	A brief description of the control's basic operation including power up, MTB panel, operator panel, access control, and E-Stop.
3	Offset Tables and Setup	Basic setup of the offset table, other initial operating parameters.
4	Manual and MDI Operation	How to use the manual operate mode including, homing the machine, jog hand-wheel, jog continuous, and jog increment. Also covered are the basics for MDI operation.
5	Editing Programs On Line	How to create, edit,and save a part program on line.
6	Editing Part Program Off Line	How to create, edit, and save a part programs from ODS offline.
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	Data Display	How to access and interpret the different position displays. How to use the Quick Check and Active Program graphics features.
9	Introduction to Programming	Structure and format of the programming language for the control.
10	Coordinate System Offsets	Machine coordinate system, Preset Work coordinate systems, logic offsets, and external offsets
11	Overtravels and Programmable Zones (G22, G23)	Hardware and software overtravels, programmable zone 2 (G22, G23), programmable zone 3 (G22.1, G23.1), and resetting overtravels
12	Coordinate Control	Describes absolute/incremental modes, inch/metric modes, radius/diameter modes, and scaling
13	Axis Motion	G-words define how the tool is positioned to the endpoint of a move. Also sections on automatic machine home, dwell, mirroring, and axis clamp
14	QuickPath Plus	Describes QuickPath Plus programming
15	Using Chamfers and Corner Radius	Describes the ,C- and ,R-words programmed for chamfering and corner radius
16	Spindles	Describes spindle speed control, spindle orientation, spindle direction, and Virtual C axis
17	Programming Feedrates	Describes acc/dec, AMP-assigned feedrates, feedrate control, short block acc/dec
18	Dual Axis Operation	Describes parking, homing, programming, offset management for a dual axis
19	Tool Control	Selecting a tool. Activating and deactivating tool length offsets. Also tool control features such as Random Tool and Tool Life Management.
20	Cutter Compensation	Describes the Tool Tip Radius Compensation feature (TTRC) that offsets for different tool diameters.
21	Using Pocket Milling Cycles	Describes the fixed cycles (canned cycles) for drilling operations and the G-words and parameters used to define them.
22	Using Post Milling Cycles	
23	Using Hemisphere Milling Cycles	
24	Using Irregular Pocket Milling Cycles	
25	Milling Fixed Cycles	
26	Skip and Gauging Cycles	Describes the 9/PC probing features. Includes the tool measuring gauge feature.
27	Paramacros	Describes paramacros including calling, arithmetic functions, looping, decision making
Appendix A	Softkeys	Describes softkeys and their functions for softkey levels 1 and 2. Also the softkey tree displaying all levels of softkeys and their location is shown.
Appendix B	Error and Operator Messages	An alphabetical listing of 9/Series and 9/PC system messages with brief descriptions.
Appendix C	G and M Code Tables	Lists the G-codes used to program the control.

Reading this Manual

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

- All explanations, illustrations, and charts presented are based on standard CNC functions. Operations may differ from the basic information provided in this manual depending on the configuration of the machine tool. For details, refer to the manuals prepared and supplied by the system installer.
- Some of the softkey functions may be purchased as optional features. This manual assumes that all of the optional features have been purchased.
- Explanations and illustrations are presented based on the movement of the cutting tool on a fixed workpiece.
- The control allows the use of any alphabetic character for expressing a numerically controlled axis. This manual uses X, Y, and Z for the first, second, and third axes on the basic coordinate system respectively. I, J, and K represent the integrand words for the axes.
- The term AMP is an abbreviation for Adjustable Machine Parameters. These parameters are used to configure a control to a specific machine. Setting of AMP is usually done by the system installer.
- Key names designated between the [ ] symbols are found on your computer's keyboard.
- Key names designated between the { } symbols are softkeys found on the Basic Display Set.
- Switch and button names on the DeviceNet MTB panel are designated between the < > symbols.
- The term "logic" refers to the SoftLogix 5™ logic engine that processes signals to the 9/PC. It is usually programmed by the system installer.
- System Characteristics:
  - Metric
  - Absolute
  - IPM

Cautionary and Important Information

We indicate information that is especially important by the following:

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.

Reading this Manual

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

- All explanations, illustrations, and charts presented are based on standard CNC functions. Operations may differ from the basic information provided in this manual depending on the configuration of the machine tool. For details, refer to the manuals prepared and supplied by the system installer.
- Some of the softkey functions may be purchased as optional features. This manual assumes that all of the optional features have been purchased.
- Explanations and illustrations are presented based on the movement of the cutting tool on a fixed workpiece.
- 9/PC allows the use of any alphabetic character for expressing a numerically controlled axis. This manual uses X, Y, and Z for the first, second, and third axes on the basic coordinate system respectively. I, J, and K represent the integrand words for the axes.
- The term AMP is an abbreviation for Adjustable Machine Parameters. These parameters are used to configure the 9/PC. Setting of AMP is usually done by the system installer.
- Key names designated between the [ ] symbols are found on your computer's keyboard.
- Key names designated between the { } symbols are softkeys found on the Basic Display Set panel.
- Switch and button names on the DeviceNet MTB panel are designated between the < > symbols.
- The term "logic" refers to the SoftLogix 5™ ladder logic program that processes signals to the 9/PC. It is usually programmed by the system installer.
- System Characteristics:
  - Metric
  - Absolute
  - IPM

Terms and Conventions

In this manual, we use acronyms and shortened product names and features. Shortened terms used in this manual include:

- 9/PC — A CNC PCI card that provides a full-featured open 9/Series motion control solution.
- AMP — Adjustable Machine Parameters
- API — Application Programming Interface

- BDS — Basic Display Set. This software provides the user interface between your PC and CNC by emulating 9/Series standard screens. The software allows you to control, program, position, and monitor your 9/PC.
- CNC — Computerized Numerical Control
- Control — The 9/PC CNC
- CPU — Central processing unit
- DDE — Dynamic data exchange. This allows two applications to share data.
- Device — Any DeviceNet-compatible device
- DRAM — Dynamic random access memory
- E-Stop — Emergency stop
- Flash Memory — Nonvolatile, programmable memory that resides on the CPU board. This memory backs up such things as paramacro tool offsets and work coordinates, and AMP and retains information even after a power failure.
- Hard drive — Any storage location defined by your system installer. In this manual, hard drive refers to the storage space located on the PC.
- HMI — Human-machine interface
- Host Computer — The PC housing your 9/PC card.
- I/O — Input/output
- Layered softkeys — The row of keys located on the bottom of the line and cycle editor screens
- OCI — Open Control Interface. Originating as a 9/Series product, OCI provides an open HMI for existing CNCs.
- ODS — Offline Development Software. This is application software used to create, download, and upload configuration for your 9/PC CNC.
- OLE — Object-linked embedding. This standard provides the ability to link/embed objects created with one application to another application.
- OPC — OLE for process control. This is a means of data exchange between applications.
- Logic — The Logic engine the SoftLogix 5 engine
- PC — Personal Computer
- PCI — Peripheral component interconnect. This is a means of connecting peripheral devices to your PC.
- PCIDS — Peripheral component interconnect device scanner
- PELV — Protected Extra Low Voltage
- Project — A directory that stores configuration, interface, and motion control files for a particular control or application.

- RAM — Random access memory
- SELV — Safety Extra Low Voltage
- SERCOS — Serial Real-time Communication System
- Softkey — A row of keys located directly below the operator panel screen on the BDS.
- System installer — The company or contractor responsible for installing your 9/PC control.
- Topic Name — The name designated to your 9/PC CNC that is entered in the Common tab of the Configuration Manager. This topic name is shared by the API, AMP, and Logic.
- UPS — Uninterruptable Power Supply

Additional Publications

The following publications are available from Allen-Bradley and can be helpful when using your 9/PC system.

Pub. No.	Document Name
8520-1.9	9/PC CNC Product Profile
MCD-5.1	9/Series CNC Offline Development System (ODS) User's Manual
8520-6.6	9/Series CNC API Developer's Guide
8520-9.1	9/PC CNC Installation and Integration Manual
8520-9.2	9/PC CNC Logic Reference Manual
8520-9.3	9/PC CNC AMP Reference Manual
8520-9.4	9/PC CNC Lathe Operation and Programming Manual
8520-9.5	9/PC CNC Mill Operation and Programming Manual

Technical Support

Before you contact technical support, try to find the answer to your question in your 9/PC-related documentation (refer to the table on page 1-6). If you can not find the answer to your question, technical support is available on the World Wide Web and by phone:

- World Wide Web: http://SUPPORTBBS.RA.ROCKWELL.COM
- Telephone Technical Support: 1 (440) 646-6800,
  available 8:00 AM to 5:00 PM EST, Monday through Friday
  (with the exception of holidays)

When You Call

When you call, make sure you are at the computer running the 9/PC CNC. Be prepared to provide the following information:

Information	Location
9/PC Processor card serial number	Label on 9/PC processor card or via the *9/PC Configuration Manager* screen. In the latter case, click the Windows NT **Start** button. Select **Programs** ⇒ **Rockwell Automation** ⇒ **9pc** ⇒ **Configuration Manager**. If the 9/PC is running, press **{Stop 9/PC}**. **Important:** The CNC must be in E-Stop. Select the **9/PC Configuration** tab. Click **{Edit CNC}** and read the serial number from the opposite field. Click **{OK}**, select the **9/PC Operation** tab, and click **{Start 9/PC}** to restart your 9/PC.
SoftLogix 5 serial number	SoftLogix 5 activation disk label or from the the Registration field on the SoftLogix 5 Status Monitor screen, which appears when you click on the SoftLogix 5 LED status in the Windows NT system tray.
9/PC Executive software revision number	Press the **{SYSTEM SUPORT}** softkey on the BDS screen, the **{→}** softkey, and then the **{PTOM SI/OEM}** softkey. Alternatively, you can also read the 9/PC software version from the 9/PC Installation disk's label.
9/PC PC software revision number	Click the **{Version Information}** button on the **9/PC Configuration Manager** screen. To view this screen, click the Windows NT **Start** button ⇒ **Programs** ⇒ **Rockwell Automation** ⇒ **9pc** ⇒ **Configuration Manager**.
SoftLogix 5 software revision number	Go to the **SoftLogix 5 Status Monitor** screen by clicking on the SoftLogix 5 LED status in the Windows NT system tray. Select **{Config}**. This takes you to the **SoftLogix 5 Configuration Manager** screen. Click on the **{Version Info}** button. Alternatively, you can also read SoftLogix 5 version number from the SoftLogix 5 Installation disk's label.
ODS software revision number	Start ODS. Press the **[F1]** key. Select "About" with the arrow cursor keys, and press **[Enter]**.
Miscellaneous • A description of what was happening with the system when the problem occurred • A description of how you tried to solve the problem • The exact wording of any messages that appear on the screen, the CNC Error Log, and Windows NT Event Log • The hardware you are using (i.e., PC make and model, type of I/O)	

END OF CHAPTER

Basic Control Operation

Chapter Overview

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

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

Proper Startup and Shutdown of the 9/PC CNC

Prior to starting the Basic Display Set, you must start your 9/PC via the 9/PC Configuration Manager.

Starting the 9/PC

To properly start your 9/PC CNC:

1. Click the **Windows NT Start** button and select **Programs**.
2. Click on **Rockwell Automation** to choose the **9pc** option.
3. Select **Configuration Manager** to start the 9/PC Configuration Manager.
4. Select the **`{Start 9/PC}`** button.

   The **`{Start 9/PC}`** button becomes ghosted when 9/PC is running.

Important: After initially loading the 9/PC executive and the 9/PC card, on the 9/PC Configuration Manager, `{Start 9/PC}` and `{Stop 9/PC}` appear ghosted and NOT CONFIGURED appears in the **Serial Number** dialog box. Once you make the association between the CNC and the serial number, `{Start 9/PC}` becomes available (unghosted). Refer to your *9/PC Installation and Integration Manual* (8520-9.1) for more information.

Stopping the 9/PC

There are two preferred methods through which your 9/PC can be shut down: via the Configuration Manager or via Windows.

Stopping the 9/PC Using the Configuration Manager

1. Select the Windows NT **Start** button and select **Programs**.
2. Click on **Rockwell Automation** to choose the **9pc** option. Select **Configuration Manager** to start the 9/PC Configuration Manager.
3. Select the `{Stop 9/PC}` button.

Stopping the 9/PC Using Windows NT

1. Select the Windows NT **Start** button.
2. Click on the **Shut Down...** option.

ATTENTION: Improper and uncontrolled shutdowns will cause loss of BBU data. If your 9/PC is not shut down via one of the user-controlled methods described on page 2-2, the system uses the last known BBU data profile (i.e., tool offsets, paramacros, axis calibration, work coordinates, tool management, error log, and AMP data). Any lost data will have to be regenerated by the user.

Uncontrolled Shutdowns

Although the 9/PC does not require the use of battery backup memory, we recommend that you use an uninterruptible power supply (UPS) to protect your CNC data from uncontrolled shutdown sequences (e.g., power loss).

The UPS is an external device that is able to keep your PC running for a prescribed amount of time after the main power is shut down, while ensuring that no vital CNC data is lost after a system power off. For information about restoring data after an uncontrolled shutdown, refer to your UPS documentation. For more information about UPS devices with relation to your 9/PC, refer to your *9/PC Integration and Installation Manual* (8520-9.1).

In the event of a power loss or an improper shutdown, the message “BBU SAVE FAILED, USING LAST VALID DATA” indicates lost BBU data. You may also review the **Windows NT Event Log**. If the Event Log indicates that your system was improperly shut down, the BBU data that you use will have the last known profile that your system stored. Any lost data will have to be regenerated by the user.

What is the Basic Display Set (BDS)?

The Basic Display Set is a compiled executable Visual Basic™ program written to provide a 9/Series compliant operator interface on the PC. With BDS, you can create, edit, and save part programs on your PC. The editor has a new interface and functions differently than the 9/Series standard front panel version. Refer to page 5-8 for more information about navigating through the editor.

Important: If you want to configure the displays for your specific applications, the Visual Basic source code is available. See your system installer for details.

Use the Basic Display Set in the Windows NT™ operating environment to provide a graphical interface to the CNC. The following is a list of enhanced features:

- Updated editor graphics and editing capabilities
- Extended part program storage on the PC
- AMP development
- Optional application program interface (API package) customized user displays and graphics can be created

Launching the Basic Display Set (BDS)

Basic Display Set screens are installed on your personal computer as part of the 9/PC CNC software installation.

Your application may have additional screens created by your system integrator that are not a part of BDS. This manual describes only those screens that are a part of BDS.

Important: Although you may operate BDS without having any other software activated, you should complete the recommended software installations (including configuration) to use BDS with your 9/PC system. Refer to the *9/PC Installation and Integration Manual* for more information about software installation.

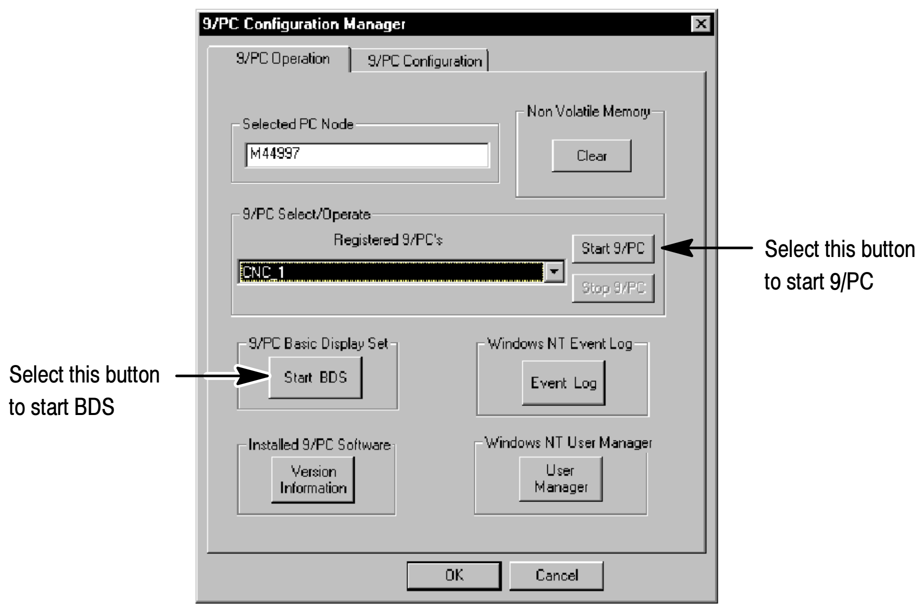
You must activate the Basic Display Set through the 9/PC Configuration Manager. To launch BDS:

1. Click the **Windows NT Start** button and select **Programs**. Click on **Rockwell Automation** to choose the **9pc** option. Select **Configuration Manager** to start the 9/PC Configuration Manager.
2. Select the **9/PC Operation** tab. Select the `[Start 9/PC]` button to activate the CNC.

   Important: You must start the 9/PC CNC prior to starting BDS. Activating BDS before starting 9/PC, causes the following message to appear:

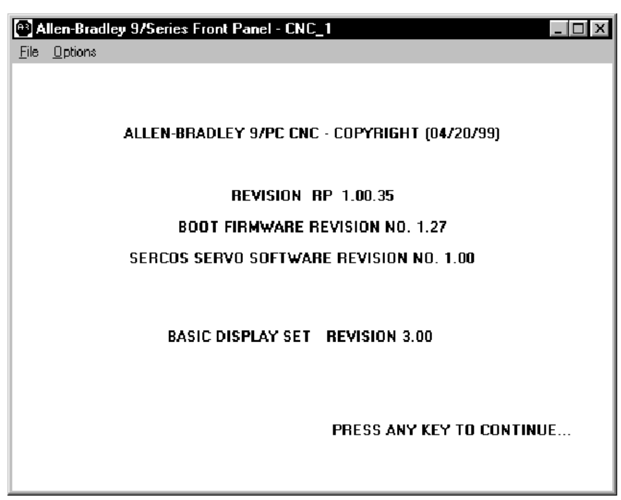
3. Select the **`[Start BDS]`** button on the 9/PC Configuration Manager. The **Allen-Bradley OCI File Handler** and **OCI Data Server** icons should appear on the task bar.

ATTENTION: If these program icons do not appear, check the Event Log for messages that may indicate the reason(s) for the icon's absence. If the problem persists after making the necessary changes, contact A-B Customer Support.

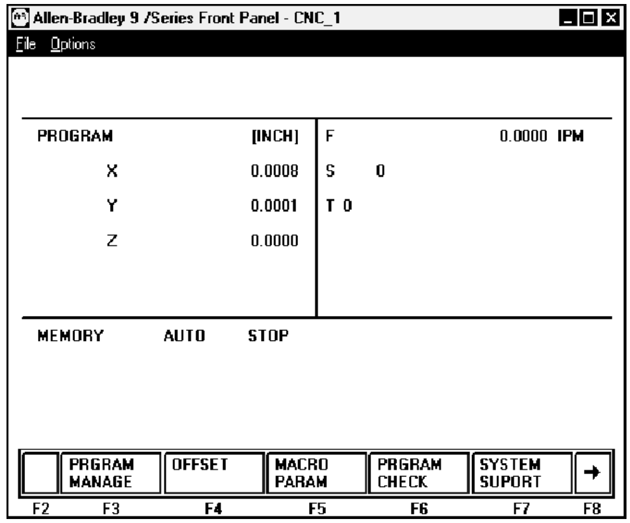


A Tour of the Basic Display Set

Most of the screens look very similar to those available with the 9/Series standard front panel. The first screen that appears is the power turn-on screen, similar to the one below.



Press any key to clear this screen. You see the screen that displays the axis position:



Power Up Conditions

When powering up BDS, you have the same screen size, password level, last active program directory, and font size as in the previous session.

Pulldown Menus

At the top of the BDS window is a menu bar, which contains the File and Options pulldown menus. These two pulldown menus allow you to manipulate BDS.

To access the File or Options pulldown menus:

1. Point your mouse or pointing device to the menu name.
2. Press your left mouse button to display the contents below the menu item.
3. Point to the desired menu item and release the mouse button.

You can also access the pulldown menus using the keyboard:

Use this Key Combination:	To Select the:
[**ALT**] + **F**	File menu
[**ALT**] + **O**	Options menu

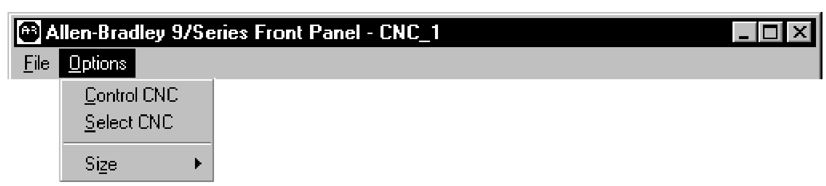
File Menu

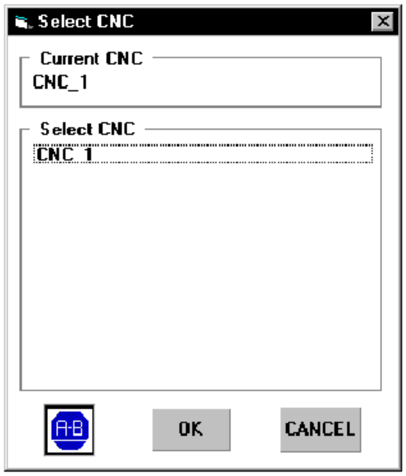
Selecting the **`File`** menu allows you to access the exit option. Choosing **`Exit`** closes BDS.

ATTENTION: Closing BDS **DOES NOT** stop the 9/PC.

Options Menu

Selecting the **`Options`** menu gives you access to **`Control CNC`**, **`Select CNC`**, and **`Size`** options.





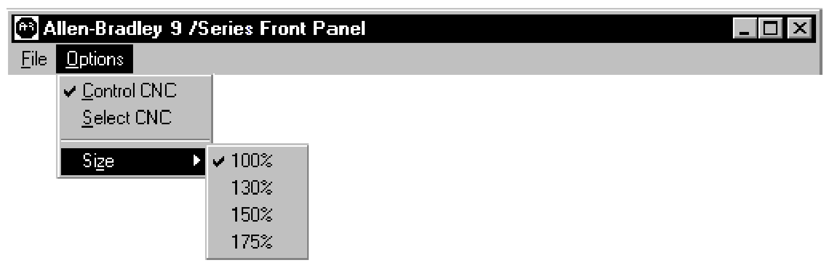
Select CNC – allows you to change the CNC that BDS is connected to, while displaying the name of each CNC configured in your 9/PC Configuration Manager file. To select a specific CNC:

- click on the CNC name in the Select CNC box with your pointing device. Choose the OK button on the bottom of the screen. Your selection is now activated.

  OR
- double click on the desired CNC name in the Select CNC box. Your selection is now activated.

To cancel your selection, choose the CANCEL button on the bottom of the Select CNC screen.

Size – allows you to change the width of the window you are currently working in. Window widths are based on the configured resolution of your computer monitor. The higher the resolution, the larger selection of window sizes you have.



Important: The size option is only available while softkey level 1 is displayed. Attempting to modify the screen size on any other softkey level causes an error.

Using the Softkeys

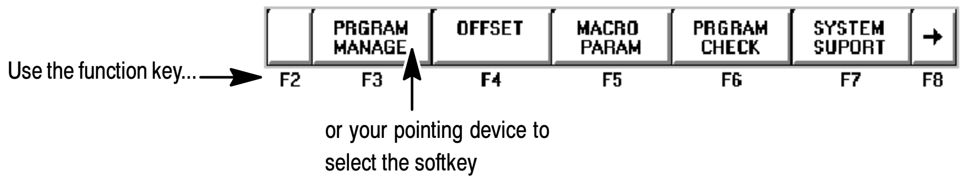
We use the term softkey to describe the row of 7 keys at the bottom of the BDS. Each function is displayed on the screen directly above the softkey. In this manual, softkey names are shown between the { } symbols.

BDS offers a variety of functions that can be initiated by using the softkeys. There are five softkeys whose function names are displayed in the softkey area at the bottom of the screen (lines 23-25). Move through BDS with the softkeys at the bottom of the screen. There are two ways to select these softkeys:

- using the function keys (F2–F8) on your keyboard
- with a pointing device (e.g., a mouse)

To access a softkey, use your keyboard or pointing device to select the appropriate function key.

`(softkey level 1)`



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

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

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

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

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

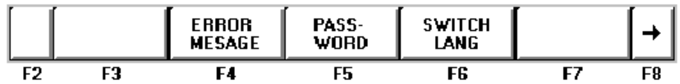
Important: The continue softkey is not active when the number of softkey functions on that level does not exceed 5.

For example:

`(softkey level 1)`

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

```
(softkey level 1)
```



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

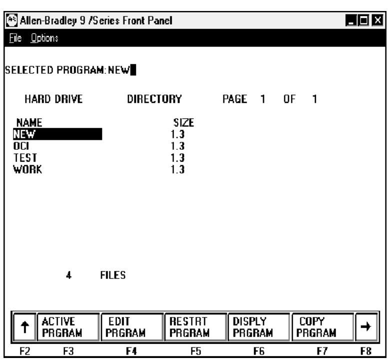
The softkey functions for level 1 and level 2 are explained in appendix A. Softkey functions for level 3, or higher, are explained in the sections that apply to their specific operations.

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

Inputting Text

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

Input Line



For BDS screens that require you to enter data, use the input line, as in the 9/Series standard front panel. To move through the input lines, use the arrow keys.

Cursor Operation:	Description:
Clearing error messages	To clear error messages from BDS, press the **[TAB]** or **[DEL]** key. Keep in mind that using your keyboard to clear an error message does not correct the error condition.
Deleting characters	To delete characters on the input lines move the cursor to the right of the character to delete. Pressing the **[BACKSPACE]** key deletes the character to the left of the cursor in the input line.
Deleting all characters on the input line	To delete all entered characters on the input lines press the **[BACKSPACE]** key while holding down the **[SHIFT]** key. All characters on the input line are deleted.
Sending information	To send information to the control from input line press the **[ENTER]** key. All information on the input line is sent to the control.

Performing CNC Functions from the PC Keyboard

The 9/Series operator panel has a keyboard with keys assigned for specific CNC functions. The Basic Display Set includes special key mapping so you can perform these functions on your PC keyboard:

If you want to:	press this on your PC keyboard:	the correlating key sequence on the 9/Series operator panel is:
enable the calculator function on the control	[F9]	Calculation Key [CALC]
delete the character to the left of the cursor on the input line(s)	[BACK SPACE]	[DEL]
toggle through the editor's layered softkeys	[F10]	N/A
clear the latest active error message	[DEL] or [TAB]	[CAN]
send the data entered on the input lines to the control	[ENTER] or [RETURN]	[TRANSMIT]
perform a block reset	[F11]	[RESET]
perform a control reset	[SHIFT] and [F11]	[RESET] + [SHIFT]
display mode selection softkeys	[F12]	[DISP SELECT]
enter an end-of-block character when writing an MDI program	[;]	[E.O.B]

The following subsections provide more information on specific function operations (e.g., calculator or block reset).

Reset Operations

Block Reset

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

Press the **[F11]** key on your keyboard to perform a block reset.

Control Reset

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

Hold down the **[SHIFT]** and **[F11]** keys to execute a control reset.

Calculator Function

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

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

Use the calculator function as follows:

1. Press the **[F9]** key on your keyboard. The “CALC:” prompt appears on the input line of the screen (line 2).
2. Enter a mathematical equation on the input line by pressing the desired keys on the operator panel.
3. Press the **[ENTER]** key to evaluate the expression. The answer to the expression is displayed on the input line.

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

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

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

Any data entered on the input lines can be edited as described on page 2-10.

To disable the calculator function, press the **[F9]** key again. The “CALC:” prompt is removed from the input line.

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

Table 2.A
Mathematical Operators

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

The control executes mathematical operations in this order:

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

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

Example 2.1
Mathematic Expressions

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

Table 2.B lists the function commands available with the **[F9]** key.

Table 2.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 Decimal to Coded Decimal
BCD	Conversion from Coded Decimal to Decimal
ROUND	Rounding Off (nearest whole number)
FIX	Truncation Down
FUP	Truncation Up
LN	Logarithms (natural log)
EXP	Exponent

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

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

Example 2.2
Format for [CALC] Functions

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

Example 2.3
Mathematical Function Examples

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

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

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

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

Paramacro Variables in CALC Operations

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

Example 2.4
Calling Paramacro Variables with the CALC Function

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

Navigating through the Display

If you choose, you can purchase a 9/PC CNC bundle pack that contains an LCD (10.4-in.) with or without a touch screen. Both have identical displays and graphics capabilities.

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

Lines:	Display information:
line 1 machine/ system message area	If an error occurs or a message is generated for any reason during machine operation or program execution, the control displays the corresponding machine/system message in this area. Only the highest priority, most current message is displayed here.
line 2 input lines	When you enter data using the keyboard, the control displays the characters corresponding to the keys pressed until you press the **[ENTER]** key.
lines 4-20 data display area	The control displays axis position data, listing of the part program, tool offset data, G-, M-, H-, T-, F-, S-, and D-codes, and other data, as determined by the selected display. Refer to chapter 8.
lines 21-22 Logic message area	The control displays any messages generated by the logic program in this area
lines 23-25 softkey display area	The control displays the currently available softkey functions in this area.

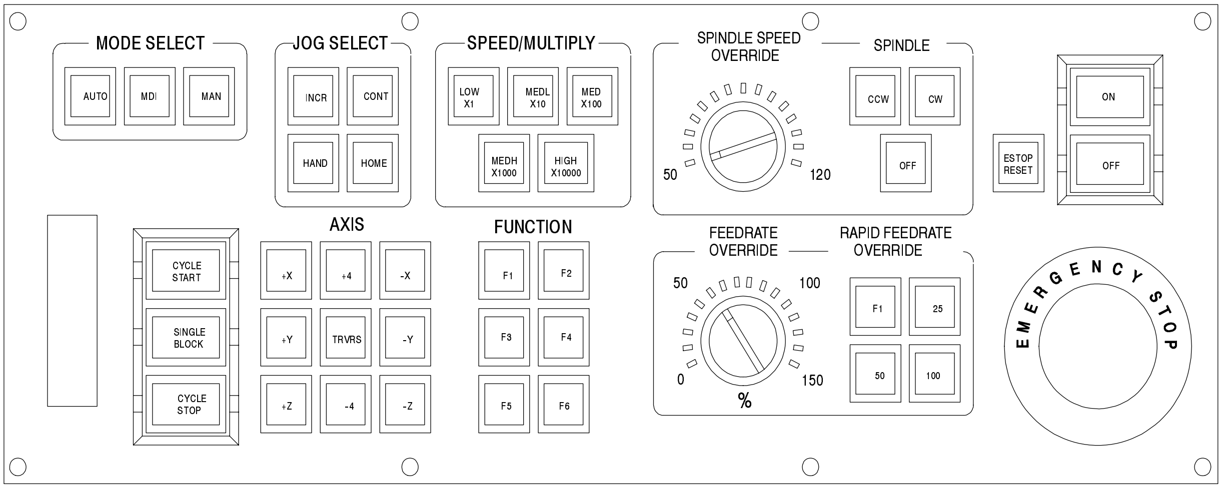
The Push-button MTB Panel

Figure 2.1 shows the push-button MTB panel. Table 2.C explains the functions of the buttons on the MTB panel. Optional or custom MTB panels may be used. Refer to the documentation prepared by your system installer for details.

Throughout this manual, we show switches and button names that are found on your MTB panel between the < > symbols. The MTB panel uses defaults when you turn on power to the control. Table 2.C contains these defaults.

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

Figure 2.1
Push-Button MTB Panel



19930

Table 2.C
Functions of the Buttons on the Push-button MTB Panel

Switch or Button Name	How It Works = Default for Push-Button MTB Panel
MODE SELECT	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 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: • 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.
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.

Switch or Button Name	How It Works = Default for Push-Button MTB Panel
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	Pressing this button causes the control to stop part program execution, MDI execution, or program check. 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 - F6	The functions for these buttons are assigned by the system installer. Refer to the documentation prepared by the system installer for details.
ON	Turns on power to the PC when connected to your 8520-OFC or a master control relay.
OFF	Turns off power to the PC when connected to your 8520-OFC or a master control relay.

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

Power Procedures

The basic procedure for turning power on and off to 9/PC is described in this section. Refer to the *9/PC Installation and Integration Manual* for more specific procedures.

Starting the Basic Display Set

Follow this procedure to turn on power to the control:

1. Make sure your PC is on. Visually check to make sure that the machine is in normal operating condition.
2. Click the **Windows NT Start** button and select **Programs**.
3. Click on **Rockwell Automation** to access the **9pc** option. Under **9pc**, click on **Configuration Manager**.

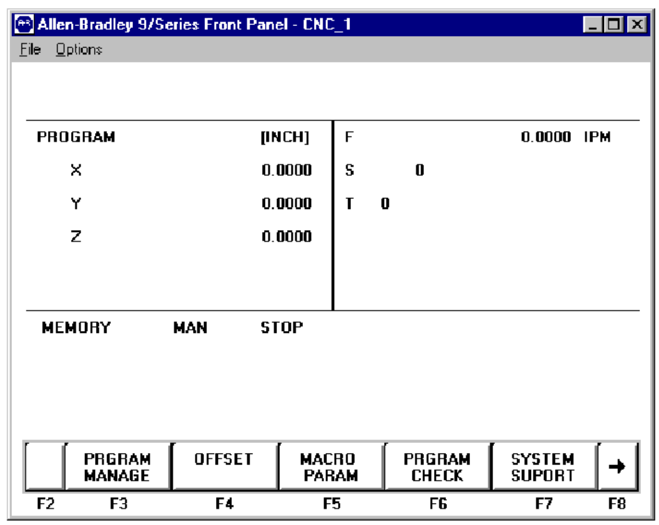
   Important: You must start the 9/PC prior to starting BDS. Starting BDS without starting on the Configuration Manager causes the error "CANNOT LINK TO ABOCISERVER|*YOURCNCNAME*" to display.

4. Select the **Start 9/PC** button on the Configuration Manager 9/PC Operation tab.
5. Select the **Start BDS** button on the Configuration Manager 9/PC Operation tab to launch BDS, the OCI File Handler, and the OCI DDE/OPC Data Server.

Important: If the screen does not display characters after you press the **Start BDS** button within a reasonable warm-up period (about 15 seconds), press `[Ctrl]` + `[Alt]` + `[Delete]` to stop BDS through the Windows NT Task Manager. Contact service personnel for further instructions.

After BDS is running, the BDS welcome screen appears and instructs you to press any button to continue.

You see the main menu screen:



The softkeys available on the main menu screen are referred to as "level 1" softkey functions. Some of the softkey functions are purchased as optional and may not appear exactly as shown. Refer to page 2-8 for more information about the softkeys.

Stopping the Basic Display Set

To turn off power to BDS, select **Exit** under the File menu.

> **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, you press the E-Stop button on the MTB panel, and you execute a proper shutdown (refer to page 2-1).

Control Conditions at Power-Up

After powering up the control or performing a control reset operation (see page 2-11), the control assumes a number of initial operating conditions. These are listed below:

- Initial Password Access is assigned to the level that was active when power was turned off (provided that level is a power-up level selected in access control). If the active level when power is turned off is not a power-up level, then the control defaults to the next lower level that is a power-up level. See page 2-23 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 logic program loaded in the system. An appropriate error message is displayed.
- The control defaults to **one** G-code from each of these groups (as set in AMP):

Modal Group:	G-code	
0	G47	Linear Acc/Dec
	G47.1	S-Curve Acc/Dec
	G47.9	Acc/Dec Disabled
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
10	G98	Init Level Return
	G99	R Point Level Return

12	NONE G54 G55 G56 G57 G58 G59.1 G59.2 G59.3	Preset Work Coordinate System 1 Preset Work Coordinate System 2 Preset Work Coordinate System 3 Preset Work Coordinate System 4 Preset Work Coordinate System 5 Preset Work Coordinate System 6 Preset Work Coordinate System 7 Preset Work Coordinate System 8 Preset Work Coordinate System 9
13	G61 G62 G63 G64	Exact Stop Mode Auto Corner Override Mode Tapping Mode Cutting Mode
18	G07 G08	Radius Programming Mode Diameter Programming Mode
20	G39 G39.1	Cutter Comp Linear Cutter Comp Rounding
22	G07 G08	Short Block Feed Clamped Short Block Full Clamped
25	G60 G60.1 G60.2	Synchronous mode Asynchronous mode Autosynchronous mode

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

To show the current operating conditions at any time, access the G-code status screen. Refer to Figure 8.13 for an example of the G-code status screen. If you do this immediately after power-up, it shows the initial operating conditions selected in AMP along with other control power-up default conditions.

Emergency Stop Operations

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

ATTENTION: To avoid damage to equipment or hazard to personnel, the system installer should connect the **<EMERGENCY STOP>** button, so that pressing the button opens the circuit connected to the E-STOP STATUS terminal on the 9/PC card. This should disable the axis drives and the spindle drive circuits, which should both be connected to this terminal. Refer to the integration manual or the documentation prepared by your system installer for details.

If equipped with a push-button MTB panel, the following occurs automatically after you press the **<EMERGENCY STOP>** button:

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

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

Emergency Stop Reset

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

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

To reset the emergency stop state, press the **<E-STOP RESET>** button. After the cause of the E-Stop is resolved, the control clears the "E-STOP" message. If the error condition is not cleared, the "E-STOP" message continues to flash as the control remains in E-Stop state.

If the E-Stop occurred during program execution, the control may reset the program when E-Stop reset is performed provided AMP is configured to do so. Assuming that a control reset is performed, program execution begins from the first block of the program when **<CYCLE START>** is pressed. If the current axis position prohibits this, the operator can manually jog the axes clear, or consider executing a Mid-Program Start. Refer to page 7-10. If no control reset is performed, the remainder of the program block being executed when E-Stop took place is aborted, and a **<CYCLE START>** begins program execution at the next block.

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

Access Control

Access control lets the system installer assign different functions of the control to different users by means of a password. Refer to page 2-26 for a list of the functions that may be protected on the 9/PC CNC.

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

Access control only applies to the front panel and softkey inputs. It cannot control inputs from outside the system. For instance, if you control access to the delete function, the user can't delete a file, but a file can be deleted by Mini-DNC software or ODS.

Important: If you do not want to use password protection, simply select all functions as accessible for access level 8. Since access level 8 is automatically available at power up, no password is necessary to access any of the functions of the control. Password protection can also be disabled by assigning a level at the power-up level by using the "POWER UP LEVEL" parameter as described on page 2-26.

Assigning Access Levels and Passwords

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

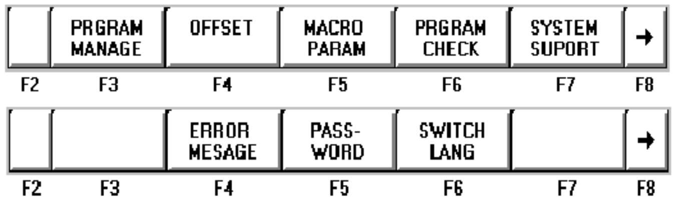
Important: Functions or passwords can be assigned to another access level only if:

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

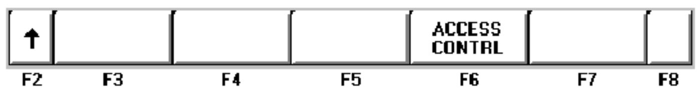
To change the functions or password of a lower user number, follow these steps:

1. Press the **{PASSWORD}** softkey.

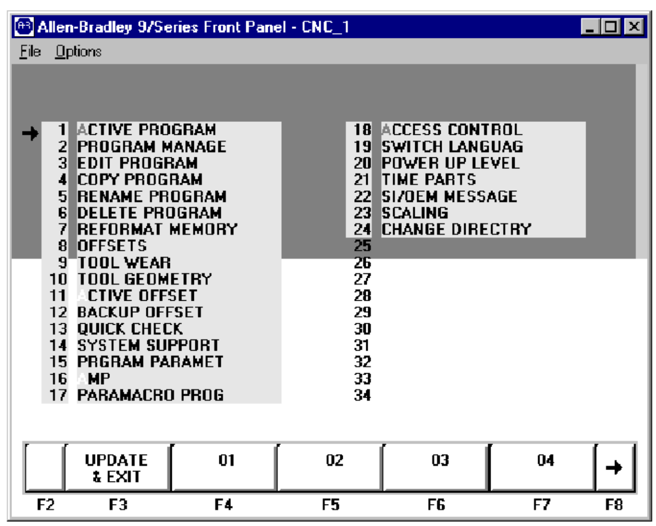
(softkey level 1)



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



This screen appears.

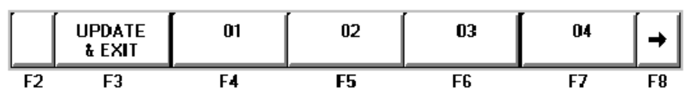


The softkey names change to display the 8 access levels along with their corresponding passwords (provided that a password has been assigned to that access level). Only the password names of access levels that are lower than the currently active access level are shown.

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

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

(softkey level 3)



4. If you want to enter or change the password for the selected level, edit the password next to the “PASSWORD NAME” prompt by using the input cursor as described on page 2-10. If you also want to change the functions for this password, move on to step 5. To save the change made to the password and leave the access control screen press the **{UPDATE & EXIT}** softkey.

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

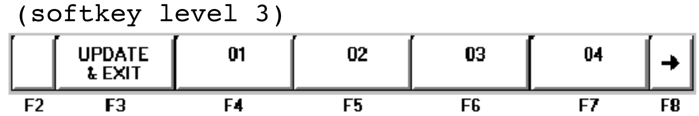
5. Use the up, down, right, and left cursor keys to select the functions to change for that access level. The selected function is shown with a flashing ⇨ to the left of the function.

6. Pressing the **[ENTER]** key toggles the function between accessible and inaccessible for that access level.

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

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)



Password-protectable Functions

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

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

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

Table 2.D
Password Protectable Functions

Parameter Name:	Function becomes accessible when parameter name is in reverse video:
1) ACTIVE PROGRAM	To access these features, both ACTIVE PROGRAM and PROGRAM MANAGE (number 2 below) must be assigned to the user. • {DE-ACT PRGRAM} — Deactivate the currently active part program. • {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. • {SEQ STOP} — Choose a sequence number for program automatic program execution to stop at. • {TIME PARTS} — Logs data relevant to part program execution (e.g., number of workpieces cut, cylec time, etc.)
2) PROGRAM MANAGE	• {ACTIVE PRGRAM} — All of the functions listed above in number 1, provided ACTIVE PROGRAM 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. • {DELETE PRGRAM} — Delete a single program stored in memory. • {VERIFY PRGRAM} — Verify that two programs are identical using the verify function. • {PRGRAM COMENT} — Add comments to a program name in the directory. • {RENAME PRGRAM} — Change a program name. • {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.
5) RENAME PROGRAM	{RENAME PRGRAM} — Rename a program name.
6) DELETE PROGRAM	{DELETE PRGRAM} — Delete a single program stored in memory
7) DELETE ALL PROGRAMS	{REFORM MEMORY} — Delete all programs currently stored in memory.

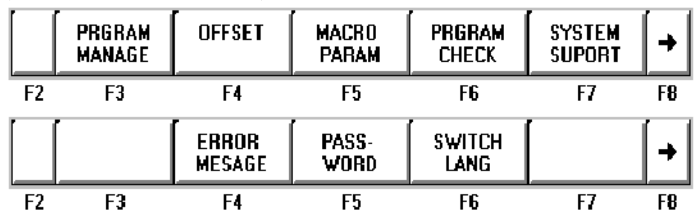
Parameter Name:	Function becomes accessible when parameter name is in reverse video:
8) OFFSETS	• {WORK CO-ORD} — Display and alter the preset work coordinate system zero locations and the fixture offset value. • {TOOL WEAR} Display and alter the tool wear amount tables for the different tools. • {TOOL GEOMET} — Display and alter the tool geometry tables. • {TOOL MANAGE} — Alter the tool life indicators and other machine specific tool functions. • {RANDOM TOOL} — Allows the use of the random tool tables used to keep track of different tools in different tool pocket (refer to chapter 19).
9) TOOL WEAR	{TOOL WEAR} — Display and alter the tool wear amount table for the different tools.
10) TOOL GEOMETRY	{TOOL GEOMET} — Display and alter the tool geometry table.
11) ACTIVE OFFSET	{ACTIVE OFFSET} — Change the currently active offset number without requiring the programming of a different offset number.
12) BACKUP OFFSET	{BACKUP OFFSET} — Make a copy of the current tool offset data.
13) QUICK CHECK	{QUICK CHECK} — Use the syntax and format checker.
14) SYSTEM SUPPORT	• {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. • {MONI-TOR} — Display the logic message editor, 1394 drives information, and the axis monitor for following error, distance to marker, etc. • {TIME PARTS} — Logs data relevant to part program execution (e.g., number of workpieces cut, cylec time, etc.) • PTOM SI/OEM} — Edit/enter system integrator messages that appear at power turn-on. • {SYSTEM TIMING} — Displays system timing values.
15) PROGRAM PARAMETERS	{PRGRAM PARAM} — Display and change the tables for programmable zones 1 and 2, and the fixed-cycle operating parameters.
16) ONLINE AMP	{AMP} — Display and change the online adjustable machine parameters.
17) PARAMACRO PARAMETERS	{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	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.
23) SCALING	When SCALING is not in reverse video, the operator still has access to the {SCALNG} softkey; however values on the screen may not be modified.
24) CHANGE DIRECTORY	Allows access to the protectable directory for file edit, direct execution selection, and allows access to the hard drive.

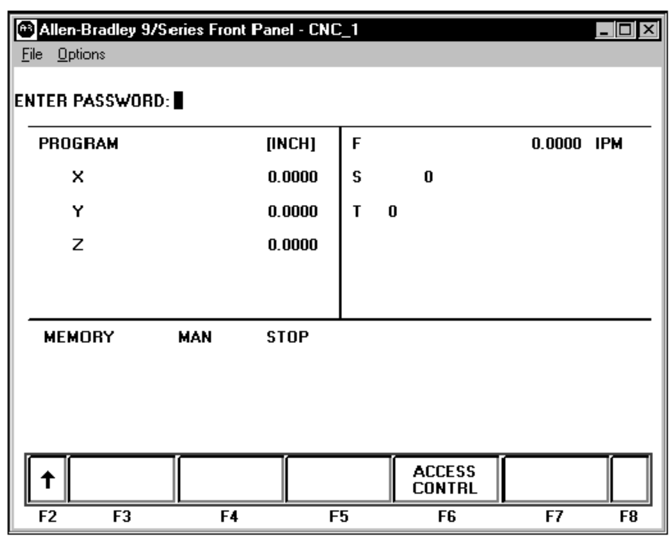
Entering Passwords

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

1. Press the **{PASSWORD}** softkey.

(softkey level 1)





2. Enter the password you want to activate by typing it in on the input line with the keys on the operator panel. The control displays * for the characters you entered. If you make an error entering the password, edit the input line as described on page 2-10.

3. When the password is correct, press the **[ENTER]** 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.

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.

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

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

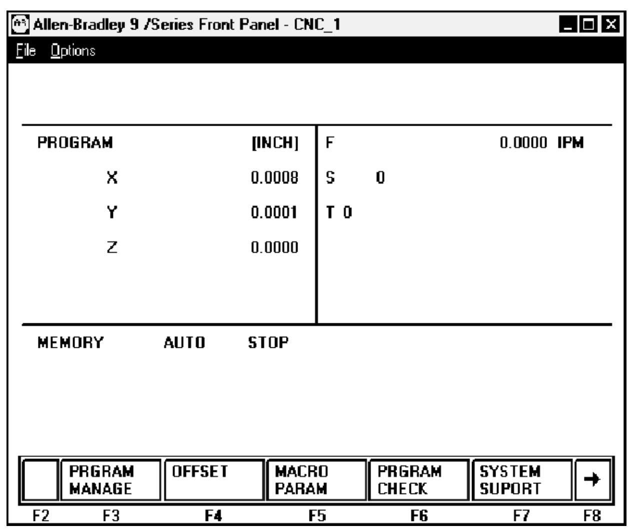
Manual Mode

To operate the machine manually,

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

For details on Manual Mode operation, see chapter 4.

Figure 2.2
Manual Mode Screen



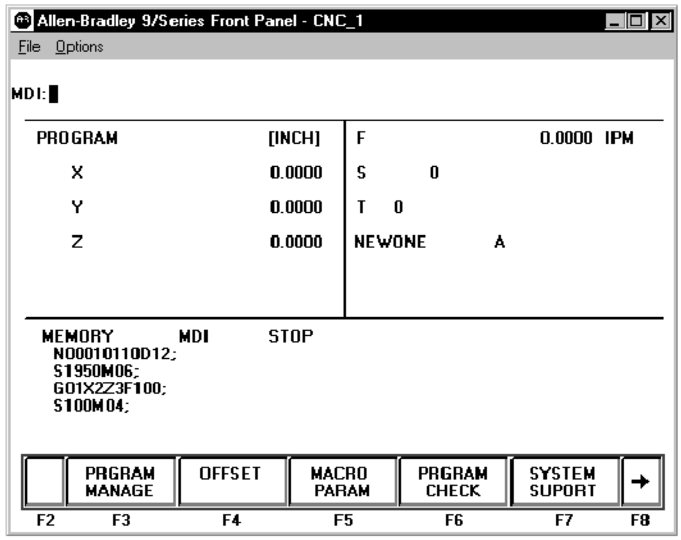
MDI Mode

To operate the machine in MDI mode,

- select MDI under **<MODE SELECT>**

For details on MDI operation, see page 4-9.

Figure 2.3
MDI Mode Screen



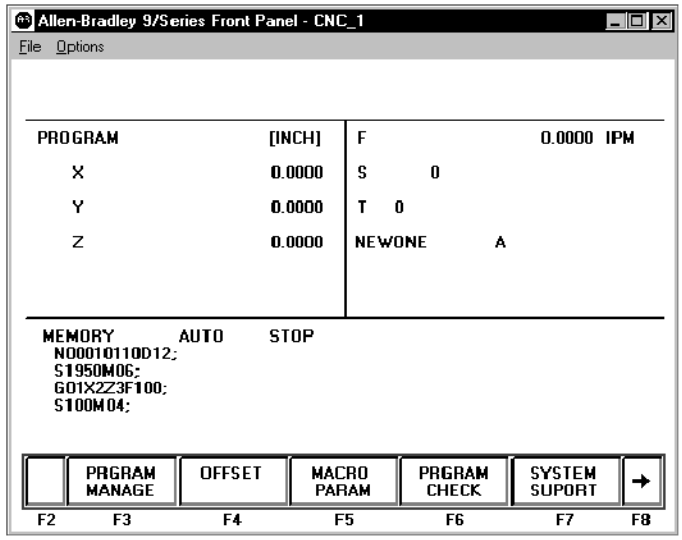
Automatic Mode

To operate the machine automatically,

- select AUTO under **`<MODE SELECT>`**

For details on automatic operation, see chapter 7.

Figure 2.4
Automatic Operation Screen



Displaying System and Machine Messages

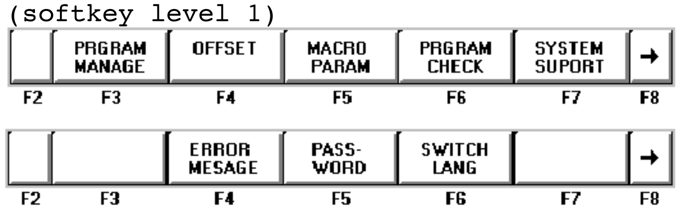
The control has a message screen dedicated to displaying messages. The screen displays up to nine of the most current system messages and nine of the most current machine messages at a time. The message screen is also able to display a log of up to 999 system error messages that occurred since the last time memory was cleared.

Important: The control automatically displays the highest priority, single, active message on all screens (other than the message screen) on line 1 of the BDS. 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 screen to display all the messages that are currently active, or a log of the most recent messages. To access the message screen, follow these steps:

1. From the main menu press the continue {⇒} softkey to change the softkey functions.
2. Press the **{ERROR MESAGE}** softkey to change the screen to the message display screen shown in Figure 2.5.

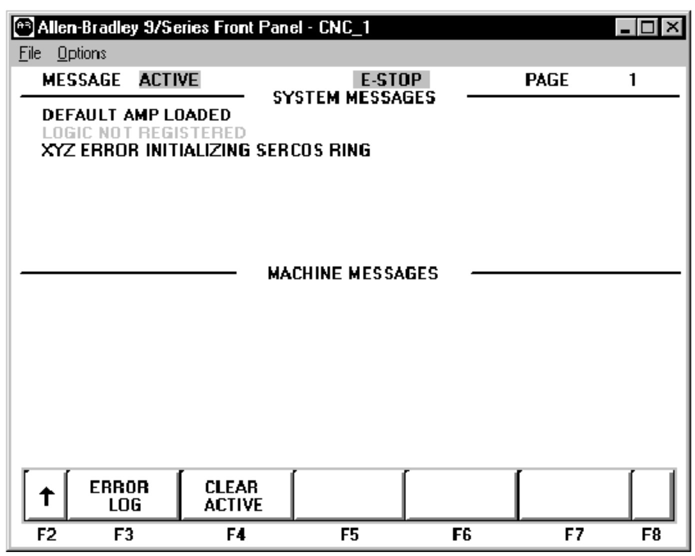
(softkey level 1)



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

- system messages in the top half of the screen
- machine messages in the bottom half of the screen

Figure 2.5
Message Display Screen

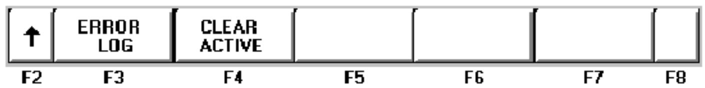


Important: For a listing of system messages and a brief description, refer to appendix B. For a description of machine messages, refer to documentation prepared by your system installer.

3. Press the **{ERROR LOG}** softkey to display the 999 most recent messages that occurred on the control since memory was last cleared. Up to nine of the most recent messages appear on the display at one time. The “MESSAGE ACTIVE” screen changes to the “MESSAGE LOG” screen.

   To display more messages, press the down cursor key while holding the **[SHIFT]** key. The next most current set of messages is displayed. The control displays up to eleven pages of the most current messages that have occurred since the last time memory was cleared.

(softkey level 2)



4. To return to softkey level 1 press the exit **{⇑}** softkey while holding the **[SHIFT]** key.

Clearing Active Messages {CLEAR ACTIVE}

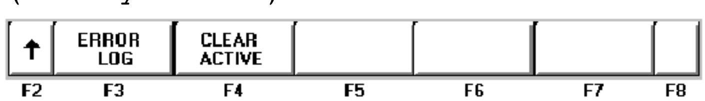
After the cause of a machine or system message has been resolved, some messages remain displayed on all screens until you clear them.

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

Active messages are cleared from the screen in this way:

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

(softkey level 2)



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

{REFORM MEMORY}

You may want to perform a Reform Memory operation to delete all part programs stored in your 9/PC's memory.

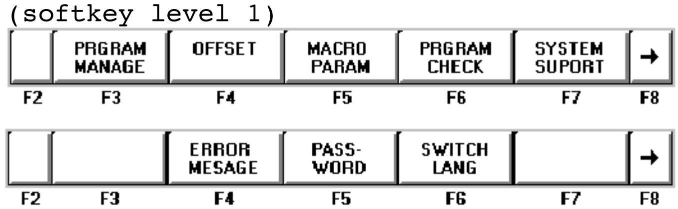
Important: The **{REFORM MEMORY}** softkey will not clear the memory of your hard drive.

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

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

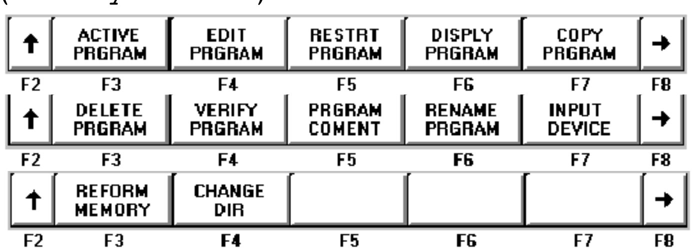
1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)



2. Press the **{REFORM MEMORY}** softkey.

(softkey level 2)



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

(softkey level 3)

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

Removing an Axis (Axis Detach)

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

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

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

Time Parts Count Display Feature

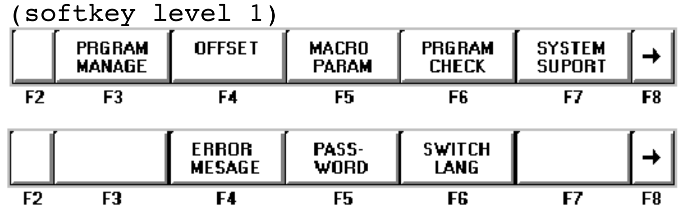
There are three levels of access to the Time Parts screen. They are listed below in order of most restrictive to least restrictive. Refer to page 2-23 for details on password protection and access control.

Access:	Protection:
No	Restricts operator from Time Parts screen entirely (softkey {TIME PARTS} not accessible). Accomplished by denying access to "Active Program."
Operator	Restricts operator from setting "Power-on time/overall" and "Workpieces cut/overall." Accomplished by denying access to "Time Parts."
Supervisor	Full access to all features of the Time Parts screen.

To access the Time Parts screen, follow these steps:

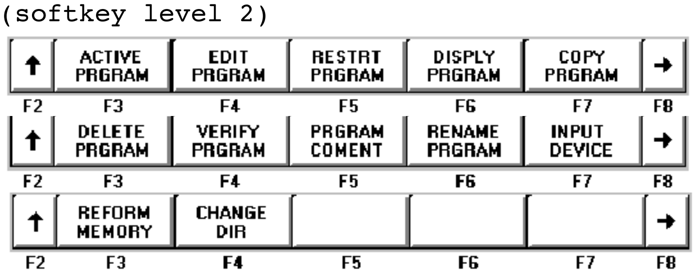
1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)



2. Press the **{ACTIVE PRGRAM}** softkey.

(softkey level 2)



3. Press the **{TIME PARTS}** softkey. This generates the screen shown in Figure 2.6.

(softkey level 3)

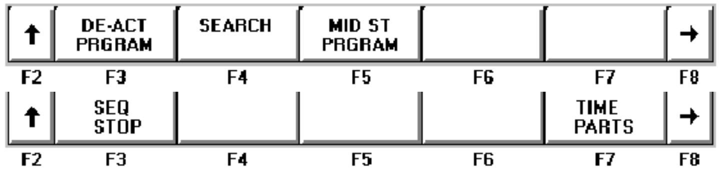
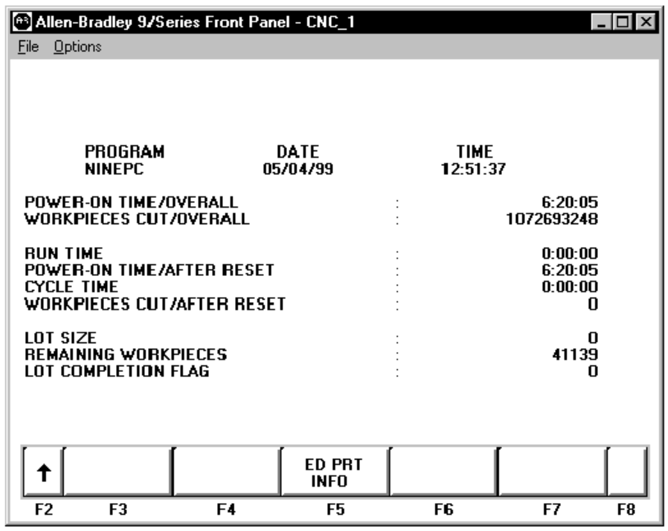


Figure 2.6
Time Parts Screen



Important: The date and time information is based on the date and time set on your PC and is read by the 9/PC when you start the CNC. Changing the date or time on your PC while 9/PC is running will not change the time and date displayed on the Time Parts screen. To resynchronize the time and date on your 9/PC with the time and date on your PC, you must stop and restart your 9/PC.

You can modify the values on this screen. Press the **{ED PRT INFO}** softkey as explained in the Screen Field Definitions that follow.

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

Time Part Screen Field Definitions

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

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

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

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

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

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

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

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

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

- press the cycle-start button to initiate program execution
- turn off the control power
- follow these steps:

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

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

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

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

Remaining Workpieces – indicates the number of workpieces that still need to be cut in the lot. The value for this field is automatically set equal to the lot size each time the "Lot Size" value is changed. When the control encounters an M02, M30, or M99 in a main part program, the remaining workpieces field is decremented by one. The control tells the system installer's logic 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 logic 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 logic dependant. See the documentation supplied by your system installer.

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

Part Program Storage

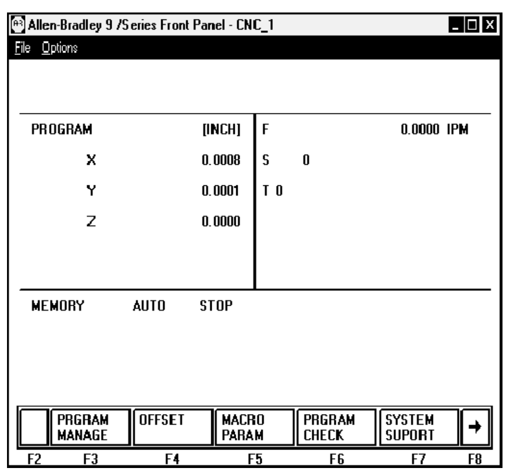
There are three directories available for storage and execution of part programs: main and protectable, which allow you to run programs resident on the CNC, and a directory on the PC’s hard drive, as configured by the File Handler.

You can execute programs from the single directory on the hard drive. Refer to the Installation manual for more information about configuring your hard drive and storing part programs on your main and protectable drives. Refer to your 9/PC Integration Manual for information about locating your File Handler directory.

Choosing a Part Program Directory

When you power up your 9/PC system, it automatically returns you to the last active part program directory. To select a new part program directory:

1. From the BDS main menu, select **`{PRGRAM MANAGE}`**. The program directory screen appears and displays the contents of the last active part program directory.



2. To access the **{CHANGE DIR}** softkey layer, press **[F8]** twice. Select the **{CHANGE DIR}** softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
F2	F3	F4	F5	F6	F7	F8
↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→
F2	F3	F4	F5	F6	F7	F8
↑	REFORM MEMORY	CHANGE DIR				→
F2	F3	F4	F5	F6	F7	F8

The **{CHANGE DIR}** menu item allows you to scroll through the main, protectable, and hard drive directories. The part programs in the main and protectable directories reside in the control's memory. The programs in the hard drive reside on the personal computer.

Copying Part Programs Between Directories

To copy your part program between directories:

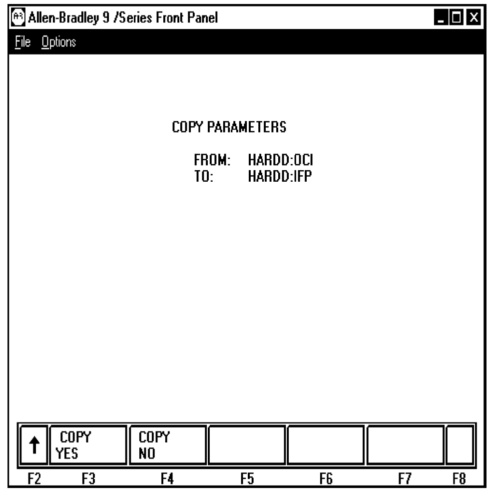
1. Select the **{PRGRAM MANAGE}** softkey.
2. If necessary, access the **{CHANGE DIR}** softkey to toggle to your desired directory. Access the **{CHANGE DIR}** softkey by pressing **[F8]** twice.
3. After choosing the desired directory, highlight the part program that you want copied.
4. After the selected program name, which appears on the Selected Program: line, type a comma, followed by the new program name. Select **[F8]** again to return to the **{COPY PRGRAM}** layer of softkeys.
5. Choose the source you want your program copied to.

↑					MEM TO MEM	
F2	F3	F4	F5	F6	F7	F8

6. Select the destination you want your part program copied to:

↑	TO MAIN	TO PROTEC	TO HARDD			
F2	F3	F4	F5	F6	F7	F8

This screen appears:



7. Select the {**COPY YES**} softkey to copy the part program OR select the {**COPY NO**} softkey to abort the process.
8. If desired, select {**VERIFY PRGRAM**} to verify that the copied part program is identical to the original.

Part Program Sizes

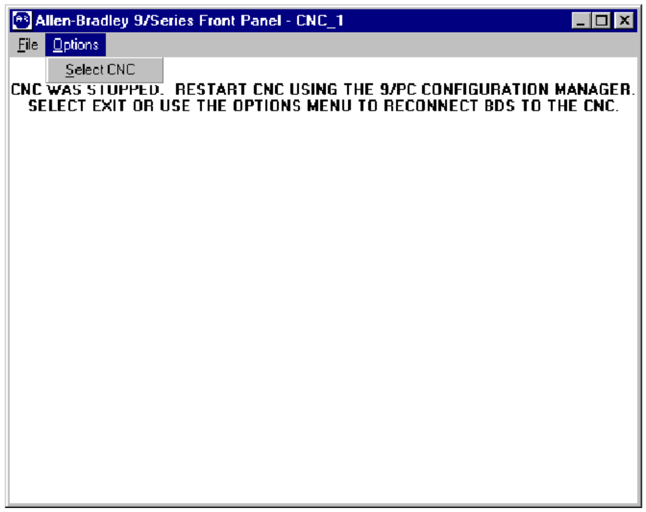
When moving part programs from one device to your 9/PC, you will notice that the length of your program may change. Since various sources have different means of recording keystrokes, your file size may increase or decrease in size, without losing or gaining any information.

For example, the control may record a return as one keystroke (end-of-block character), whereas the PC records it as two (carriage return + line feed character). In this event, your PC file may have originally been 452.4 meters, but after it is downloaded to the control, it may read as 449.8 meters.

Cycling Power

Whenever you are required to cycle power to your 9/PC (i.e., after downloading AMP), you must restart BDS. To restart BDS:

1. Select the “Select CNC” option under the Options menu.



2. Highlight the applicable CNC.
3. Click **[OK]** to restore the connection to that CNC.

END OF CHAPTER

Offset Tables and Setup

Chapter Overview

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

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

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

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

This data can be entered into the offset tables:

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

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

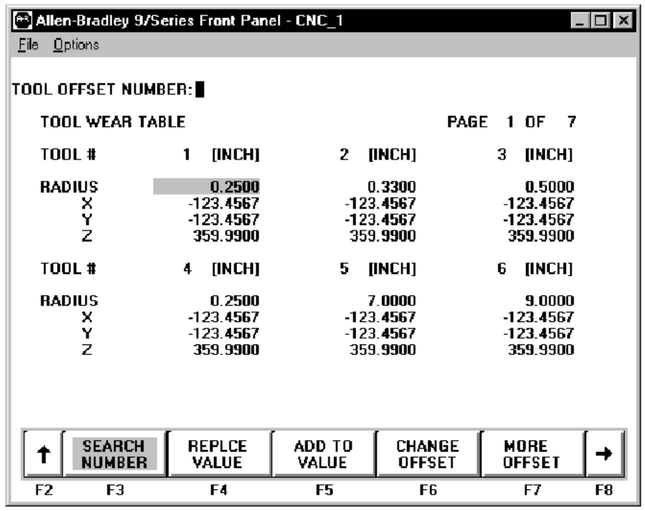
- AMP reference manual
- ODS User's Manual, publication MCD-5.1

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

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

For details on using tool offsets see chapter 19. For information on cutter compensation refer to chapter 20.

Figure 3.1
Offset Table Screen for Wear



Tool Offset Dimensional Parameters

Tool Offset Numbers (Geometry and Wear Table)

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

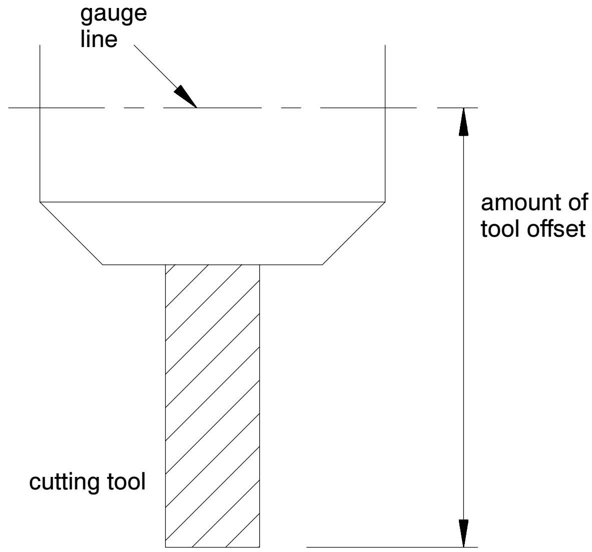
For more on calling offset numbers refer to chapter 19.

Tool Length Offset Data (Geometry Table)

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

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

Figure 3.2
Tool Length Offset



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

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

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

Tool Diameter Compensation Data (Geometry Table)

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

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

The control can compensate for this difference using the cutter compensation feature discussed in chapter 20.

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

Tool Length Wear Data (Wear Table)

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

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

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

Tool Diameter Wear Compensation Data (Wear Table)

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

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

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

Setting Tool Offset Tables

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

- Using **{MEASURE}** (page 3-8)
- Programming G10s (chapter 19)
- Skip functions using a probe (chapter 26)
- Setting Paramacro System Parameters (chapter 27)
- Altering through the logic program

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

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

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

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

1. Press the **{OFFSET}** softkey.

(softkey level 1)

F2	F3	F4	F5	F6	F7	F8
	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
		ERROR MESAGE	PASS-WORD	SWITCH LANG		→

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

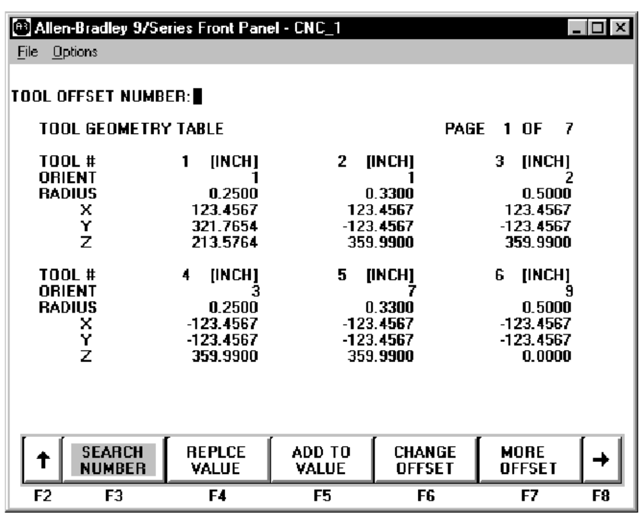
   To display the **geometry offsets** (tool length offsets and the tool diameter data), press the **{TOOL GEOMET}** softkey. An example of a tool offset geometry screen is shown in Figure 3.3.

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

(softkey level 2)

F2	F3	F4	F5	F6	F7	F8
↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
↑		BACKUP OFFSET	SCALNG			→

Figure 3.3
Tool Offset (Geometry) Screen



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

4. Select the units using **{INCH/METRIC}**

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

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

(softkey level 3)

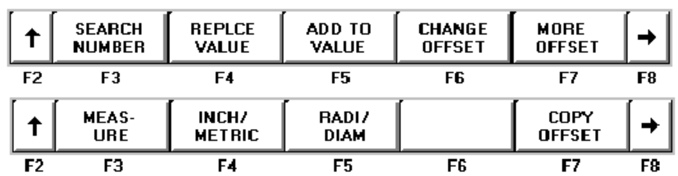
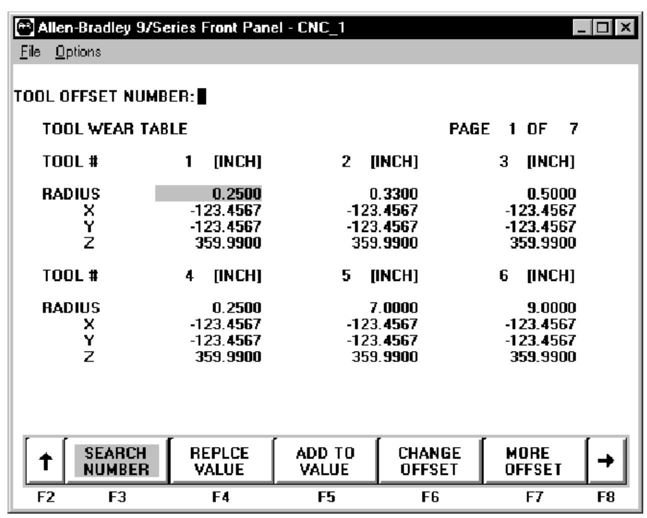


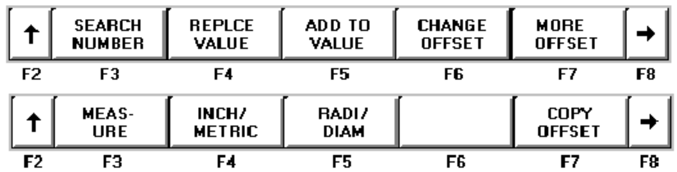
Figure 3.4
Tool Offset (TOOL WEAR) Screen



5. Replace or add data as follows:

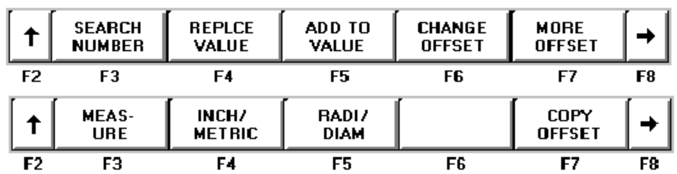
- To replace stored offset data with new data, key-in the new data, and press the **{REPLCE VALUE}** softkey.
- To add to previously stored offset data, key-in the amount to be added, and press the **{ADD TO VALUE}** softkey.

(softkey level 3)



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

(softkey level 3)



Setting Offset Data Using {MEASURE}

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

1. Establish a fixed machine position without a tool in the chuck. This position may be any fixed, non-movable location on the machine that the tool may be jogged against consistently, using a variety of different tools.

- If entering a tool length in the **geometry offset table**, jog the machine gauge line (on the axis being updated) to this position. No tool offsets should be active and no tool should be in the chuck. The value of this position, located in the work coordinate system, must be recorded. The user keys in this value in steps 6 and 7.

- If entering a tool length wear in the **wear offset table**, jog the machine gauge line to the fixed position. No tool offsets should be active, and no tool should be in the chuck. The value of this position, located in the work coordinate system, must be recorded. Add the original tool length offset from the tool geometry table to the fixed machine location. The user keys in this value in step 7.

2. Access the tool geometry or wear offset table as discussed on page 3-5.
3. Cursor down to the offset that is to be changed. Note that the offset can be displayed in either inch or metric measurements.
4. Load the tool that is to be measured into the chuck.
5. Using incremental, continuous or handwheel mode, jog the tool tip to the fixed location determined in step 1.
6. Press the **{MEASURE}** softkey.
7. Key in the coordinate value of the fixed location determined in step 1.
8. Press the **[ENTER]** key.

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

Tool Offset Range Verification

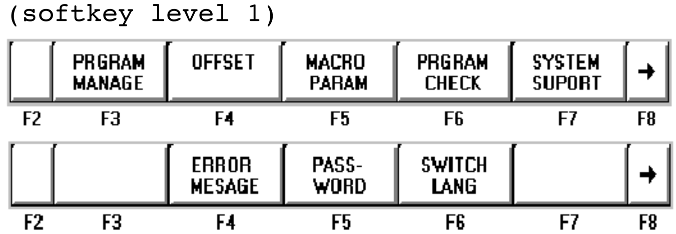
Tool offset range verification checks:

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

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

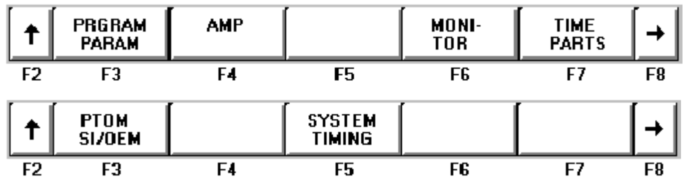
9. Press the **{SYSTEM SUPORT}** softkey.

(softkey level 1)



10. Press the **{AMP}** softkey.

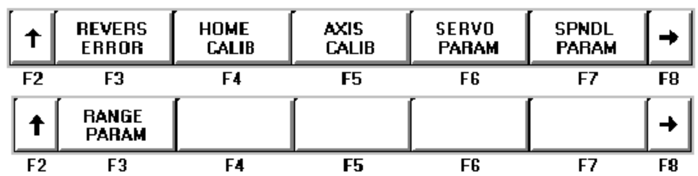
(softkey level 2)



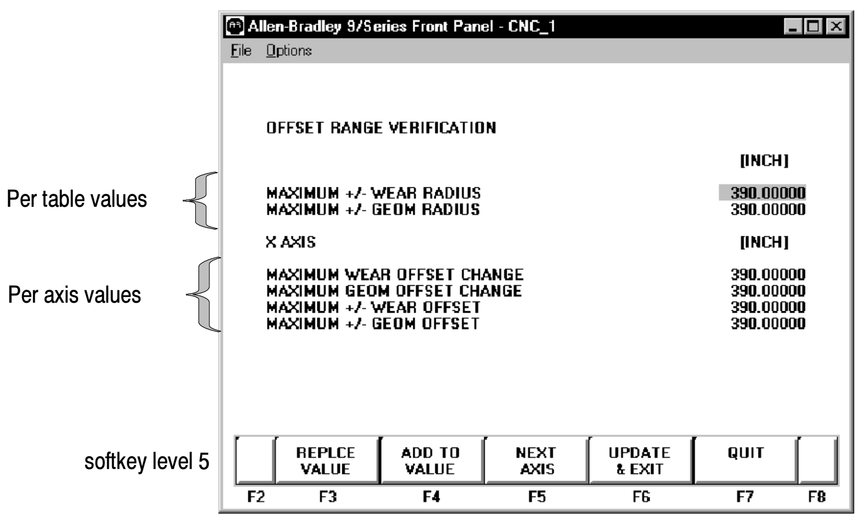
11. Press the **{AXIS PARAM}** softkey.

12. Press the **{RANGE PARAM}** softkey.

(softkey level 4)



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



About the Offset Range Verification Screen

- display format is fixed

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

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

When Does Verification Occur

Verification occurs when a value enters the table from:

- data entry screens
- logic
- 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."

Changing the Active Tool Offset {ACTIVE OFFSET}

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

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

The axis that is selected as the length axis (the axis that length offsets are applied to) is shown in reverse video. The length axis is selected in AMP or through programming as discussed on page 19-8.

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

1. Press the **{OFFSET}** softkey.

```
(softkey level 1)
```

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
F2	F3	F4	F5	F6	F7	F8

		ERROR MESAGE	PASS-WORD	SWITCH LANG		→
F2	F3	F4	F5	F6	F7	F8

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

```
(softkey level 2)
```

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
F2	F3	F4	F5	F6	F7	F8

↑		BACKUP OFFSET	SCALNG			→
F2	F3	F4	F5	F6	F7	F8

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

3. Move the cursor on the offset table until the desired offset is shown in reverse video. If it is desired to activate a length offset (normally programmed with an H-word), make sure the selected offset value is in the "LENGTH" column. If it is desired to activate a diameter or radius offset (normally programmed with a D-word), make sure the selected offset value is in the "DIAMETER" column. Only one length offset and one diameter offset may be active at the same time.

4. Press the **{ACTIVE OFFSET}** softkey when the desired offset is selected. The offset will be made active provided that that offset mode is active on the control G41 or G42 for tool diameter offsets and G43 or G44 for tool length offsets. Refer to chapters 19 and 20 for details on programming these tool offsets.

```
(softkey level 3)
```

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	CHANGE OFFSET	MORE OFFSET	→
F2	F3	F4	F5	F6	F7	F8

↑	MEAS-URE	INCH/ METRIC	RADI/ DIAM		COPY OFFSET	→
F2	F3	F4	F5	F6	F7	F8

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

If cutter compensation is active (G41 or G42) and the control is not currently in MDI mode, then the new radius is activated as discussed in chapter 20. If G40 is currently active, then only the D-word is activated until a G41 or G42 is programmed.

Work Coordinate System Offset Tables {WORK CO_ORD}

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

Zero Point Parameters

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

Enter a machine coordinate system position for each of the work coordinate systems. Refer to chapter 11 for more information about machine coordinate systems.

External Offset

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

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

Setting Work Coordinate System Tables

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

- Programming G10s (chapter 10)
- Setting paramacro system parameters (chapter 27)
- Modify offsets through PAL (see the system installer's documentation)

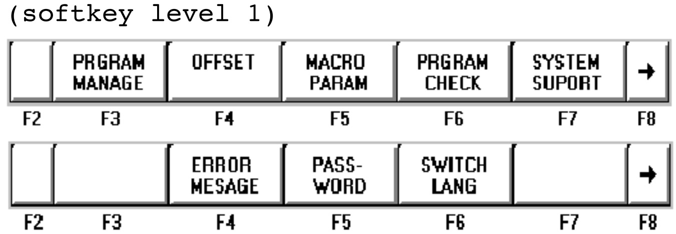
The fourth method, and the one discussed in this section, lets you modify the work coordinate values immediately by using the keyboard.

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

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

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

(softkey level 1)



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

(softkey level 2)

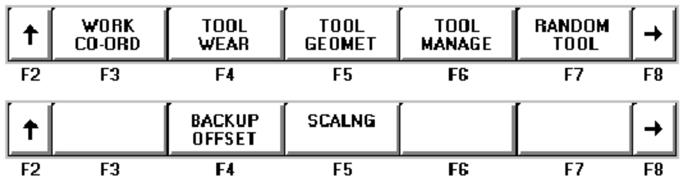
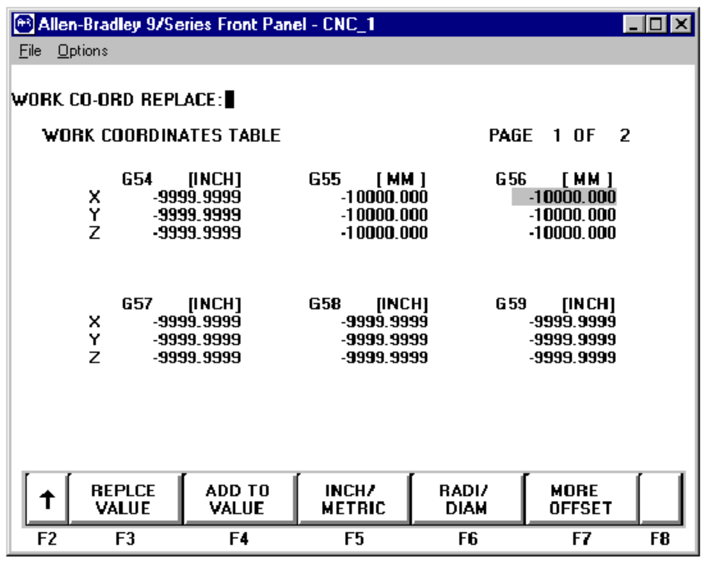


Figure 3.5
Work Coordinate System Setting



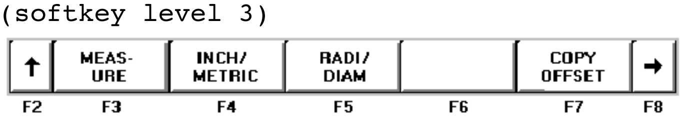
3. Move the cursor to the offset data to be modified. Use the up, down, left, or right cursor keys to move the block cursor to the offset data on the current page. Press the **{MORE OFFSET}** softkey to change pages. The selected item will be shown in reversed video.

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

4. Units selection **{INCH/METRIC}**

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

(softkey level 3)



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

Data can be replaced or added to as follows:

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

(softkey level 3)

5. Replace or add data.

Backing Up Offset Tables

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

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

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

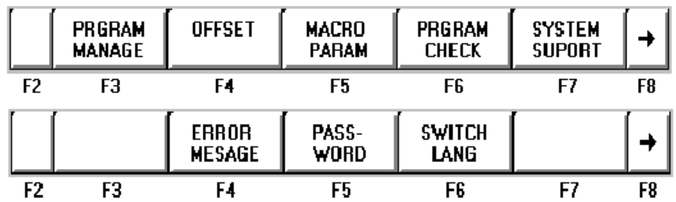
The offset table program can be saved in either control memory as a program. This is very useful if the same tools are to be used on different machines. The offset tables can be easily set up by executing the same offset table program on all the other machines.

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

To backup the offset tables follow the directions below:

1. Press the **{OFFSET}** softkey.

(softkey level 1)



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

(softkey level 2)

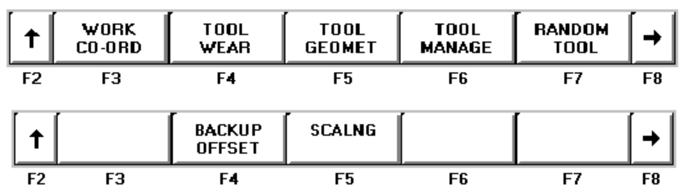
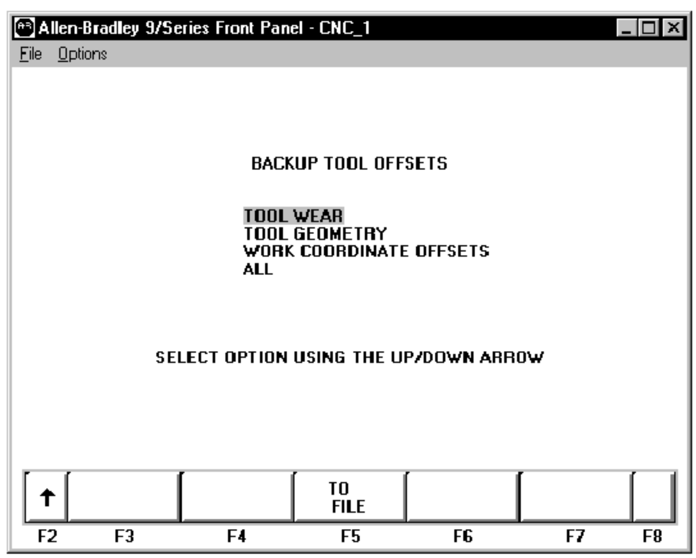


Figure 3.6
Backup Offset Screen



3. Select the offsets to be backed up by moving the cursor to the desired offset using the up and down cursor keys. The selected offset will be shown in reverse video. There are four options here:

   - TOOL WEAR -- When wear is selected all data from the tool offset wear tables is stored as a G10 program.

- TOOL GEOMETRY -- When geometry is selected all data from the tool offset geometry tables is stored as a G10 program.
- WORK COORDINATE -- When work coordinate systems is selected the work coordinate offset information for the G codes G54 - G59.3 and the external offset value are stored as a G10 program.
- ALL -- When all is selected all data from the tool offset geometry and wear tables and work coordinate offset tables is stored as a G10 program.

4. Once the data to save has been selected, press the **{TO FILE}** softkey to send the G10 program to control memory. The control asks for a program name under which to store the program. Enter the program name by using the alphanumeric keys on the operator panel and press the **[ENTER]** key. Refer to chapter 9 on program names. The G10 program is saved under the file name just entered.

Programmable Zone Table

The programmable zone feature provides a means to prevent tool motion from entering or exiting a designated area. For details on programmable zones refer to chapter 11.

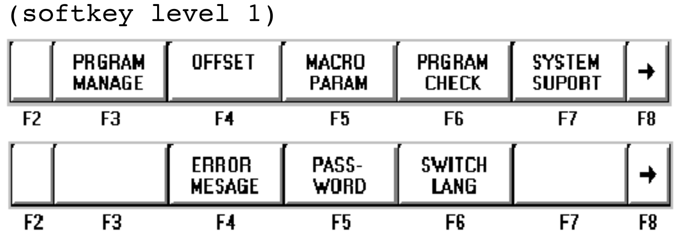
This table contains the values for programmable zones 2 and 3. These values define the boundaries for the programmable zones and are referenced from the machine coordinate system.

Important: These values may also be entered in AMP by the system installer. Programmable zone 3 table values may also be modified by programming a G22 command (refer to chapter 11).

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

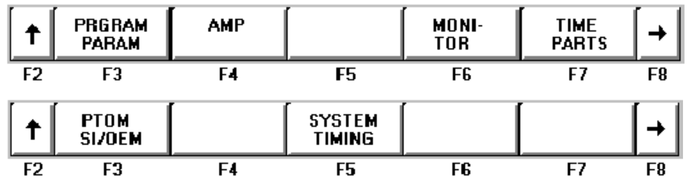
1. Press the **{SYSTEM SUPORT}** softkey.

(softkey level 1)



2. Press the **{PROGRAM PARAM}** softkey.

(softkey level 2)



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

(softkey level 3)

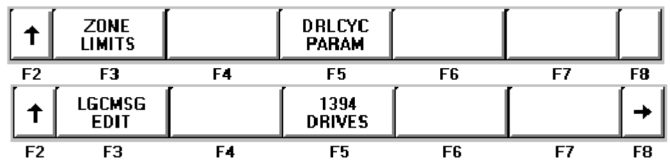
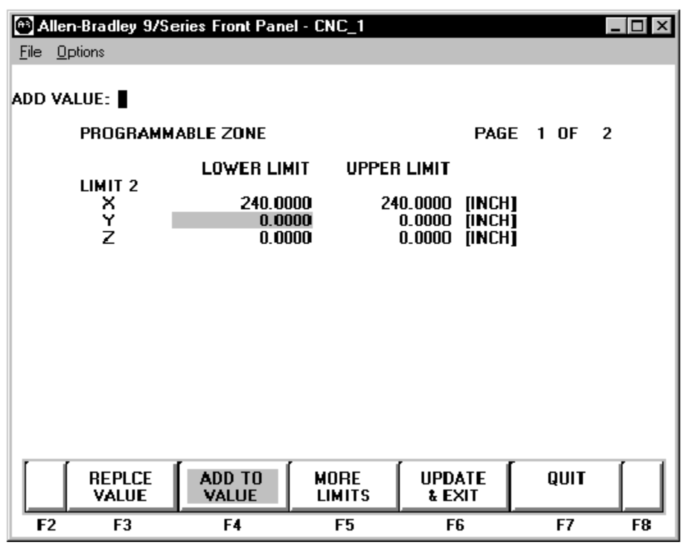


Figure 3.7
Programmable Zone Table



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

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

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

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

(softkey level 4)

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

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

(softkey level 4)

END OF CHAPTER

Manual/MDI Operation Modes

Chapter Overview

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

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

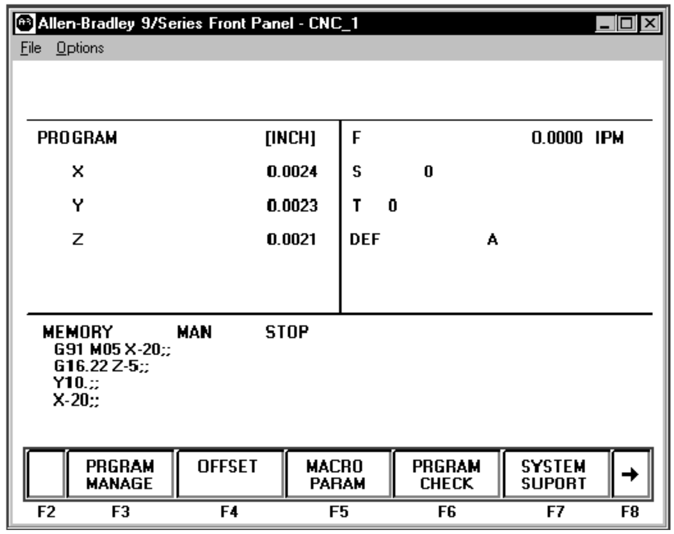
Manual Operating Mode

To go into the manual mode, select MANUAL under **<MODE SELECT>**.

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

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

Figure 4.1
Data Display in MANUAL Mode



Jogging an Axis

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

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

- continuous jog -- the axes move continuously while a pushbutton on the MTB panel is held.
- incremental jog -- the axes move a predetermined amount each time a pushbutton on the MTB panel is pressed.

Normally, the axes can only be jogged in manual mode. Your system installer can write logic to allow jogging in the automatic and MDI modes. Refer to page 4-4 for more information about jog offsets.

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

Only normal single-axis jogs (one axis at a time in the continuous, or incremental modes) are permitted during a jog retract operation. Refer to page 7-14 for details about basic program execution.

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

Continuous Jog

To continuously jog an axis:

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

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

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

Incremental Jog

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

1. Select INCREMENTAL under **<JOG SELECT>**.
2. Select the jog increment under **<SPEED/MULTIPLY>**. The jog increment is equal to an amount specified in AMP for each selection under **<SPEED/MULTIPLY>**.
3. Press the **<AXIS/DIRECTION>** button for the axis and direction to jog. The control makes one incremental move each time the **<AXIS/DIRECTION>** button is recognized. Until the control completes the execution of the incremental move, no other jog moves are recognized on that axis. This includes attempts to perform other incremental moves on that axis.

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

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

Jog Offset

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

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

To use this feature, follow these directions:

1. Turn on the switch to activate the jog offset function. Refer to documentation provided by the system installer.
2. Change to manual mode unless the control is equipped for the “Jog on the Fly” feature which allows jogging in MDI and Automatic modes. If equipped with “Jog on the Fly,” turn on the switch to activate it. For details, refer to documentation prepared by the system installer.
3. Jog the axis by using any of the available jog types, with the exception of homing, as described on page 4-6. The control adds the amount of the jog move as offsets to each jogged axis immediately when the jog takes place.

Important: When the jog move is made, the axis position displays do not change on the screen unless the currently active screen is the absolute screen as described on page 8-4. This is because the value is being added to the work coordinate system offset and the control does not recognize any tool motion on the coordinate system.

Resetting Overtravels

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

- **Hardware Overtravel --** the axes reach a travel limit, usually set by a limit switch or sensor mounted on the axis. Hardware overtravels are always active.
- **Software Overtravel --** commands cause the cutting tool to pass a software travel limit. Software overtravels are active only after the axis has been homed provided the feature has been activated in AMP by the system installer.
- **Programmable Zone Overtravel --** the axes reach a travel limit established by independent programmable areas. Programmable Zones are activated through programming the appropriate G-code.

These 3 causes of overtravel are described in detail in chapter 11.

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

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

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

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

To reset a software or programmable zone overtravel condition:

1. Determine whether the control is in E-Stop. If it is not, go to step 4.
2. Look for and eliminate any other possible conditions that may have caused emergency stop, then make sure that it is safe to reset the emergency stop condition.

3. Press the **<E-STOP RESET>** button to reset the emergency stop condition. If the E-Stop does not reset, it is a result of some cause other than overtravel causing E-Stop.
4. Make sure it is safe to move the axis away from the overtravel limit.
5. Use any of the jog features described on page 4-1, except homing and jog offset, to manually move the axis away from the limit. Any attempt to jog the axis in the direction of the overtravel will not be allowed.

Mechanical Handle Feed (Servo Off)

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

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

Removing an Axis (Axis Detach)

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

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

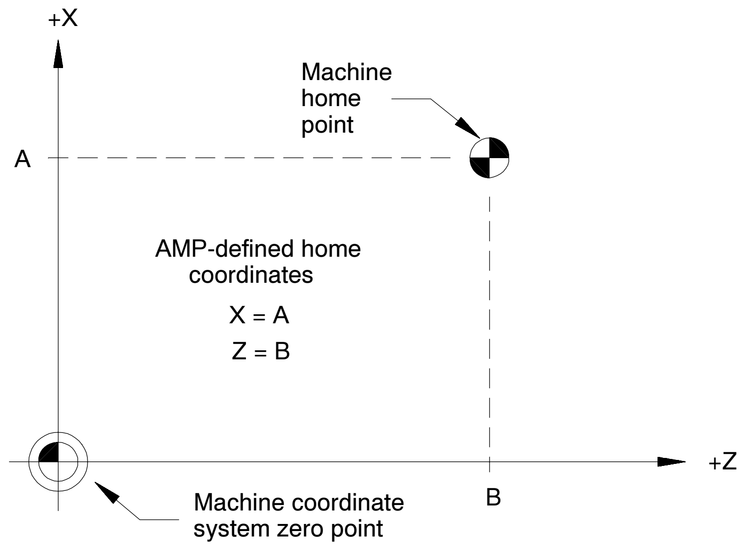
This feature can only be enabled in AMP. The axis must be selected as "Detached" to be considered removed. Refer to your system installers documentation for the necessary steps involved in actually physically removing axis hardware from a specific machine.

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 encoder marker.

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

Figure 4.2
Machine Home

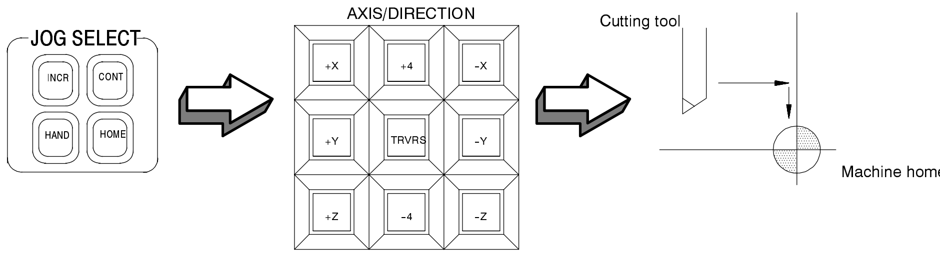


The following procedure describes how the control is homed manually by using the pushbuttons on the MTB panel. Manual homing may be different for some machines depending on the logic program written by your system installer.

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

Important: Automatic homing is available. Refer to page 13-28.

Figure 4.3
Manual Machine Home



To execute the manual return to machine home position:

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

Important: If you choose the wrong direction for an axis, it will continue to travel in the selected direction until it contacts a hard limit and an overtravel will occur. Refer to chapter 11. Your system installer has the option to enable some button or switch (typically Cycle Stop) through the logic program to abort a jog operation or prevent the user from homing the axis in the wrong direction. Refer to your system installer's documentation for details.

The axis homes when :

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

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

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

MDI Mode

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

To begin MDI operations, select MDI under **<MODE SELECT>**.

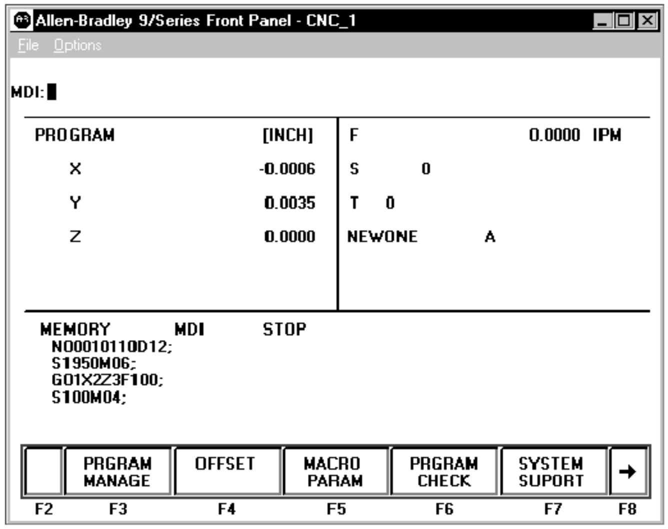
Important: If desired, your system installer has the option of disabling G- or M-code AMP-defined paramacro calls in MDI mode. For details on paramacros, refer to chapter 27.

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

ATTENTION: When program blocks are executed in MDI, no tool tip radius compensation (TTRC) is allowed. If TTRC was previously active before the MDI blocks are executed, it is temporarily canceled for the execution of the MDI blocks. Refer to chapter 20 for details on the effect of MDI on TTRC. Any TTRC G-codes that are programmed in MDI mode affect the cutter compensation mode (G41, G42, or G40) when compensation is reactivated.

Important: It is possible to call subprograms or paramacros within an MDI program, however, there are limitations to the allowable commands. Refer to chapter 27 on paramacros for details on illegal MDI commands for these features.

Figure 4.4
Program Display Screen in MDI Mode



MDI Basic Operation

Operating procedures in the MDI mode include:

1. When it is in MDI mode, the control accepts standard programming blocks.
2. Key in programming blocks (refer chapter 9). Each block, up to a maximum of 62 characters, is separated with an end of block statement. The blocks entered appear in the input area of the screen (line 2). The complete MDI program should be entered on these lines since once you send the blocks to control memory, they cannot be edited or added to.

   The input cursor is the cursor shown on the input lines (line 2 on the screen). To edit information in the input area, use the **[BACKSPACE]** key to delete everything to the left of the cursor.

   If you make a mistake keying in a character before it is sent, that character can be edited by using the input cursor described on page 2-10.
3. Pressing the **[ENTER]** key transmits the blocks to control memory. Once the blocks have been sent to control memory, you cannot send any more MDI blocks until all of the previous set has been executed.

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

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

4. The MDI blocks can then be executed continuously by pressing the **<CYCLE START>** button in either the AUTO or MDI mode. The single block, jog retract, and block retrace features are also available for MDI programs (refer to pages 7-3, 7-27, and 7-30 respectively for details on these features).

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

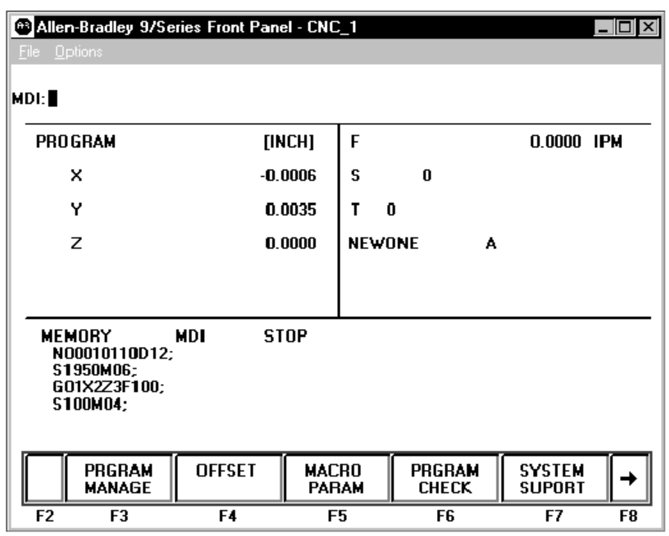
   The error message:

   "NO MORE MDI BLOCKS"

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

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

Figure 4.5
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 page 2-11 for details. By performing a control reset operation as described, the control erases all MDI blocks that have not been executed in the MDI program.

END OF CHAPTER

Editing Programs Online

Chapter Overview

You can create and edit part programs online with either the part program editor or any ASCII editor that runs on your PC. Since all programs stored on the hard drive must have a PPG extension, the OCI editor automatically saves your programs with a PPG extension. In this manual, we describe how to use the editor supplied with the OCI software.

You can also edit programs offline with your personal computer. For more information about offline operations (e.g., uploading, copying, and some file management operations), refer to chapter 6.

Important: If you intend to execute part programs from the hard drive, save your part program to the default OCI directory defined for the OCI file handler by your system integrator. Refer to your system integration documentation for instructions.

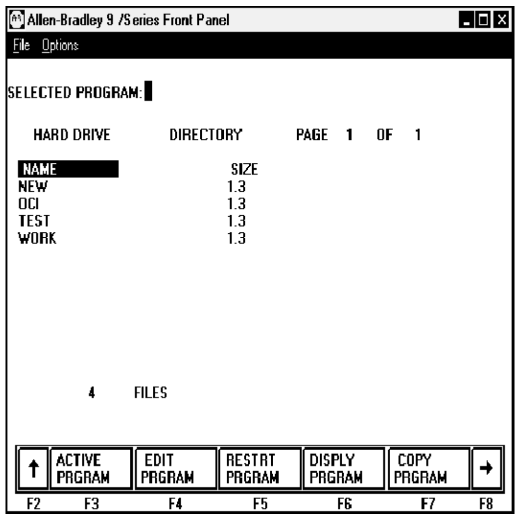
The cycle editor is a subset of the line editor, and can only be activated when the line editor is active.

The cycle editor allows you to program canned cycles using a graphical representation of the cycle along with prompts for programming data. These screens prompt you for the basic syntax of the available canned cycles.

Creating a Part Program

Creating part programs to edit using OCI is the same as using the standard front panel. To create a new part program:

1. From the BDS main menu, select the **{PRGRAM MANAGE}** softkey. The program directory screen appears and displays the contents of the last active part program directory.
2. Use the up arrow to highlight the **NAME** line.



3. At the **SELECTED PROGRAM:** line, type in the program name.
4. Select the **{EDIT PRGRAM}** softkey to begin editing the program.

Subprograms and Paramacros

When creating a subprogram or paramacro, the program must match the 9/Series format of O12345. Programs with five or fewer numeric characters that are either created with the BDS part program editor or stored in the main and protectable directories are automatically expanded to the O12345 format. However, if you used another application to create the part program on the PC (including ODS), the numbered program name is not expanded. The following illustrates the format of numbered program names in the G65 block on the available drives:

On the Main and Protectable drives:

G65P1 – will use O00001

On the Hard drive:

G65P1 – will use O1.ppg

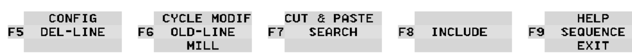
G65P00001 – will use only O00001.ppg. It will not use O1.ppg or 1.ppg.

If a subprogram resides on the hard directory, you must expand it in order for the control to recognize it. However, whenever you use the control to copy the program to the main directory, it will automatically be expanded.

Important: If a new program name is entered with five or fewer 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 existing programs with more than five numeric characters to be subprograms.

Using Layered Softkeys

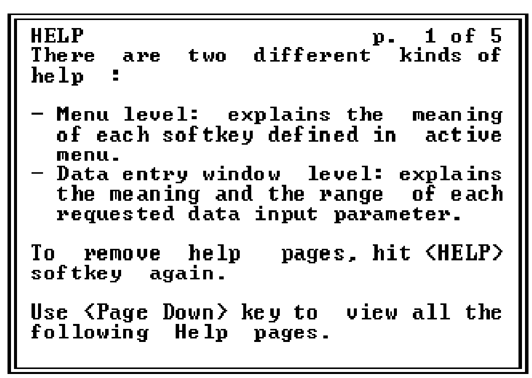
When you are in the line and cycle editors, layered softkeys appear at the bottom of the screens. The active layer is highlighted. To toggle through the main level of layered softkeys, use **[F10]**. When you reach your desired layer of softkeys, select the appropriate function key. The following is an example of layered softkeys:



Using Online Help

The OCI editor contains an online help function for reference information. Online help is accessible at any level of the editor, and while other processes (e.g., sequence and search) are active.

There are two types of help: Menu level and Data Entry Window level. Menu level help defines each softkey. Data Entry Window level help defines each data entry input parameter. Below is the first screen of online help at the main softkey level:



```
HELP                         p.  1 of 5
There  are  two  different  kinds of
help  :

- Menu level:  explains the  meaning
  of each softkey defined in  active
  menu.
- Data entry window  level: explains
  the meaning and the range  of each
  requested data input parameter.

To  remove  help   pages, hit <HELP>
softkey  again.

Use <Page Down> key to  view all the
following  Help  pages.
```

To use online help from any level of the editor, select the **{HELP}** softkey. To exit online help, select the **{HELP}** softkey again.

Selecting the Program To Edit

This section provides information on how to select a part program for editing. You can only edit part programs on line that you have stored in control memory. If a part program is on tape or another storage device and you must edit it on line, copy this program to memory as described in chapter 9.

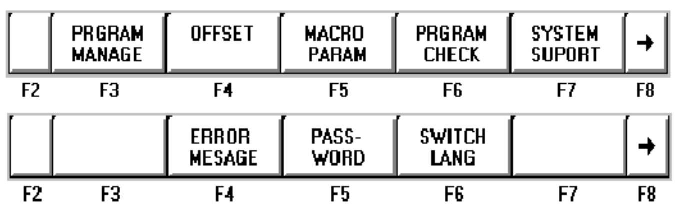
Important: You can edit programs that are selected as active for execution. Edit operations being performed on an active program must be exited before that program can actually be executed in automatic mode.

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

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

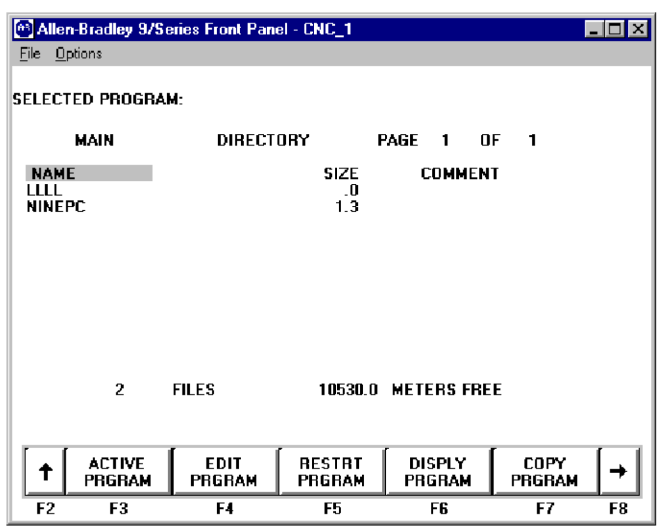
1. Press the **{PRGRAM MANAGE}** softkey. The program directory screen appears (see Figure 5.1).

(softkey level 1)



The control displays this main part program directory screen:

Figure 5.1
Part Program Directory



2. Select the part program you want to edit using two methods:

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

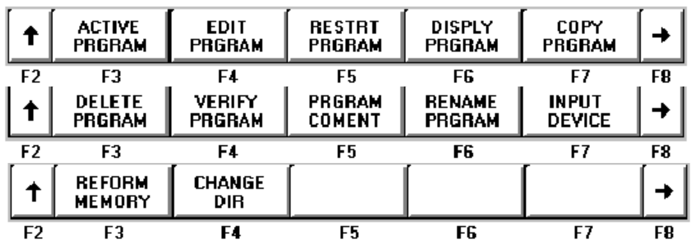
   or

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

Important: If you create a new program that is to be used as a subprogram, refer to chapter 9 on program names. Programs used as subprograms must have the letter O as the first character in the program name, followed by as many as 5 **numeric** characters.

3. Press the **{EDIT PRGRAM}** softkey.

(softkey level 2)

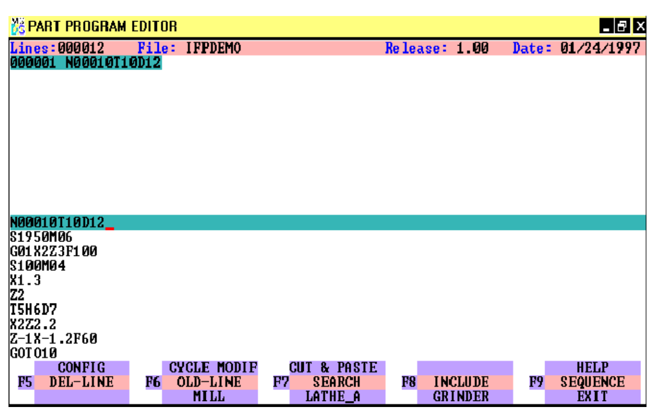


Editing a Part Program

Important: The following section describes the OCI Part Program Editor. You can create and edit part programs with either this part program editor or any ASCII editor that runs on your PC. In the event that your OEM or system installer selected an editor other than the one provided with this software, refer to that editor's documentation.

You can select programs to edit that are resident in the control's memory or on the hard or network drive of a PC. To select a part program from another directory, refer to page 2-39.

1. Select the **{EDIT PRGRAM}** softkey. The control displays the selected part program in the line editor:



Important: The {**CHANGE DIR**} softkey is deactivated once you are editing a part program.

Using Cut & Paste

The {**CUT & PASTE**} softkey allows you to select, delete, copy, and move the selected block range. Help, search, and exit options are available on every level in the cut and paste function. For more information about help, refer to page 5-3. For more information about search, refer to the previous section.

1. Select the {**CUT & PASTE**} softkey. A second level of layered softkeys appears.
2. Select the {**START SELECT**} softkey to begin choosing the lines that you want to include in the select range OR use the {**SEARCH**} softkey to search for the line that you want to include in the select range.

   Use the arrow or page up and down keys to include more part program blocks in the current select range. All selected data appears in yellow.

Two things can be done with the selected data:

- {**END SELECT**} - marks the data as "to be acted upon" and will allow you to delete, copy, or move the selected blocks.
- {**DESELECT**} - turns off the selection and returns to the {**START SELECT**} level

After you select the desired data, you have the following options to choose from:

Select:	To:
{**DELETE**}	delete the selected data. A confirmation box displays. To delete the selected data, enter **YES**. To keep the data selected without deleting it, enter **NO**. To confirm your choice, press [**ENTER**] on the numeric keypad.
{**COPY**}	insert the selected data after the currently highlighted edit line data. You can create multiple copies since the data remains as the select range data until you select the {**DESELECT**} softkey.
{**MOVE**}	move the selected data from its original location to the line(s) following the current edit line data. The data becomes deselected immediately after it is moved.
{**DESELECT**}	turn off the current selection.
{**SEARCH**}	search for a new string or line number. Refer to page 5-12 for more information.

3. To exit {**CUT & PASTE**}, select {**EXIT**}.

The maximum number of programs that you can have is **328**. To store a program, it will use at least one sector of memory. Use this table to find out how much part program space there is in your system.

If your system has	this is your part program storage
64K	150 meters
1MB	2589.6 meters/1,019,904 bytes
4MB	10540.4 meters/4,151,296 bytes

Including a Part Program

If you want to merge programs to the CNC, you must access them from your PC hard drive. Use the **{INCLUDE}** softkey to merge a whole part program into the program that you are editing.

Important: If you want to merge programs to the CNC, you must access them from your hard drive.

To include a part program:

1. Determine the point where you want to include another part program in the current program. Move the block that immediately precedes the inclusion point into the edit line.
2. Select the **{INCLUDE}** softkey. The system prompts you for a file name:

```
File name :
```

3. Enter the full path name (up to 48 characters) of the part program that you want to include.
4. Press **[ENTER]** on the numeric keypad or the **{INCLUDE}** softkey again to begin the include operation.

Saving and Exiting

To save a part program and exit from the editor:

1. Press the **{EXIT}** softkey.
2. Press **Y** to save the part program OR

   Press **N** to exit the editor without saving the program.
3. Press **[ENTER]** on the numeric keypad or select the **{EXIT}** softkey. The system returns to the program directory screen.

Using the Line Editor

As you activate the part program editor, the first screen you see is the line editor. Use the line editor to edit new or existing part programs.

Line Editor Dimensions

The line editor screen has 25 lines, which can display up to 90 characters each before wrapping to a new line. Part program blocks can contain a maximum of 127 characters, but can only display 89 characters on the edit line at a time. To view beyond the eighty-ninth character, use the arrow keys to scroll across the line display.

Navigating through the Line Editor

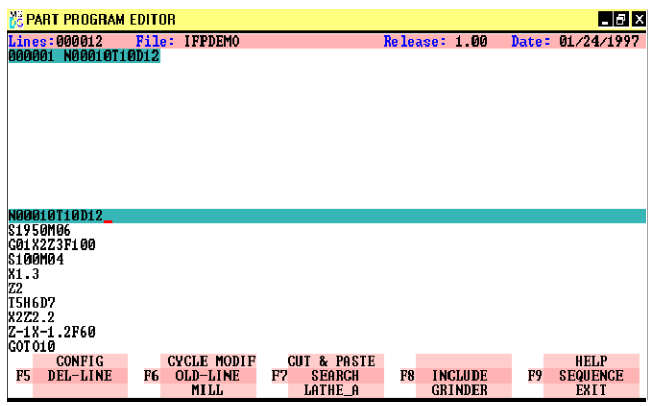
As we talk about using the editor, we'll be using terms that describe certain areas of the screen. Use this figure of a typical line editor screen to become familiar with the terms that we use:

Status line – displays the number of lines, file name, release number, and date.

Old-block line – displays what is in the edit line before your current edits are made. If the block in the edit line was incorrectly changed, press the {OLD LINE} softkey to restore the original block to the current program block.

Edit line – is the only place on the screen that allows you to make modifications. Changes to the edit line are not moved to the old-block line until you move to a new edit line.

Layered softkeys – display the softkeys for the line and cycle editors. The active layer is highlighted.



The OCI line editor is significantly different from the standard 9/Series editor. Use this table to learn how to navigate through the OCI editor:

If you want to:	Press
move to a specific line in the part program	the up and down arrow keys until you are at your desired location. The current line and line number are displayed on the second line of the screen.
scroll through the layered softkeys	F10
select a specific softkey	F10 until you highlight the layer that contains your selection. Then select the function key that corresponds to your selection.
move the cursor down one page	PgDn
move the cursor up one page	PgUp
go to the beginning of a block	Ctrl Left Arrow
go to the end of a block	Ctrl Right Arrow

If you want to:	Press
go to the beginning of the part program and place the first block of the part program on the edit line	Ctrl Home
go to the end of the part program and place the last block of the part program on the edit line	Ctrl End
delete the character left of the cursor	Back Space
open a new line after the block displayed in the edit line OR accept the new data to a current block	Enter
split the block in the edit line at the cursor position.	End

Entering Blocks

Since the edit line is the only place on the screen that allows you to add data, the cursor remains on this line. Once you enter a block in the edit line, close the block by using the up and down arrow keys.

You can change existing blocks only when they are within the edit line. To modify an existing block, move the cursor to the edit line using the up or down arrow key.

Important: You can also move an existing block to the edit line by using the following keys: **[PgUp]**, **[PgDn]**, **[Ctrl]** + **[Home]**, or **[Ctrl]** + **[End]**.

Creating a New Line

You can create a new line in two ways:

- by pressing **[ENTER]** with the cursor in any position. The new line is inserted below the block in the edit line.
- by positioning the cursor in front of the block and pressing **[END]**. This action inserts a line before the block in the edit line.

  Important: If you press **[END]** with the cursor positioned inside a block, the block will split. The only way to correct the split is to press the **{OLD-LINE}** softkey before performing any other action. This action restores the block that is in the edit line, but the last half of the split will be on the line below. To delete the last half of the split, highlight the split line and select the **{DEL-LINE}** softkey.

Creating a Blank Line

Similar to the standard front panel, OCI allows you to use blank lines in your part program. To insert a blank line in your part program, create a new line using **[ENTER]** or **[END]**. In the new line, use the space bar to add one blank space. When you exit the editor, the blank line will insert the end-of-block character.

Important: If you are using another editor, a carriage return will create an end-of-block character.

Deleting Lines

Use the **{DEL-LINE}** softkey to delete a block from the part program. The **{DEL-LINE}** softkey operates differently, depending on whether you press the softkey once or twice.

If you press {DEL-LINE}:	Then:
once	the contents of the block in the edit line are deleted. You can recover the deleted line with the {OLD-LINE} softkey
twice followed by the up/down arrow key	the block in the edit line is deleted with no possibility for recovery

Recovering Lines

You can recover lines using the **{OLD-LINE}** softkey in these cases:

- to restore the original text of a modified block in the edit line
- to recover a block deleted in the edit line after pressing the function key for the **{DEL-LINE}** softkey once
- to recover the original text when a block is split after pressing the **[END]** key while the cursor is inside of the block

Important: The **{OLD-LINE}** softkey does not delete the new block that was added. For details on deleting lines, see the previous section.

Numbering Lines

The **{SEQUENCE}** softkey allows you to number the lines of your part program.

1. Select the **{SEQUENCE}** softkey. The system prompts:

```
Start number : 1
Increment    : 1
```

2. At the **`Start number`** prompt, enter the number you want to start numbering at. The start number can range from 0 to 999000.

3. At the **`Increment`** prompt, enter the numerical increment that you want your lines to be numbered at (e.g., 10, 20, 30) using the numeric keypad. The increment can range from 0 to 200. Use 0 if you do not want the part program to be numbered or if you want to delete all of your line numbers.

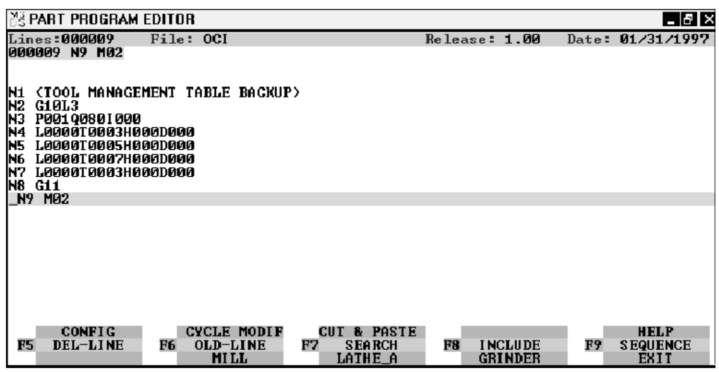
```
Start number : 10
Increment    : 10
```

4. To confirm line renumbering and exit the sequence screen, press **`[ENTER]`** or the **`{SEQUENCE}`** softkey.

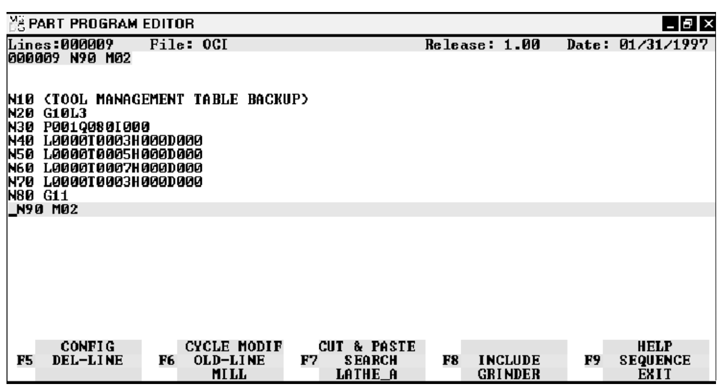
Important: Sequence only renumbers the N-word if it is the first work in the block.

Sequence does not store any targets of GOTO statements in its memory.

Before renumbering lines ➛



After renumbering lines ➛



Using the Search Softkey

The **{SEARCH}** softkey searches for strings or line numbers. When you select the **{SEARCH}** softkey, the display prompts you for string or line number to search for.

To search for a line number or a particular string:

1. Select the **{SEARCH}** softkey. While search is active, the softkey appears in yellow. The system prompts:

```
Character string : 
Line number      : 0
```

2. Use the **[RETURN]** or up and down arrow keys to choose between a character string or line number search. Enter either the character string or line number you want to search for.

If you choose this type of search:	Then:
character string	enter up to 12 characters and press **[ENTER]** followed by either the up or down arrow key on the numeric keypad. The **{SEARCH}** softkey remains active until you press it again.
line number	enter a line number of up to 6 digits (without the N character) and press **[ENTER]** on the numeric keypad. When found, the line appears on the edit line. If the search number is greater than the last number of the file, then the last line of the program appears on the edit line. The **{SEARCH}** softkey is deactivated once the search is complete.

If the character string or line number cannot be found, the system returns an error.

Important: You cannot enter negative numbers in the line number search block. Entering negative numbers returns an error. To remove the error, select **[ENTER]** on the numeric keypad.

3. If you want to abandon the search at anytime, select the **{SEARCH}** softkey.

Configuring the Cycle Editor

Before you use the cycle editor to edit part programs or select fixed cycles, you should customize the editor for your configuration. To customize the editor:

1. Select the **{CONFIG}** softkey to begin configuring the cycle editor. The first of two configuration screens appears:

```
----- CONFIGURATION PARAMETERS -----
Keystroke      : 80
Cycle screen 1 : CYCmil
Cycle screen 2 : CYClata
Cycle screen 3 : CYCgrd
                                1 of 2
```

2. To move through the fields, press **[ENTER]** on the nonnumeric keypad or use the up and down arrow keys.

Use this table to set the parameters for your configuration:

Field	Description
Keystroke	This field displays the number of keystrokes that can be completed before they are saved in the recovery file. Values range from 0 to 999, with the default set at 80. If you enter 0, there is no recovery file management.
Cycle screen 1-3	In these fields, tell the system what cycles you want to display in the editor softkeys. You can enter up to three cycles in any combination. File names you can enter are: • CYCLata Cycles for type A lathes • CYCLatb Cycles for type B lathes • CYCLatc Cycles for type C lathes • CYCmil Mill Cycles • CYCgrd Grinder Cycles (surface and cylindrical)

Important: Recovery file management is used only if the DOS window (editor) crashes before the editor is exited.

3. After you configure the global parameters screen, press **[PgDn]** to move to the next configuration screen:

```
--------- CYCLE PARAMETERS ---------
        AMP name    Symbolic
Name1      X           X
Name2      Y           Y
Name3      Z           Z
Name4      I           I
Name5      J           J
Name6      K           K
Name7
Name8
Name9                          2 of 2
```

Important: The order of the AMP names do not necessarily reflect the axis names configured in AMP.

4. If your machine configuration does not match the default parameters in the AMP column, enter your normally configured axis names in the AMP name column.

   Important: We recommend that you do not change the symbolic names. The cycle editor provides you with symbolic names that the cycle editor recognizes (i.e., X, Y, Z, A, B, C, U, V, W) to represent the axis name configured in AMP.

   Important: 9/PC axis names beginning with $ cannot be used in the OCI cycle editor.

   To move from field to field, use the **[ENTER]** key on the nonnumeric keypad. Once you enter all axes names, press **[ENTER]** on the numeric keypad to save your changes.

   Important: Cycle names must be entered in capital letters.

5. Press the **{CONFIG}** softkey again to return to the line editor.

Using the Cycle Editor (Quick View)

Included in the part program editor is a fixed-cycle editor, which automatically creates common tool paths using dimensions specific to the part you want to create. Use this editor to create programs using 9/PC cycles.

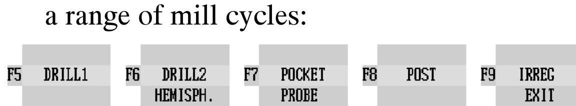
When you configured the cycle editor in the previous section, you selected the cycles that would be available in the cycle editor. For our example, this manual assumes your editor is configured to display the mill cycles. Your editor may be configured differently.

Important: Reducing the window size of the cycle editor using **[ALT]** + **[ENTER]** causes the editor to become frozen. To reactivate the editor, you must maximize the window by repeating the **[ALT]** + **[ENTER]** action.

Displaying the Cycle Prompts

1. If necessary, use **[F10]** to scroll to the layer in the softkey area that displays the cycle softkeys.

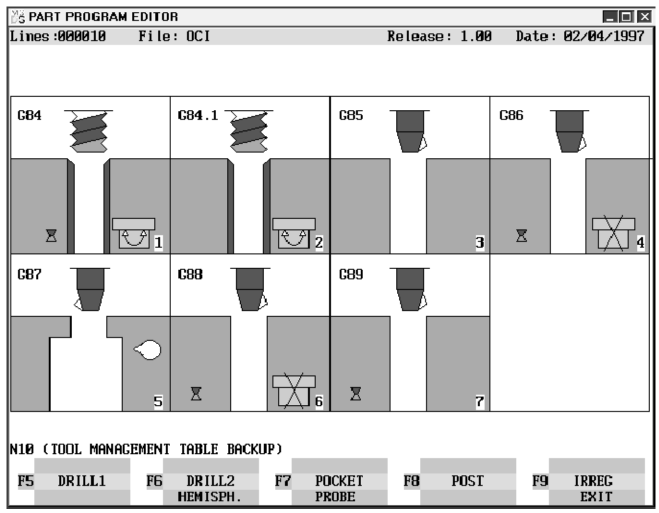
2. Press the function key that selects the type of cycles that you want. In our example, pressing **[F6]** causes the system to display a range of mill cycles:



The cycles for each control type are broken into different categories. In our example, the mill provides:

- Drilling Cycles
- Pocket Cycles
- Post Cycles
- Irregular Cycles
- Hemispherical Pocket Cycles
- Probe Cycles

3. To display a set of cycles, press the function key that selects the set of cycles that you want to work with. In our example, pressing `[DRILL2]` displays:



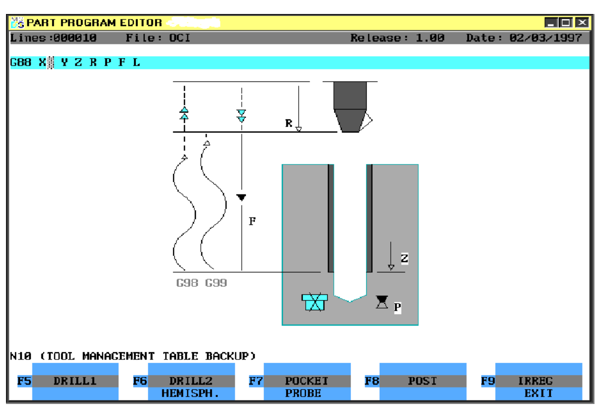
4. To select a cycle, enter the number displayed in the lower right corner of the graphic box. The system displays the format of the cycle and detailed graphic representation of the cycle. For a definition of each graphic, refer to page 5-16. In our example, we selected cycle 6 – G88:

Cycle format - are all parameters that need values for the cycle to work. Optional fields allow you to bypass it without entering data. Use the right arrow key to move to each parameter.

Cycle graphics - depict how the letter parameters affect basic cycle motion. The letter is highlighted as its associated parameter is entered.

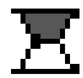
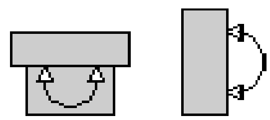
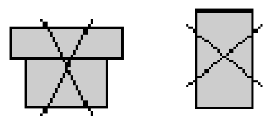
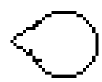
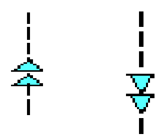
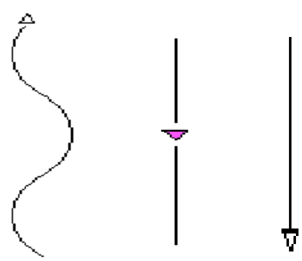
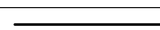
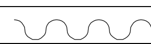
Old-block line - displays what is in the edit line before any edits are made.

Layered softkeys – display the softkeys for the line and cycle editors. The active layer is highlighted.



5. To return to the cycle screen, select [F6] again. If you wish to choose a different cycle, select that softkey.

The table below defines the graphics used in the cycle editor:

Graphic	Definition
	Dwell
	Spindle reverse (clockwise/counterclockwise)
	Stop spindle
	Shift direction
	Rapid move
	Feed move
	Cutting feed
	Rapid feed
	Manual operation

Available Cycles

The following table lists the available 9/PC cycles.

Control	Cycle Set	Cycles
Lathe Type A	Single Turning	G90, G94
	Compound Turning	G70, G71, G72, G73
	Threading	G32, G34, G76, G92
	Groove	G74, G75
	Drill1	G81, G82, G83, G83.1, G84, G84.1, G84.2, G84.3
	Drill2	G85, G86, G86.1, G87, G88, G89
Lathe Type B	Single Turning	G77, G79
	Compound Turning	G70, G71, G72, G73
	Threading	G33, G34, G76, G78
	Groove	G74, G75
	Drill1	G81, G82, G83, G83.1, G84, G84.1, G84.2, G84.3
	Drill2	G85, G86, G86.1, G87, G88, G89
Lathe Type C	Single Turning	G20, G24
	Compound Turning	G72, G73, G74, G75
	Threading	G33, G34, G21, G78
	Groove	G76, G77
	Drill1	G81, G82, G83, G83.1, G84, G84.1, G84.2, G84.3
	Drill2	G85, G86, G86.1, G87, G88, G89
Mill	Drill1	G73, G74, G74.1, G76, G81, G82, G83
	Drill2	G84., G84.1, G85, G86, G87, G88, G89
	Pocket	G88.1, G88.2
	Post	G88.3, G88.4
	Irregular	G89.1, G89.2
	Hemispherical Pocket	G88.5, G88.6
	Probe	G38, G38.1

Important: Drilling cycles include tapping and boring.

6. At the cycle format line, use the left and right arrow keys to move through the cycle screen to enter your desired values. Since most parameters are nonoptional, values must be entered sequentially.

7. After entering all parameter values, press **[ENTER]** or the **{EXIT}** softkey to confirm the data at the input lines. You will then return to the cycle editor screen.

   The most recently entered block appears at the bottom of the cycle editor screen.

8. To exit the cycle editor, press the **{EXIT}** softkey.

Modifying an Existing Cycle Block

Once you insert a cycle block into a program, you can still modify it with the cycle editor, even though the program resides in the line editor. To modify an existing cycle block using the cycle editor:

1. Place the cycle block that you want to modify in the edit line.

2. Press the **{CYCLE MODIFY}** softkey to automatically activate the cycle editor.
3. Edit the cycle block by replacing the unwanted values with your replacement values.
4. Press **[ENTER]** on the numberic keypad to enter the new cycle block.

Important: When modifying parameters, the source line cycle parameter order (system-inserted characters) must be identical to the order in the cycle editor.

Deleting a Program {DELETE}

To delete part programs stored in memory:

ATTENTION: Once you delete a program from memory, it can not be recovered. Abort the delete program operation by pressing the **{DELETE NO}** softkey.

1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)

F2	F3	F4	F5	F6	F7	F8
	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
		ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the **{DELETE PRGRAM}** softkey.

(softkey level 2)

F2	F3	F4	F5	F6	F7	F8
↑	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. Select one of these two choices:
   - Key in the the program name and press the **{DELETE YES}** softkey
   - Move the block cursor down until the desired program is in reverse video and press the **{DELETE YES}** softkey.

(softkey level 3)

F2	F3	F4	F5	F6	F7	F8
↑		DELETE YES	DELETE NO			

You can delete all programs at once by formatting the RAM disk as described in chapter 2.

Renaming Programs {RENAME}

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

1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)

F2	F3	F4	F5	F6	F7	F8
	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
		ERROR MESAGE	PASS-WORD	SWITCH LANG		→

2. Press the **{RENAME PRGRAM}** softkey.

(softkey level 2)

F2	F3	F4	F5	F6	F7	F8
↑	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. 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.
- Press the **{RENAME YES}** softkey. To abort the operation press the **{RENAME NO}** softkey.

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

(softkey level 3)

F2	F3	F4	F5	F6	F7	F8
↑	RENAME YES	RENAME NO				

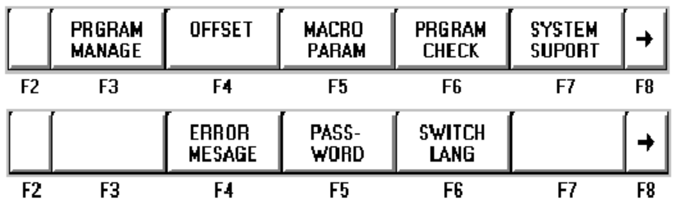
Displaying a Program {DISPLY PRGRAM}

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

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

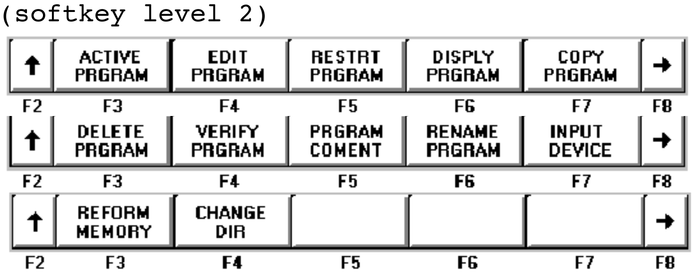
1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)



2. Select the input device using the **{INPUT DEVICE}** softkey (as described 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.
4. Press the **{DISPLY PRGRAM}** softkey.

(softkey level 2)



5. To scroll the part program blocks, hold down the **[SHIFT]** key, then press the up or down cursor keys.
6. To end the displaying operation, press the exit {⇑} softkey. The display returns to the program directory screen.

Displaying Comments {PRGRAM COMENT}

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

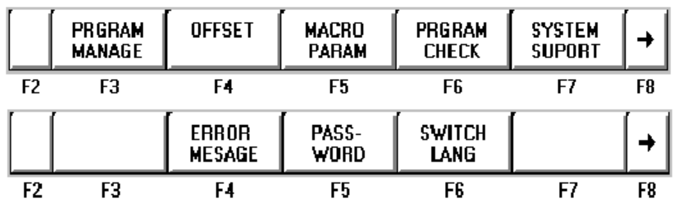
Important: These are not normally the same as a comment block made within a part program. Comment blocks are described on page 9-5. If a comment block is assigned as the **first** block of the part program, it will be displayed on the program directory screen as a comment. Any other comment blocks have no affect on the comment display.

Important: The **{PRGRAM COMENT}** feature does not allow you to add comments to programs stored in the hard drive directory.

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

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

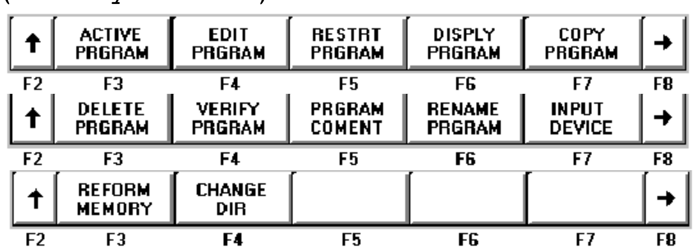
(softkey level 1)



2. Use the up or down cursor keys to select the program to add the comment to. The selected program name appears in reverse video.

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

(softkey level 2)



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 described on page 2-10, or the old comment can be deleted by pressing the **[BACKSPACE]** key.

4. Type in the new comment or edit the old comment by keying it in using the keyboard. Up to 28 characters can be entered on single process systems, and 14 characters on a dual processing system.

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

Copying Programs {COPY PRGRAM}

This section describes making a duplicate of a part program in control memory.

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

1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)

F2	F3	F4	F5	F6	F7	F8
	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
		ERROR MESAGE	PASS- WORD	SWITCH LANG		→

2. Press the **{COPY PRGRAM}** softkey.

(softkey level 2)

F2	F3	F4	F5	F6	F7	F8
↑	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. Key in or cursor down to the program name of the program to be copied.

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

COPY: **FROM_NAME,TO_NAME**

5. Press the **{MEM TO MEM}** softkey.

(softkey level 3)

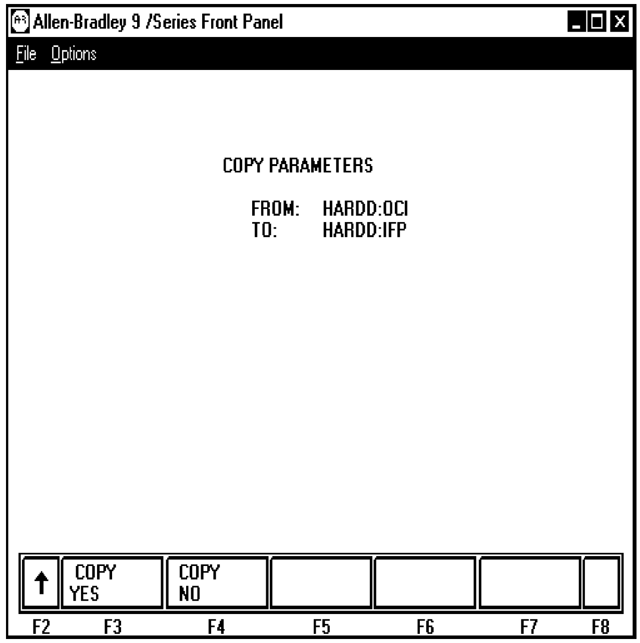
F2	F3	F4	F5	F6	F7	F8
↑					MEM TO MEM	

6. Select the destination you want your part program to be copied to.

(softkey level 4)

F2	F3	F4	F5	F6	F7	F8
↑	TO MAIN	TO PROTEC	TO HARDD			

A similar screen appears:



7. Select softkey **{COPY YES}** or **{COPY NO}**. **{COPY YES}** copies the part program, while **{COPY NO}** aborts the copy operation.
8. If you want to verify that the copied program identically matches the original, use the **{VERIFY PRGRAM}** feature described in chapter 8.

Important: You can not copy part programs can not copy part programs stored on the main part program directory to the protectable directory and vice versa.

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

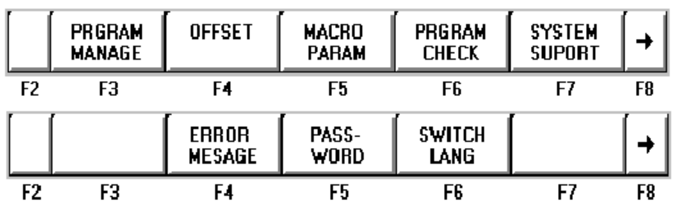
You can only call a protected program from a main program using a subprogram, G-code macro, or M-code macro call without access to the **{CHANGE DIR}** softkey.

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.

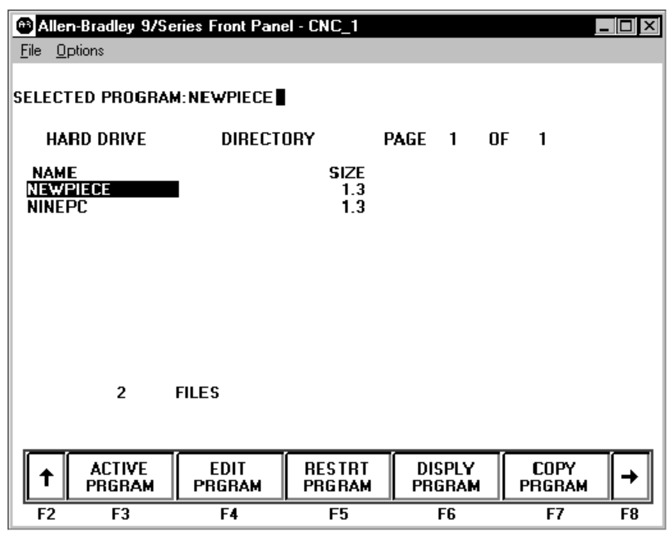
To access the protectable part program directory:

1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)

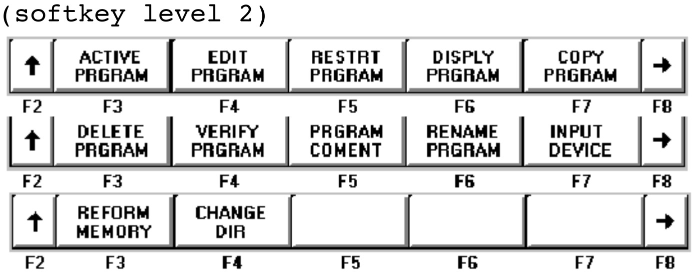


The control displays the hard drive directory screen:



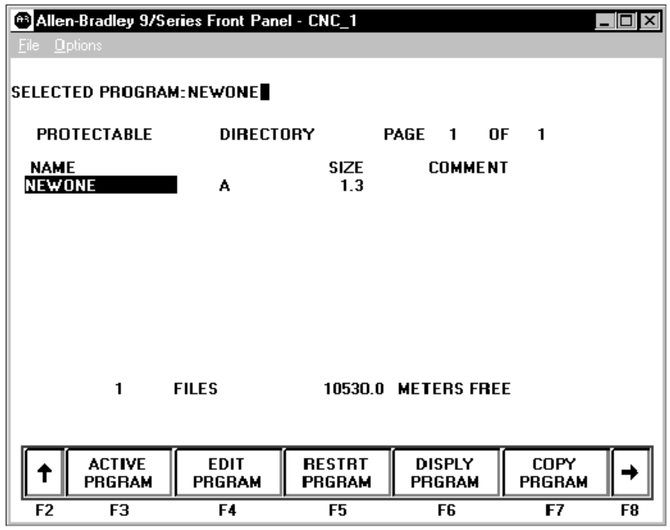
2. Press the **{CHANGE DIR}** softkey.

(softkey level 2)



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:



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)

END OF CHAPTER

Editing Programs Offline

Chapter Overview

This chapter describes how to use the Offline Development System (ODS) to edit part programs. Major sections include:

Use the Offline Development System (ODS) to write or edit part programs. Once you complete these part programs, download them to the PC that contains the 9/PC control. Programs that already exist on the control can be uploaded to the PC for editing or backup. You can edit programs on ODS by using the screen or text editor that is configured in ODS. You can purchase enhancements to this feature in a Mini-DNC package from Allen-Bradley. If you purchased the Mini-DNC package, see its accompanying documentation.

We make these assumptions:

- ODS is installed on the PC that the 9/PC is installed in or that on a PC that is networked with the PC where the 9/PC is installed
- a compatible screen or text editor has been configured using the Text Editor Setup option of the F5-Configuration menu
- the programmer understands the basics of the ODS system and how it operates

For additional information, see the ODS manual, publication MCD-5.1.

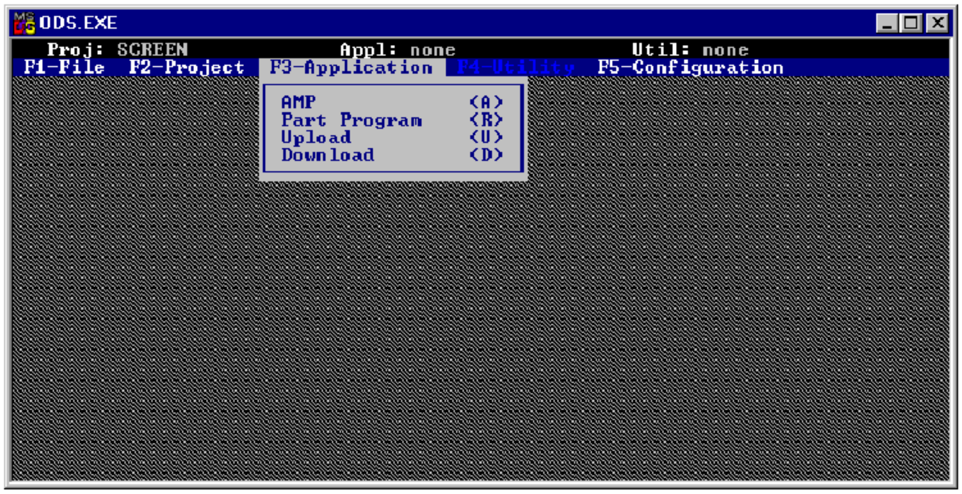
Important: Be aware that some features described here may not be available with your ODS. Some may require the purchase of the Mini-DNC package to be functional.

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 PC displays this screen:



3. Press [R] to select the Part Program option.

The status line of the screen displayed by the PC shows that the Part Program application has been selected.

Editing Part Programs Offline

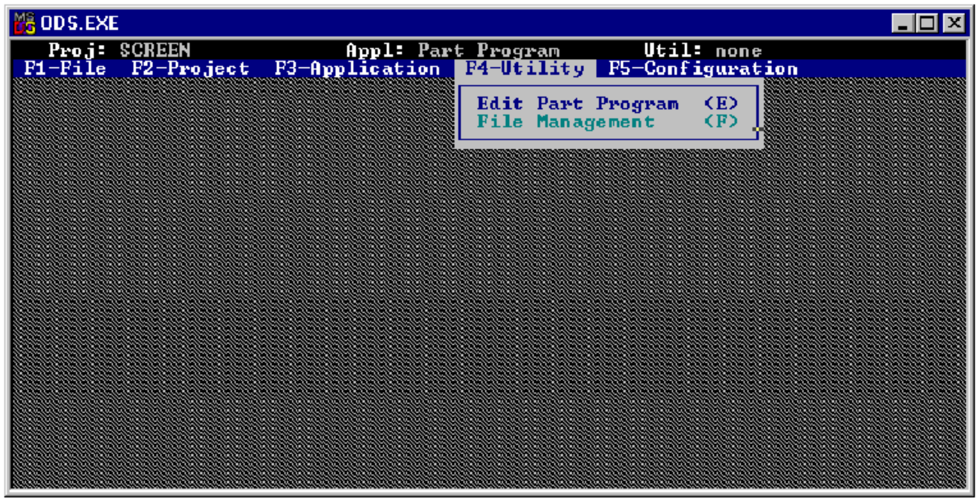
Use the Edit Part Program utility of ODS to edit part programs on a PC. Programs that already exist on the control can be uploaded to the PC for editing. These programs or programs created using ODS can be edited using the screen or text editor that is configured in ODS.

To edit part programs thorough ODS:

1. Select the Part Program Application. Refer to the previous section.

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

   The PC displays this screen:



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

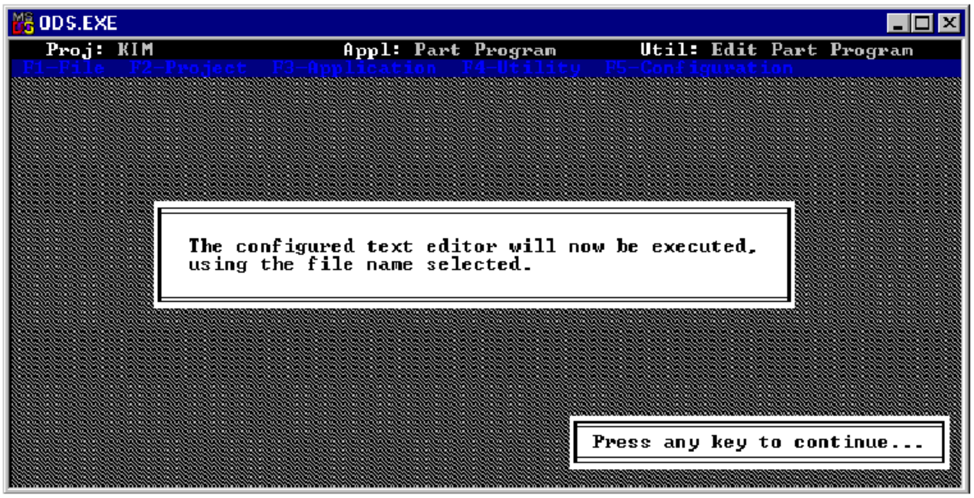
   The PC displays this screen:

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 you select a file, the PC 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.

You can find details about programming blocks in later chapters.

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.

The maximum number of programs that you can have is **328**. To store a program, it will use at least one sector (2048bytes/1.3m) of memory. Use this table to find out how much part program space there is in your system.

If your system has	this is your part program storage
64K	150 meters
1MB	2589.6 meters/1,019,904 bytes
4MB	10540.4 meters/4,151,296 bytes

Downloading Part Programs from ODS

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

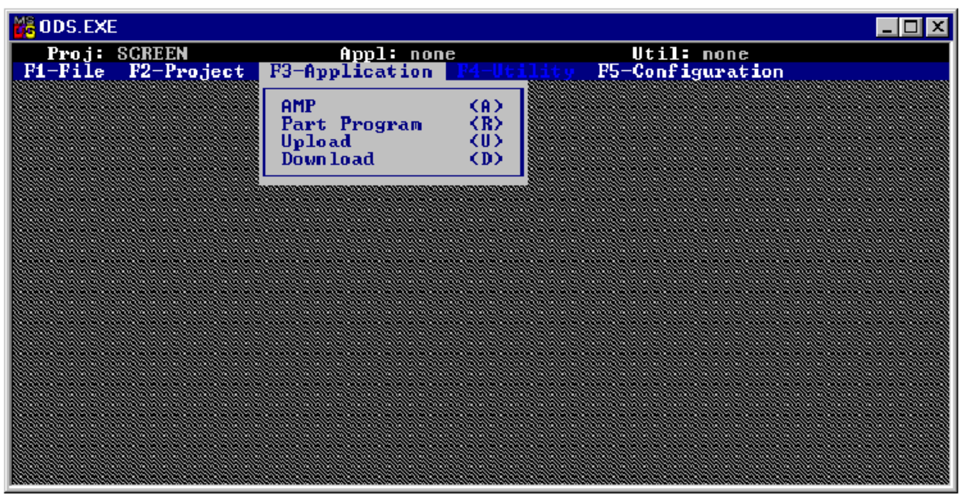
Important: When you download a program from ODS to the control, it is automatically inserted into the normal program directory on the control. The control automatically inserts the end of block statements ";" at the end of each line when the program is downloaded to the control.

Important: Use the Download application of ODS to download part programs to your Main and Protectable directories only.

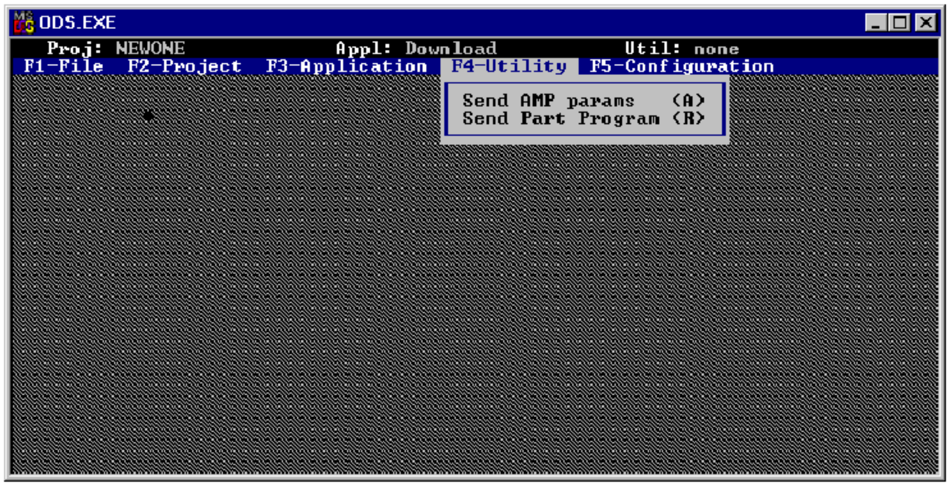
To download a part program from ODS to the control's memory, follow these steps:

1. Make sure you are at ODS' main menu line.
2. Press [F3] to pull down the Application menu.

   The PC displays this screen:

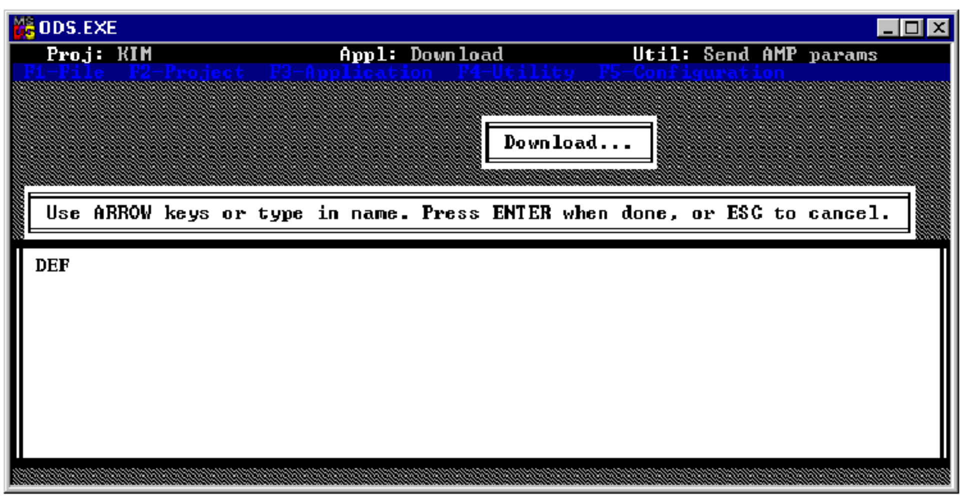


3. Use the arrow keys to highlight the Download application, then press **[ENTER]** or press **[D]**.
4. Press **[F4]** to pull down the Utility menu.



5. Use the arrow keys to highlight the Send Part Program option, then press **[ENTER]**, or press **[R]**.
6. Use the arrow keys to highlight the download destination or press the letter that corresponds to the download destination. When selected, press **[ENTER]**.

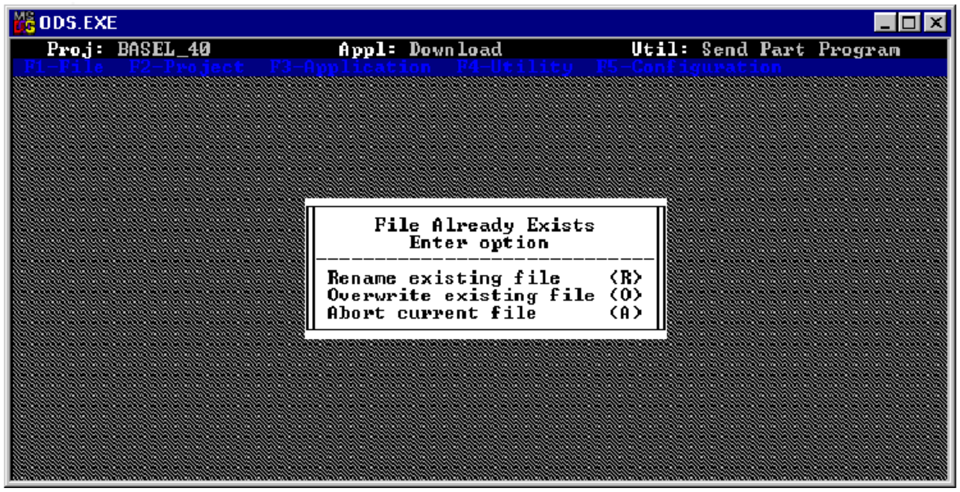
   The PC displays the part program files that are stored in the active project directory of the PC:



7. Use the arrow keys to highlight the name or type in the part program name to download, then press **[ENTER]**.

Important: You can upload and download more than one part program by using wildcards (“*” or “?”) in place of all or part of a file name. Refer to the PC’s DOS manual for additional information about using wildcards.

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

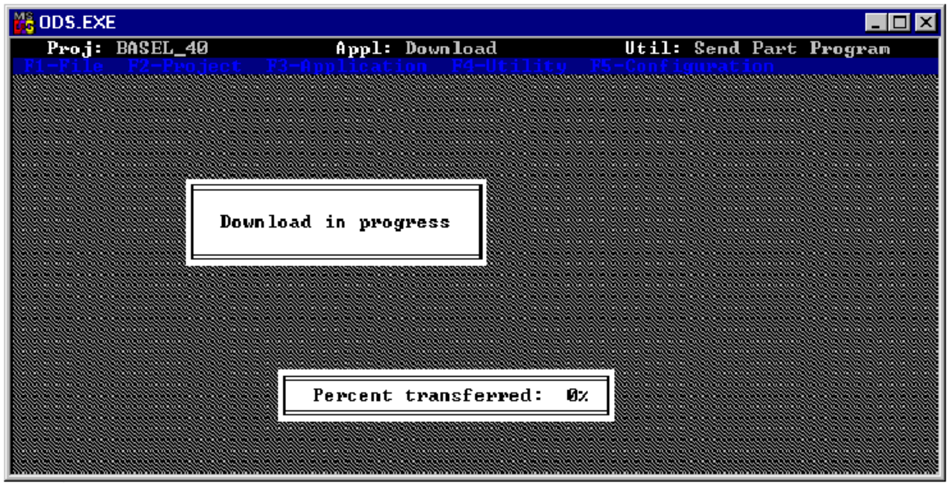


Important: The currently active or open part program on the control can not be renamed or overwritten during a download procedure.

If you select this option:	This happens:
Rename	the PC renames the existing file, which has the same name as the file being uploaded, on the PC.The PC displays the part program files stored on the PC Type in the new name for the existing part program on the control.
Overwrite	the part program file being downloaded overwrites the file having the same name on the control.
Abort	the download process is discontinued and the PC 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.

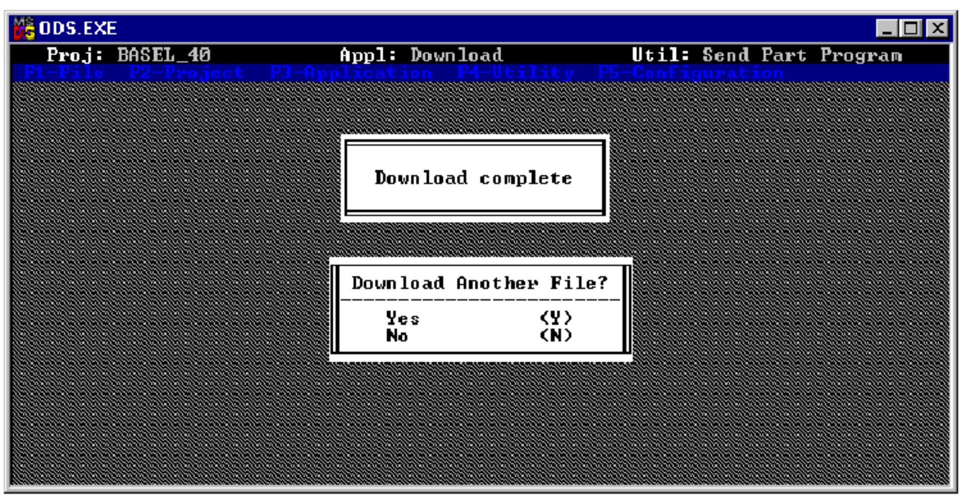
After selecting the Rename or Overwrite option, or if the file being downloaded did not already exist on the control, the PC displays this screen:



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

When you download a program to a control, the control does not display a message to indicate that a download is taking place. If you download a large program it can take several minutes for the control to complete the download. As the program downloads, the control updates the size of the program shown.

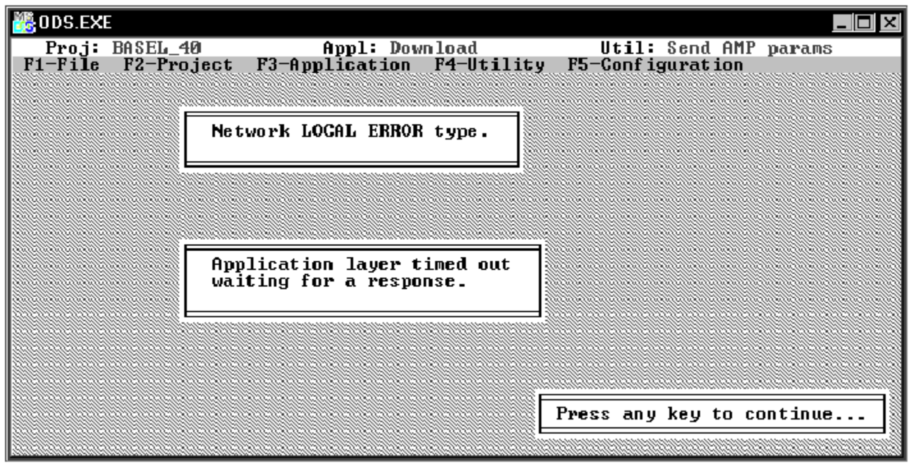
When the download process is complete, the PC displays this screen:



8. Select "Yes" or "No."

If you select:	Then:
Yes	the system prompts the programmer through the download procedure again
No	the PC returns to ODS the main menu line.

If the PC was unable to complete the download procedure in the allotted time frame, it displays this screen:



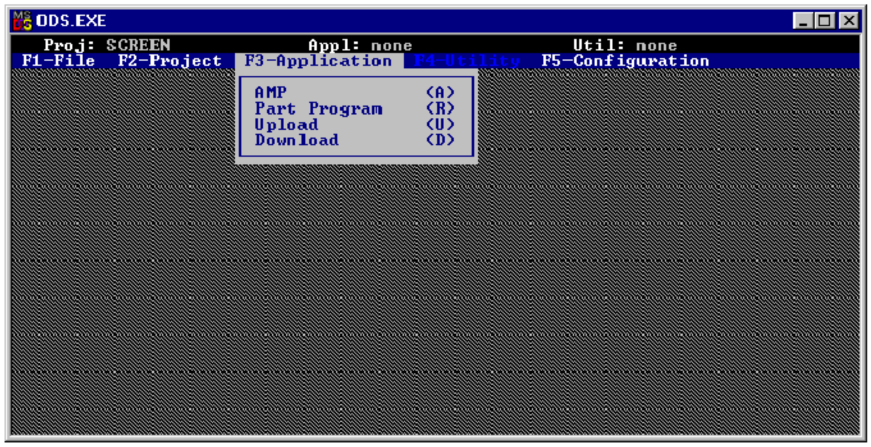
Pressing any key causes the PC to return to the ODS main menu.

Upload Part Programs to ODS

The programmer can upload a part program from the control's memory to the PC by using the ODS Upload application. This allows the part program to be edited or stored on the PC.

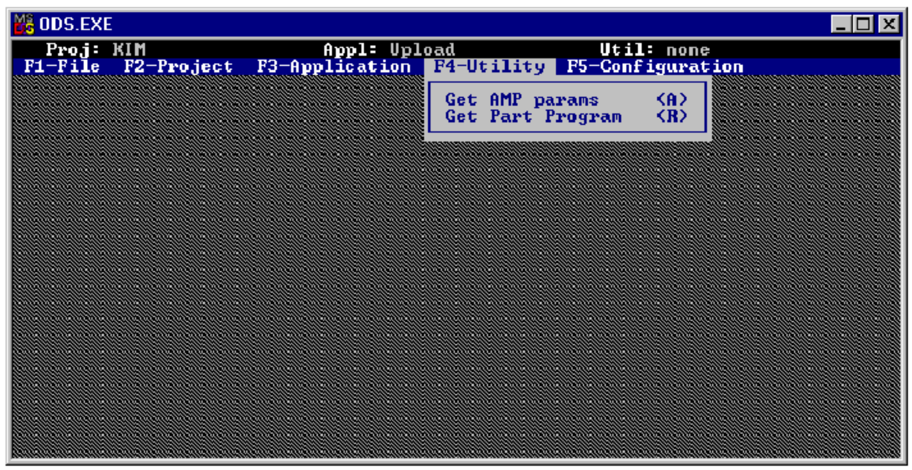
1. Interface the PC with the control. Refer to your *9/PC Installation and Integration Manual* (8520–9.1).
2. Return to the main menu line of ODS.
3. Press [F3] to pull down the Application menu.

The PC displays this screen:



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

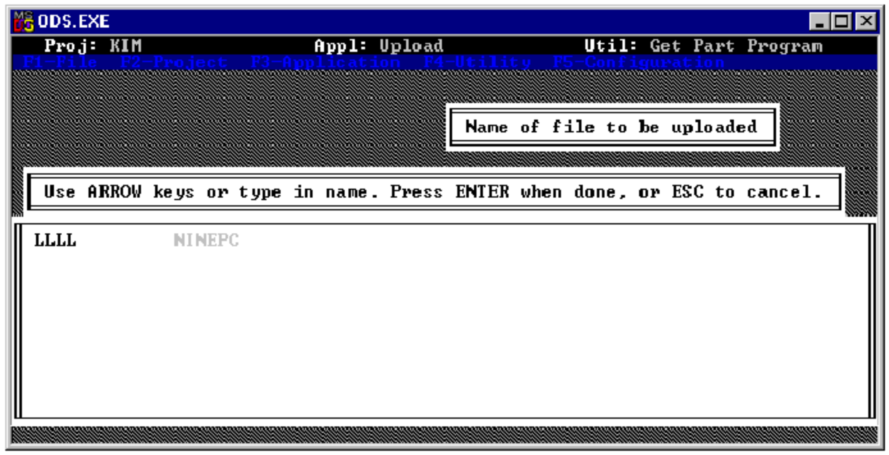
5. Press [F4] to pull down the Utility menu.



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

7. Use the arrow keys to highlight the upload origin, then press [ENTER] or press the letter that corresponds to the upload origin.

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

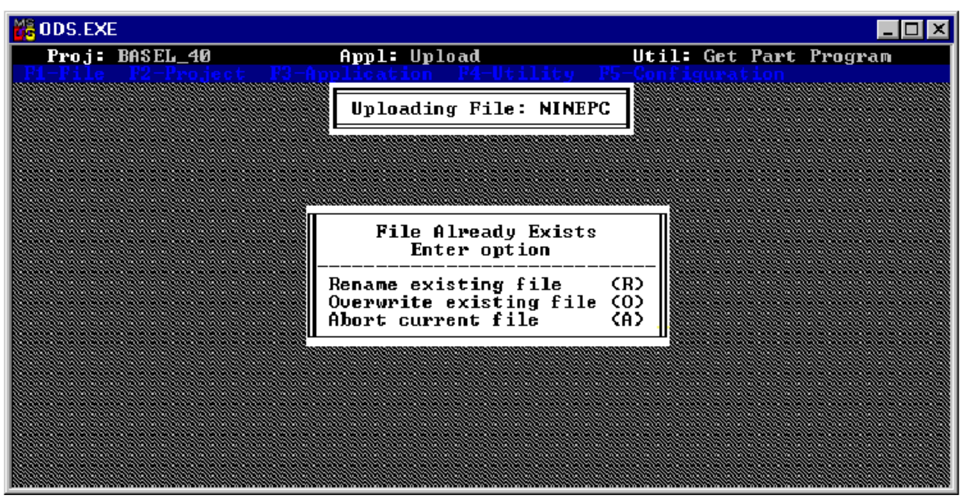


8. Use the arrow keys to highlight the name of the part program to be uploaded to the PC or type in the part program name, then press **[ENTER]**.

When you upload a program from the control, the control does not display a message to indicate that an upload is taking place. If you upload a large program it may take several minutes for the upload to complete. If you try to edit the program while it is uploading you see an error message that says the program is already open. You have to wait until the upload is complete to edit the program.

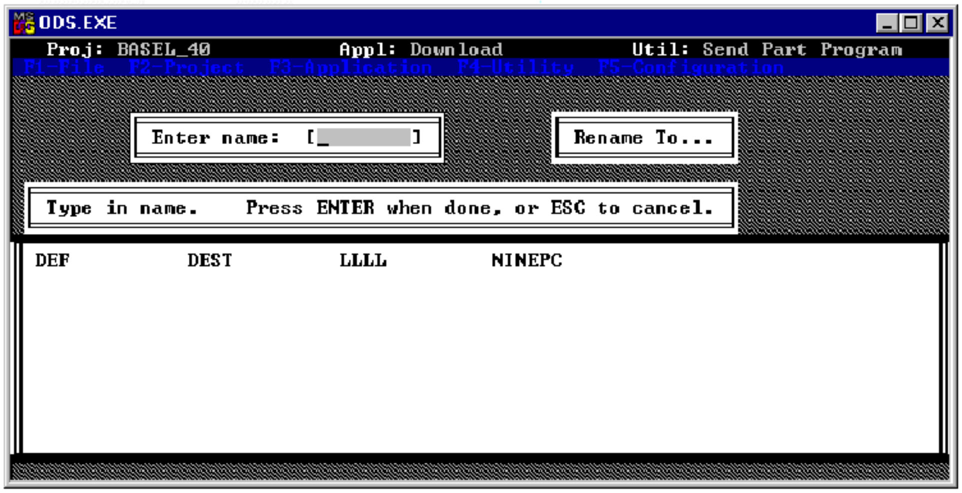
Important: You can upload and download more than one part program by using wildcards ("*" or "?") in place of all or part of a file name. Refer to the PC's DOS manual for additional information about using wildcards.

If the selected part program already exists on the PC, the PC displays this screen:



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

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

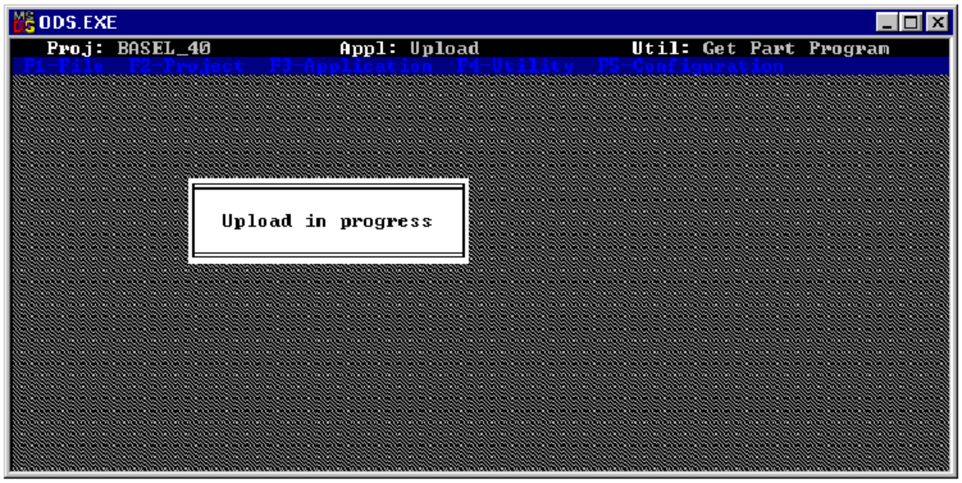


9. Type in the new name for the existing part program file on the PC.

If you select this option:	This happens:
Overwrite	the part program file being uploaded overwrites the file having the same name on the PC.
Abort	the upload process is discontinued and the PC 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.

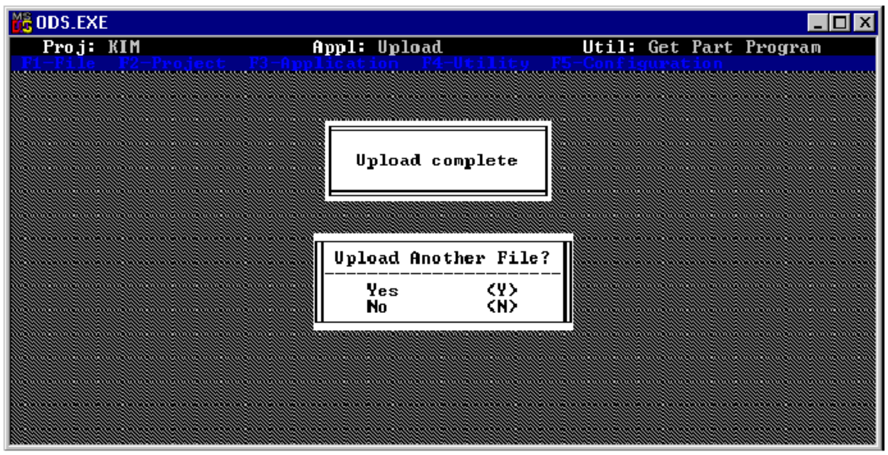
If the name of the part program that was entered does not exist on the PC or the Overwrite option was selected the PC displays this screen:



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

When you upload a program from the control, the control does not display a message to indicate that an upload is taking place. If you upload a large program it can take several minutes for the upload to complete. If you try to edit the program while it is uploading, you see an error message that says the program is already open. You have to wait until the upload is complete to edit the program.

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



Select “Yes” or “No.”

If you select:	Then:
Yes	the system prompts the programmer through the upload procedure again
No	the PC returns to ODS the main menu line.

END OF CHAPTER

Running a Program

Chapter Overview

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

Selecting Special Running Conditions

The following subsections describe some of the functions available on the 9/PC that affect how it executes a program. The use of these "special running conditions" is optional. They are activated either through the MTB panel, through programming, or some combination of the two.

Block Delete

When programming a slash "/" followed by a numeric value (1-9) anywhere in a block, 9/PC skips (does not execute) all remaining programmed commands in that block if a optionally installed switch on the MTB panel is activated. If the "block delete type" parameter in AMP is set to "delete whole," then the entire block is deleted regardless of the position of the block delete character. For details on the block delete feature, refer to page 9-5.

To activate the block delete feature, your system installer may have installed a switch corresponding to a block delete number (refer to the documentation prepared by your system installer).

Miscellaneous Function Lock

When the MISCELLANEOUS FUNCTION LOCK is made active, 9/PC displays M-, second auxiliary functions (B-codes), S-, and T-codes in the part program and activates the corresponding Tool Wear Offset, except for M00, M01, M02, M30, M98, M99.

To activate the MISCELLANEOUS FUNCTION LOCK feature, your system installer may have installed a switch corresponding to the MISCELLANEOUS FUNCTION LOCK feature (refer to documentation prepared by your system installer).

Sequence Stop {SEQ STOP}

Use this feature to cause automatic program execution to stop after a specified block. This block is determined by assigning its sequence number (N-word) as the sequence stop block. This sequence number may be entered before or after part program execution begins. If this sequence number is entered after program execution begins, it must be entered before 9/PC executed that block. If it is not entered before the block is executed, it is ignored and execution continues as normal.

Automatic execution stops after the sequence stop block is completed. 9/PC is placed in cycle stop. To resume execution from the current position in the program, press the **<CYCLE START>** button.

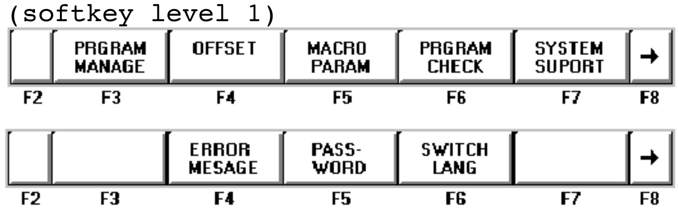
Important: Once you enter a sequence stop number for a program, it remains active for all programs that are executed until it is replaced with a different sequence stop number, or power is lost. Not entering a value for the sequence stop number or entering a value of zero results in the sequence stop function being canceled.

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 already have been selected for automatic execution as described in chapter 7.

(softkey level 1)



2. Press the **{ACTIVE PRGRAM}** softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
F2	F3	F4	F5	F6	F7	F8
↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→
F2	F3	F4	F5	F6	F7	F8
↑	REFORM MEMORY	CHANGE DIR				→
F2	F3	F4	F5	F6	F7	F8

3. Press the **{SEQ STOP}** softkey.

(softkey level 3)

↑	DE-ACT PRGRAM	SEARCH	MID ST PRGRAM			→
F2	F3	F4	F5	F6	F7	F8
↑	SEQ STOP				TIME PARTS	→
F2	F3	F4	F5	F6	F7	F8

4. Key in the sequence number where you want automatic operation in the part program to stop, then press the **[ENTER]** key.

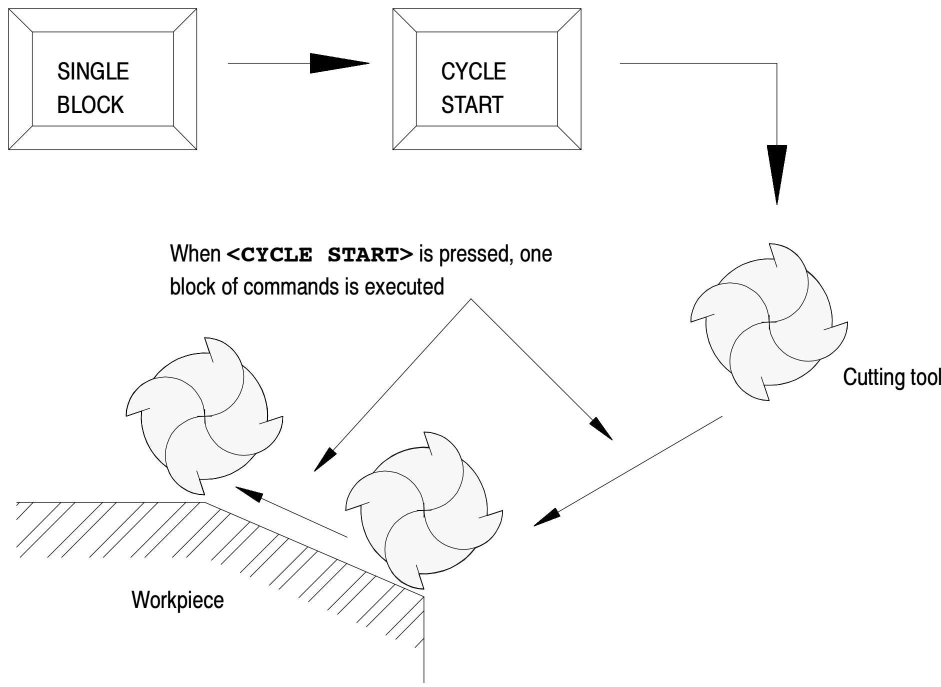
Important: 9/PC stops automatic operation after it completes the commands in the block.

5. Press the **<CYCLE START>** button to continue execution of the program from the point at which program execution was stopped.

Single Block

In single block mode, 9/PC executes the part program block by block. Each time you press the **<CYCLE START>** button, 9/PC executes one block of commands in the part program when in single block mode.

Figure 7.1
Single Block



To activate the single block function, press the **<SINGLE BLOCK>** button. The light inside the button lights up when active.

If you press the **<SINGLE BLOCK>** button while 9/PC is running a part program in the automatic or MDI mode, 9/PC activates the single block function after it completes the commands in the block that is currently being executed.

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

Selecting a Part Program Input Device

Before selecting a part program, you must tell 9/PC where the part program currently resides. The program can reside:

- in the control's RAM memory
- in the PC's hard disk

To view the the part program input device softkey:

1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)

F2	F3	F4	F5	F6	F7	F8
	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→

F2	F3	F4	F5	F6	F7	F8
		ERROR MESAGE	PASS-WORD	SWITCH LANG		→

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

(softkey level 2)

F2	F3	F4	F5	F6	F7	F8
↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→

F2	F3	F4	F5	F6	F7	F8
↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→

F2	F3	F4	F5	F6	F7	F8
↑	REFORM MEMORY	CHANGE DIR				→

By default, **{FROM MEMORY}** is the location that the part program is read from. Clicking on the **{FROM MEMORY}** softkey returns you to the previous level.

F2	F3	F4	F5	F6	F7	F8
			FROM MEMORY			

Important: The 9/PC has three directories available for storage and part program execution: main, protectable, and the PC's hard drive. To select a different location for part programs to be executed from, use the **{CHANGE DIR}** softkey. For more information about the **{CHANGE DIR}** softkey, refer to page 2-39.

To activate a part program, it must be selected as discussed in the following section.

Selecting a Part Program

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

Important: Consider the following when selecting a program:

- 9/PC cannot select a program for execution if that program file is still open for editing. Refer to chapter 5 to learn how to exit the edit mode.
- Your system installer may have written logic to allow some other method of part program selection. Refer to 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 selects the part program out of control memory.

- If a program was previously activated and not deactivated, 9/PC cannot select a different part program. If you want a different part program, you must first deactivate the active program as described on page 7-7. You can use a different method to select a program; it is described in this chapter.

To select a program for automatic execution:

1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)

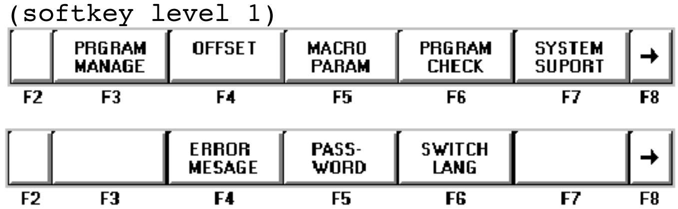
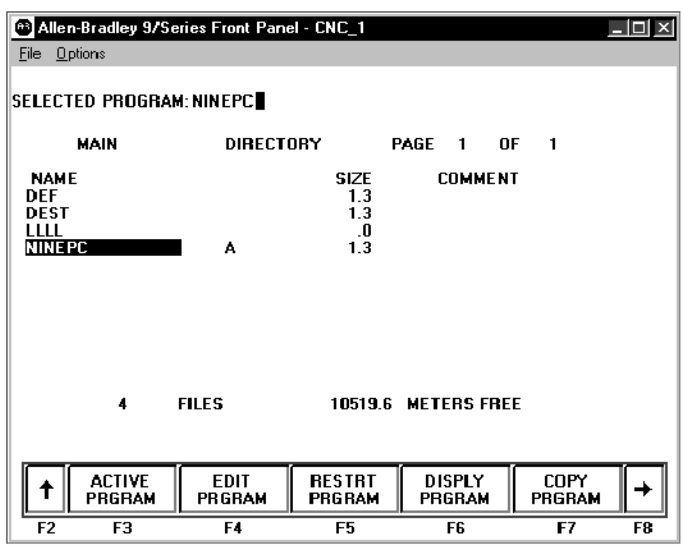


Figure 7.2
Part Program Directory

This screen appears:



Important: This screen shows program NINEPC as active and being edited. Make sure no part program is currently already active. If a part program is currently active, 9/PC cannot select a different part program until the currently active one is deactivated. Refer to page 7-7 to learn how to deactivate a part program.

If a program is:	This appears to the right of a program name:
active	A
being edited	E

2. Key in the name of the part program to activate. If the program is being selected from control memory, the ⇑ or ⇓ cursor keys may be used to select the program to activate from the directory screen.

3. Press the **{ACTIVE PRGRAM}** softkey to activate the selected program. 9/PC displays the part program name, followed by the first few blocks of the selected program.

   Important: Note that the following softkey level 2 indicates that the control is using control memory as an input device.

```
(softkey level 2)
```

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
F2	F3	F4	F5	F6	F7	F8

↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→
F2	F3	F4	F5	F6	F7	F8

↑	REFORM MEMORY	CHANGE DIR				→
F2	F3	F4	F5	F6	F7	F8

Important: Before you can execute the program, you must place 9/PC in automatic mode.

Deselecting a Part Program

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

1. Press the **{PRGRAM MANAGE}** softkey. 9/PC displays the program directory screen.

```
(softkey level 1)
```

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
F2	F3	F4	F5	F6	F7	F8

		ERROR MESAGE	PASS-WORD	SWITCH LANG		→
F2	F3	F4	F5	F6	F7	F8

Important: If a program is active, an "A" appears to the right of the program name. If a program is being edited, an "E" appears to the right of the program name. The screen on page 7-6 shows program NINEPC as active and being edited.

2. Press the **{ACTIVE PRGRAM}** softkey. 9/PC displays the first few blocks of the currently active program.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
F2	F3	F4	F5	F6	F7	F8
↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→
F2	F3	F4	F5	F6	F7	F8
↑	REFORM MEMORY	CHANGE DIR				→
F2	F3	F4	F5	F6	F7	F8

3. If the program selected is not the active program you wanted, press the **{DEC-ACT PRGRAM}** softkey. 9/PC deactivates the part program and return to the directory screen.

(softkey level 3)

↑	DE-ACT PRGRAM	SEARCH	MID ST PRGRAM			→
F2	F3	F4	F5	F6	F7	F8
↑	SEQ STOP				TIME PARTS	→
F2	F3	F4	F5	F6	F7	F8

Program Search {SEARCH}

Use the Program Search feature to begin program execution from some block other than 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.

9/PC can start a program at a chosen block and establishing any previous G-, M-, S-, F-, and T-words, work coordinate offsets, etc. that were established in previous blocks using the search with memory feature. For details, refer to page 7-10.

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

To perform a program search operation:

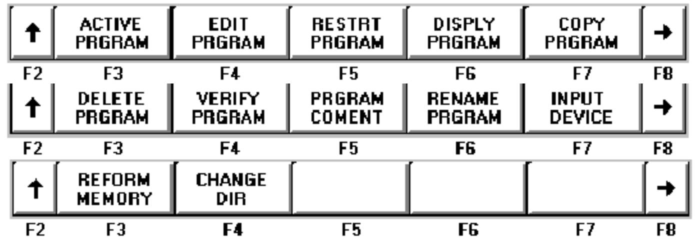
1. Press the **{PRGRAM MANAGE}** softkey. The program to search must have been previously selected for automatic execution as described in page 7-5.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
F2	F3	F4	F5	F6	F7	F8
		ERROR MESAGE	PASS-WORD	SWITCH LANG		→
F2	F3	F4	F5	F6	F7	F8

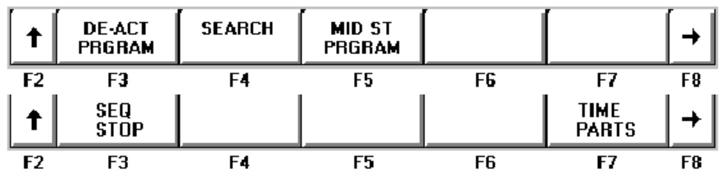
2. Press the **{ACTIVE PRGRAM}** softkey.

`(softkey level 2)`



3. Press the **{SEARCH}** softkey.

`(softkey level 3)`



4. Choose from the six search options:

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}**

5. When you use 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}**
exit, when the desired block is found	**{EXIT}**

Important: When performing an EOB search, the search is executed from the beginning of the part program, NOT from the point of display.

When you use the N search, O search, or STRING search features, first key in the desired N number, O number, or character string you want to search for. After it has been keyed in:

If you want to:	Press this softkey:
start the search	**[ENTER]**
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}**
exit when the desired block is found.	**{EXIT}**

If no number is keyed in for an N or O search, 9/PC simply searches for the next N- or O-word in the program.

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 cutting tool at a location that allows this block to execute without damaging the workpiece or cutting tool. This can be done through a manual operation or through MDI.

Search with Recall {MID ST PROGRAM}

Use the Mid-Start Program feature to begin program execution from some block other than the first block of the program. This feature will scan the program as it searches and from within the search area:

- send to logic the last programmed modal G–codes from each modal group.
- send to logic the last programmed modal M–codes from each modal group and set its associated logic strobe (nonmodal codes including user–defined M–codes are not sent to logic).
- send to logic the last programmed T–code and set its associated logic strobe
- send to logic the last programmed auxiliary function code (B–word) and set its associated logic strobe
- send to logic the last programmed spindle commanded speed and set its associated logic strobe
- resolve paramacro equations and assign paramacro variable values

  Important: Note on dual process systems (not available in release 1 of 9/PC), shared paramacro variables can be different than expected depending on the state of the part program in the other process. Equations that use logic paramacro variables may also evaluate differently since no paramacro interaction with logic occurs during a search operation.

- establishes any work coordinate system, including all offsets and rotations to the work coordinate system.

Important: Incremental moves that occur during a program search with recall operation, are always referenced from the last known absolute position in the part program. If no absolute position is specified in the searched part program blocks, 9/PC will use the current axis position as the start point for incremental moves.

When a search with recall is performed, 9/PC finds a character string or sequence number in a specific block for execution to begin from. Note that execution always begins from the beginning of the block, regardless of the location in the block of the searched string or sequence number. This searched block must be a block that would normally be executed during the full programs execution (a block that would be skipped by some means such as a jump, etc., cannot be searched for).

The program search with recall feature may be used to search into any subprograms or paramacros that may be contained in the main program. This is provided of course, that the searched block is in the path of normal program execution.

Important: The search with recall feature will not:

- send logic nonmodal M-codes including user-defined groups 0 - 3, group 4, group 5, and group 6 M-codes.
- read from or write paramacro variables to logic

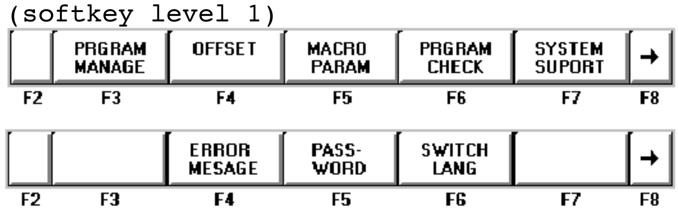
Important: This feature will not search into any cycle that calls a set of profile blocks (typically specified with the P- and Q-word in the cycle). Refer to the description of your cycle for details on profile blocks.

- send to logic gear change requests based on spindle speed

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

1. Press the **{PRGRAM MANAGE}** softkey.

(softkey level 1)



2. Press the **{ACTIVE PRGRAM}** softkey.

(softkey level 2)

↑	ACTIVE PRGRAM	EDIT PRGRAM	RESTRT PRGRAM	DISPLY PRGRAM	COPY PRGRAM	→
F2	F3	F4	F5	F6	F7	F8
↑	DELETE PRGRAM	VERIFY PRGRAM	PRGRAM COMENT	RENAME PRGRAM	INPUT DEVICE	→
F2	F3	F4	F5	F6	F7	F8
↑	REFORM MEMORY	CHANGE DIR				→
F2	F3	F4	F5	F6	F7	F8

Make sure that the program to search is the currently active program. If it is not, select it for automatic execution as discussed on page 7-5.

3. Press the **{MID ST PRGRAM}** softkey.

(softkey level 3)

↑	DE-ACT PRGRAM	SEARCH	MID ST PRGRAM			→
F2	F3	F4	F5	F6	F7	F8
↑	SEQ STOP				TIME PARTS	→
F2	F3	F4	F5	F6	F7	F8

4. To search for a sequence number press the **{SEQ # SEARCH}** softkey. To search for a character string press the **{STRING SEARCH}** softkey.

(softkey level 4)

↑	SEQ # SEARCH	STRING SEARCH				
F2	F3	F4	F5	F6	F7	F8

5. Key in the desired character string or sequence number to search for and press the **[ENTER]** key. 9/PC puts an @ symbol to the left of the block immediately before the block that automatic execution begins from.

   If this is not the block to begin execution from press either the:

   {CONT} softkey to continue to search for the entered character string or sequence number.

   {TOP OF PRGRAM} to return to the first block in the program.

   {QUIT} softkey to end either the sequence number search or the character string search operations.

(softkey level 5)

	CONT	TOP OF PRGRAM	QUIT	EXIT	EXIT & MOVE	
F2	F3	F4	F5	F6	F7	F8

ATTENTION: When you exit a mid-program start (search with memory), M- and S-codes are sent to logic. If, during normal execution, that program activated a spindle, mid-program start may also start it. To prevent this, your system installer can use the BR_BLKSTAT flag , which indicates 9/PC is in mid-program start.

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

{EXIT} - Use this softkey if the tool is at the exact location for execution of the searched program block. While 9/PC searches for your starting block it performs calculations to determine what the absolute position of the axes should be before your selected block is executed. If the cutting tool is not at this position when you press the **{EXIT}** softkey, 9/PC aborts the mid-start operation. When this occurs 9/PC displays the message "AXIS POSITION INCORRECT".

{EXIT & MOVE} - Use this softkey if the tool is not at the exact location for execution of the searched block. Be aware that the absolute position of the axes necessary at the start of the searched block is dependant on the previous blocks. There can be offsets activated or incremental moves that can make it difficult for you to determine the exact absolute starting point for the axes. 9/PC generates a motion block to place the tool at the position necessary to generate the intended contour when the searched block is executed. The block generated is always a linear move with a feedrate based on the last motion block prior to the searched block. If the last motion block was a cutting move with a feedrate, then the generated block will be a linear move at that cutting feedrate. If the last motion block was a rapid move, that the generated block will be a linear move at the rapid feedrate.

ATTENTION: It is the responsibility of the system installer's logic application to make sure proper activation of all necessary machine functions has occurred before allowing a search with recall operation to continue. You should verify that all machine functions are in the correct state before pressing **<CYCLE START>**.

A message is generated telling the operator to check that all generated modal codes are correct. This message reads "WARNING - VERIFY MODAL CODES". These modal codes should be checked on the G- or M-code status screen.

9/PC begins program execution from the selected block when you press the **<CYCLE START>** button. If you have pressed the **{EXIT & MOVE}** button 9/PC first executes the generated block to place the tool at the proper location. If you do not want 9/PC 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.

Basic Program Execution

After a program is written or loaded into 9/PC, it should be thoroughly tested before a part is mounted and machined. 9/PC offers three distinct testing modes in addition to fully automatic operation.

These modes are briefly described below in the order in which they would normally be implemented.

- QuickCheck™ (refer to page 7-15) — This mode is a basic syntax checker for a part program. It checks that proper format and syntax has been followed. No actual axis motion is produced during QuickCheck, however, offsets and coordinate system shifts are performed.
- Axis Inhibit (refer to page 7-17) -- 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.
- Dry Run (refer to page 7-18) -- Dry run simply replaces all F-word feedrates in a program with a special feedrate determined by the system installer in AMP.
- Part Production/Automatic (refer to page 7-19) -- In automatic mode all of the axes are active and all of the programmed feedrates are in effect.

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

When you see this to the left of the block:	9/PC:
*	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. Execution can be resumed at the interrupted location by pressing the **<CYCLE START>** button.

(1) Pressing <CYCLE STOP>

When you press the **<CYCLE STOP>** button, motion of the cutting tool decelerates and stops, and 9/PC stops automatic operation. If you press the **<CYCLE STOP>** button during a dwell, the dwell is interrupted and any remaining time/revolutions for the dwell are stored for later execution.

(2) Execution of an M00 or M01 in a Part Program

Execution of:	Description:
M00	9/PC stops automatic operation after it executes the remaining commands in the M00 block.
M01	if the OPTIONAL STOP condition is set to ON, 9/PC 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 9/PC continues executing the part program as normal. The optional stop condition may be turned off or on using a switch installed by your system installer.

(3) Entering a Sequence Stop Number

To interrupt execution at a specific block in the part program, use the sequence stop feature described on page 7-2. 9/PC stops automatic operation after it completes the commands in the designated block.

(4) Feedhold Status

Your system installer may have written logic to allow the activation of a feedhold state through the use of a button or switch. When activated 9/PC decelerates all moving axes to a feedrate of zero until the feedhold state is deactivated. For details on using feedhold, refer to documentation provided by your system installer.

QuickCheck

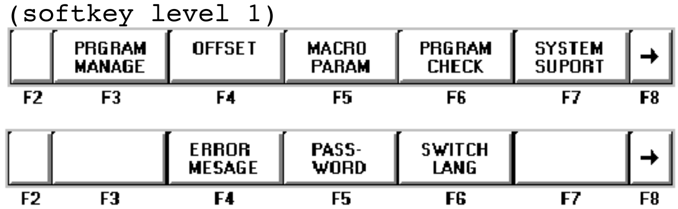
QuickCheck is a basic syntax checker for a part program. It checks that proper format and syntax have been followed during programming. No actual axis motion is produced in QuickCheck mode.

To use the QuickCheck feature, follow these steps.

1. Select a program to check as described on page 7-5 and return to softkey level 1.

2. Press the **{PRGRAM CHECK}** softkey.

(softkey level 1)

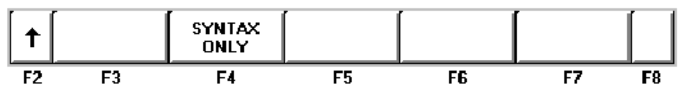


3. Press the **{QUICK CHECK}** softkey.

(softkey level 2)

4. Press the **{SYNTAX ONLY}** softkey.

(softkey level 3)



   When you press the **{SYNTAX ONLY}** softkey, it appears in reverse video.

5. Press the **<CYCLE START>** button.

   When you press the **<CYCLE START>** button, the program blocks are executed.

If a program block contains an error, the program check stops and 9/PC displays the message “ERROR FOUND.”

To continue checking the remaining program blocks, press the **<CYCLE START>** button again. If no more errors are found, 9/PC displays the message “COMPLETED WITH ERROR(S)” and the part program is automatically deactivated to allow editing.

If 9/PC finds no errors during QuickCheck the program screen displays the message “COMPLETED WITH NO ERRORS”. 9/PC then automatically resets the program to the first block.

To disable QuickCheck, press the **{STOP CHECK}** softkey.

ATTENTION: Note that when a program is run during quick check mode, 9/PC 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.

Axis Inhibit Mode

When you activate AXIS INHIBIT, 9/PC can execute a part program without moving specified axes. 9/PC simulates axis motion by updating the axis location and feedrate displays, using the commanded feedrates, acceleration, and deceleration.

The program is executed in approximately the same amount of time as it would be in automatic mode, even though some or all axes may not move. You can use the axis inhibit feature in conjunction with Dry Run.

ATTENTION: When testing a program using Axis Inhibit 9/PC 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. Refer to page 7-1.

You can activate AXIS INHIBIT to inhibit motion of any or all of the axes depending on the configuration determined by your system installer. This includes jogging moves. When axis motion has been inhibited for a single axis, the remaining axes still execute normally and the axis location display is updated as if axis motion were 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 using a switch installed by your system installer (refer to the documentation provided by the system installer). 9/PC must be in cycle stop or E-Stop to activate or deactivate the Axis Inhibit feature. Any attempt to activate or deactivate the feature during program execution or when in cycle suspend or feedhold states is ignored. Attempts to activate the Axis Inhibit feature during jogging are also ignored.

Press **<CYCLE START>** to program execution with the Axis Inhibit feature. Make sure you select a program for execution. Refer to page 7-5.

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.

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

The spindle motion may also be inhibited by using a switch installed by your system installer. Refer to the documentation provided by your system installer.

Dry Run Mode

The Dry Run function permits the checking of a part program to make sure that machine motions are correct. It is intended to be executed without the material or part mounted. The dry run function replaces all programmed feedrates with the maximum cutting feedrate. Jogging moves and moves that are programmed using rapid traverse (G00) are not affected by dry run.

The Axis Inhibit feature can be used in conjunction with Dry Run if desired.

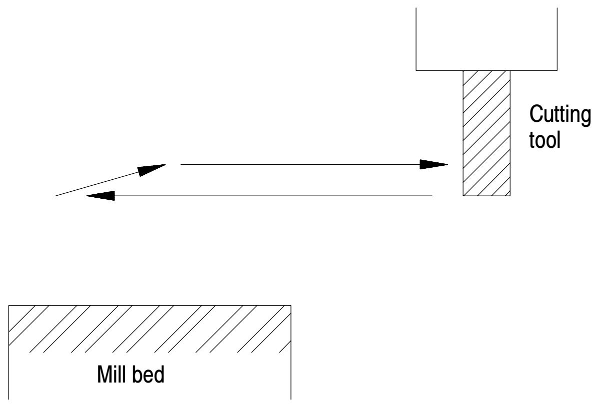
If you use the external decel feature simultaneously with the Dry Run feature, the feedrates that are assigned to External decel feature are used and the Dry run request is ignored.

You can use the **<FEEDRATE OVERRIDE>** to modify the cutting feedrate. Your system installer determines in AMP if rapid feedrates are overrides by the **<RAPID FEEDRATE OVERRIDE>** switch/button or the **<FEEDRATE OVERRIDE>** switch during Dry Run.

ATTENTION: When testing a program using Dry Run, 9/PC 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. Refer to page 7-1.

ATTENTION: Your system installer can write logic 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 machine, part, or injury to the operator.

Figure 7.3
Dry Run



The Dry Run feature can be activated using a switch installed by your system installer (refer to the documentation provided by your system installer).

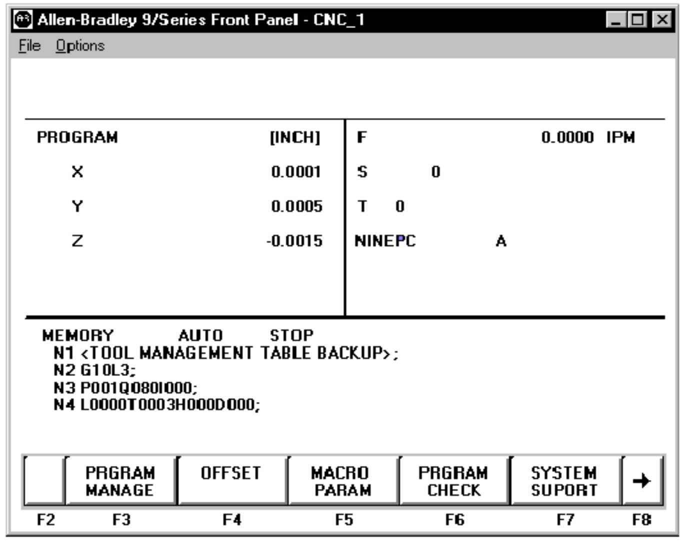
Part Production/Automatic Mode

Automatic mode is the normal operating mode of 9/PC. A program that is run in the automatic mode is executed with all of the axes active and all of the programmed feedrates active.

To select the automatic mode, place the **<MODE SELECT>** switch/button (on the MTB panel) in the AUTO position.

Automatic mode is the default mode whenever AUTO appears on the Main Menu screen, and it is always active unless one of the program checking modes has been selected.

Figure 7.4
Main Menu Screen in AUTO Mode

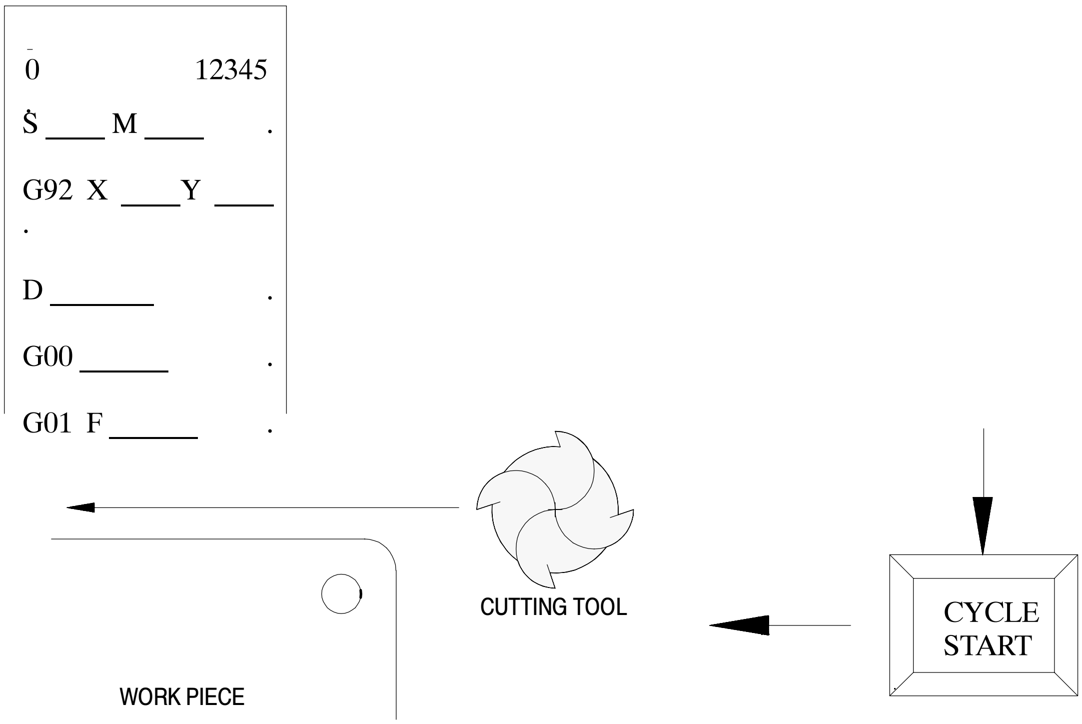


In automatic mode, 9/PC manages machine operations according to the commands in a part program.

Command:	Process:
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 workspace is clear and all safety features are intact before pressing **<CYCLE START>**.

Figure 7.5
Automatic Mode



Execution of a part program continues until 9/PC encounters an M02 or M30. If 9/PC 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 7-5 or by pressing the **<EMERGENCY STOP>** button.

Programmable Block Execution

Use the Programmable Synchronous/Asynchronous Block Execution feature to set the mode that determines how your part program and the machine's logic are synchronized.

There are three types of programmable block execution:

- Synchronous — G60
- Asynchronous — G60.1
- Autosynchronous — G60.2

Important: Each of these G-codes must be programmed in a nonmotion blocks. Blocks that contain these G-codes are executed in Synchronous mode.

Synchronous Mode (G60)

Use the “Synchronous Mode” to force at least one logic scan to occur for the start of each program block. This mode causes the 9/PC to activate the next sequential program block only after the control triggers the machine logic to run. Execution of this G-code happens at a time interval set by your system installer in AMP.

Synchronous mode allows the following to occur on the active program block:

- logic to decode any M-code, T-word, or any other auxiliary words in the block
- logic to hold the block in pre- or postblock state

Important: In G60 mode, the time interval for interpolation is the Coarse Scan Time.

Important: If your system is executing very short motion blocks while in Synchronous Mode, you may experience delays between blocks waiting for the logic to complete.

Asynchronous Mode (G60.1)

Use “Asynchronous Mode” to allow multiple blocks to be activated during a single scan of logic. This mode causes the 9/PC to execute:

- motion blocks at the fastest rate possible (Fine Foreground Scan Time)
- nonmotion blocks at the fastest rate possible, allowing multiple, contiguous nonmotion blocks to be executed within one system scan time

Asynchronous Mode may be used if your part program contains blocks that execute more quickly than the programmed system logic scan time, but within the 9/PC Fine Foreground Scan Time.

Important: In G60.1 mode, the time interval for intepolation is the Fine Foreground Scan Time.

This mode permits very short duration blocks to run without the delay of waiting for the logic to complete. It is important to note that certain types of blocks cannot be exectued in Asynchronous Mode. Refer to the following table for limitations in this mode:

This action:	Causes this 9/PC condition:
Programming auxiliary functions (S, T, M)	a decode error and cycle stop
Programming G-codes in groups 9, 10, 17, 21, 23	
Programming CSS or any fixed cycles	
Execution of a threading block	
Solid tapping	
Execution of any time-dependent paramacro calculations[1]	
Logic request for postblock, preblock, or transfer inhibit	

[1]Time-dependent paracros include logic input flags, sytem clock, coordinates of commanded position, machine coordinate position, skip signal position work coordinate position, skip signal position machine coordinate position, and current following error, For more information regarding these paramacros, refer to chapter 27, *Paramacros*.

Important: When the logic program requests transfer-inhibit and postblock conditions in asynchronous mode preblock, the control is forced into a CYCLE STOP state and returns an error.

ATTENTION: When in asynchronous mode, the CNC flag status is not updated with the execution of each part program block.

Autosynchronous Mode (G60.2)

Use “Autosynchronous Mode” to allow the control to switch between synchronous and asynchronous block execution, depending upon the contents of the part program block.

In Autosynchronous Mode, it is not necessary for the part program to change the execution mode. The control handles all part program blocks by automatically selecting either synchronous or asynchronous mode. The control makes this selection based upon the contents of the block. Refer to the following table for limitations in this mode:

This action	Causes this 9/PC condition
Programing auxilliary functions (S, T, M)	execution of the block in synchronous mode
Programming G-codes in groups 9, 10, 17, 21, 23	
Programming CSS or any fixed cycles	
Execution of a threading block	
Solid tapping	
Execution of any time-dependent paramacro calculations[1]	
Execution of all **motion-only** blocks	execution of the block in asynchronous mode
Logic request for postblock, preblock, or transfer inhibit	a decode error and cycle stop

[1]Time-dependent paracros include logic input flags, sytem clock, coordinates of commanded position, machine coordinate position, skip signal position work coordinate position, skip signal position machine coordinate position, and current following error, For more information regarding these paramacros, refer to chapter 27, *Paramacros*.

Important: Certain logic requests require a CYCLE START if Autosynchronous Mode is active.

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 scans the program as it searches for the interrupted block and from within the search area:

- send to logic the last programmed modal G-codes from each modal group.
- send to logic the last programmed modal M-codes from each modal group and set its associated logic strobe (nonmodal codes including user-defined M-codes are not sent to logic).
- send to logic the last programmed T-code and set its associated logic strobe
- send to logic the last programmed auxiliary function code (B-word) and set its associated logic strobe
- send to logic the last programmed spindle commanded speed and set its associated logic strobe
- resolve paramacro equations and assign paramacro variable values (note equations that use logic paramacro variables may evaluate differently since no paramacro interaction with logic occurs during a search operation).
- establishes any work coordinate system, including all offsets and rotations to the work coordinate system.

Important: Incremental moves that occur during a interrupted program recover operation, are always referenced from the last known absolute position in the part program. If no absolute position is specified in the searched part program blocks, 9/PC will use the current axis position as the start point for incremental moves.

Unless **Cutter Compensation** is active, when a program recover is performed, 9/PC automatically returns the program to the beginning of the block that was interrupted. In the case of power failure, 9/PC will even reselect the program that was active prior to the interruption.

When a program recover is performed 9/PC automatically returns the program to the beginning of the block that was interrupted. In the case of power fail 9/PC will even re-select the program as active.

ATTENTION: When a program recover is performed 9/PC automatically returns the program to the beginning of the block that was originally interrupted. The beginning of the block is probably not the point that axis motion was interrupted. For absolute linear moves this causes no problem if the tool is still somewhere along the path of the block that program execution was interrupted while cutting. In incremental or circular mode however, if the cutting tool is still located at the point that program execution was interrupted a restart may damage the part. If a program recover operation is performed in incremental mode it is important that the cutting tool be at the location that the interrupted program block began, not the location that the program was interrupted at.

This feature may also be used to search into any subprogram or paramacro that may be contained in the main program.

ATTENTION: It is the responsibility of the system installers logic application to make sure proper activation of all necessary machine functions has occurred before allowing a interrupted program to continue. You should verify that all machine functions are in the correct state before pressing **<CYCLE START>**.

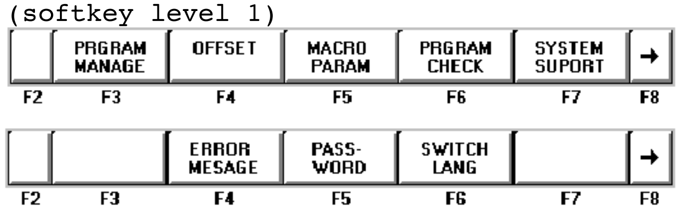
Important: The interrupted program recover feature will not:

- send logic nonmodal M-codes including user-defined groups 0 - 3, group 4, group 5, and group 6 M-codes.
- read from or write paramacro variables to logic
- send to logic gear change requests based on spindle speed

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

1. Press the **{PRGRAM MANAGE}** softkey.

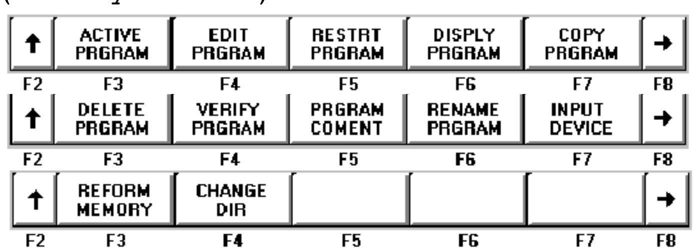
(softkey level 1)



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. 9/PC automatically re-selects the interrupted program if it was disabled by 9/PC when power was lost.

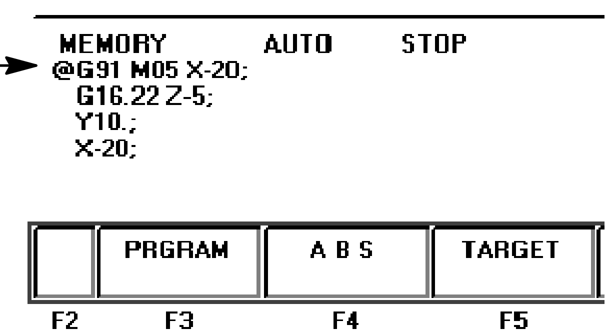
(softkey level 2)



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

9/PC will locate an @ symbol to the left of the block immediately before the block that automatic execution was interrupted at.

This is where execution was interrupted.



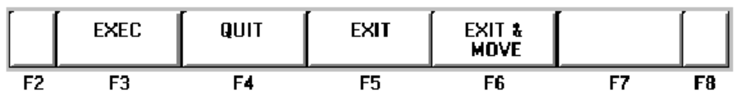
If this is not the block to begin execution from, press the **{QUIT}** softkey. The program restore feature will be aborted.

ATTENTION: When you exit a program restart operation (search with memory), M- and S-codes are sent to logic. If, during normal execution, that program activated a spindle, mid-program start may also start it.

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

   Press the **{EXIT & MOVE}** softkey if you wish to exit this particular program and search another program.

(softkey level 3)



When the **{CYCLE START}** button is pressed 9/PC resumes program execution from the block selected with the program restart feature.

Jog Retract

Use the jog retract feature to allow for inspection or change of the cutting tool during automatic program execution. It allows the cutting tool to be jogged from the workpiece in multiple steps, and then returned to the workpiece automatically by having 9/PC retrace the jogging steps that were used.

9/PC 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. 9/PC returns the tool along the jog retract path at a feedrate specified in AMP.

Important: If the same axis is used in succession during a jog retract operation, 9/PC assumes that only one jog retract move has been executed on that axis.

Only simple single axes jog moves can be performed during the jog retract function. You cannot perform multiple axis jogs, and jogging offset.

Tool offsets can be changed at any time during jog retract. Refer to chapter 3. 9/PC does not make these offsets active until the execution of the first block after the tool has been returned from jog retract.

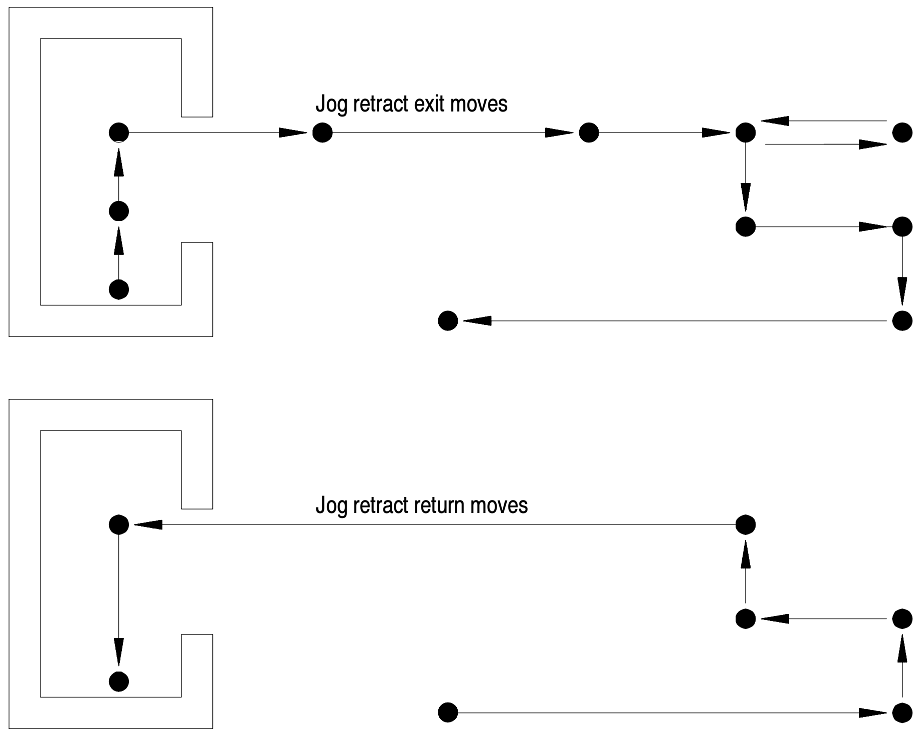
ATTENTION: If the Jog Retract function is deactivated during its execution (performing a control reset, E-Stop, etc.), attempting to return the tool by pressing **<CYCLE START>** can cause the Jog Retract function to abort. The program returns to the start point of jog retract along a linear path. In the event that Jog Retract is deactivated during execution, we recommend that the cutting tool be jogged to the point from which jog retract was started prior to pressing **<CYCLE START>** to avoid possible part or tool damage.

To perform a jog retract operation:

1. Press the **<CYCLE STOP>** or activate **<SINGLE BLOCK>** feature button to stop program execution.
2. Press the **<JOG RETRACT>** button. The light inside the button turns on to indicate that the function is active.
3. Move the cutting tool from the workpiece using either continuous jog, or incremental jog operations (refer to page 4-1 for jogging information.)
4. Inspect and change the tool or tool offset as desired. Details on how to do this are in chapter 3.
5. After completing the desired inspection or tool change, press the **<CYCLE START>** button. Any tool offset changes you have made become active when the cycle start is requested. The tool returns to the location where jog retract began, following the same path used when you jogged the tool away from the work piece (+ or – any new tool offset values).

   You can press **<CYCLE STOP>** during the tools automatic return to the jog retract start position. When this is done, the tool can be retracted from this point using jog moves and 9/PC adds these moves to any remaining jog retract steps that have not yet been returned.
6. Once the cutting tool is fully returned from a jog retract operation, 9/PC continues on in the part program unless in single block mode. If in single block mode, 9/PC 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, 9/PC only recognized six jog moves upon returning instead of the actual 11 moves that were made to retract the tool. This is because the jog retract feature records consecutive jog moves on the same axis as one move.

ATTENTION: If the number of jog retract moves performed exceeds the maximum allowed number set in AMP, 9/PC moves the cutting tool directly from the final point of jog retract to the last remembered jog retract point along a straight line when you press **<CYCLE START>**. The tool is then returned in the normal jog retract fashion.

Figure 7.7
Jog Retract Moves that Exceed the Maximum Allowed in AMP

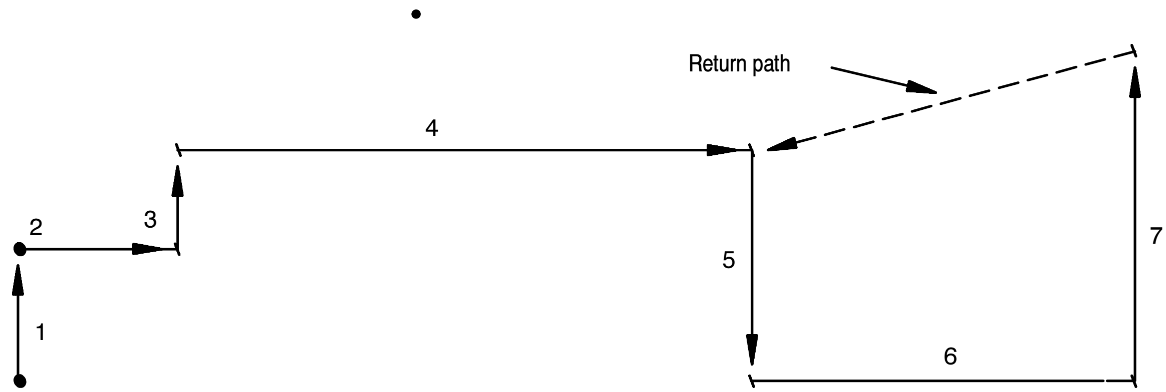


Figure 7.7 emphasizes the possible problems that can result from exceeding the maximum allowed jog retract moves. In this example, the number of allowed moves set in AMP is four.

When you press the cycle start button at the end of the 7th jog move, 9/PC 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, 9/PC continues the jog retract return operation as normal.

If the jogging moves of 5, 6, and 7 were intended to avoid a protrusion during the jog retract, a collision could result when returning the tool.

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 machine application. This is because the number of allowable Block Retraces directly affects 9/PC's block look ahead operation.

This function can only be enabled when 9/PC is in cycle stop or cycle suspend state, and it is ignored if 9/PC has already executed an M02 or M30 end of program.

To perform a block retrace operation:

1. Press the **<CYCLE STOP>** or activate the **<SINGLE BLOCK>** feature button to stop program execution.
2. Press the **<BLOCK RETRACE>** button.

After you press the **<BLOCK RETRACE>** button, 9/PC retraces the block that was being executed when the cycle stop occurred or retraces 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 still appears 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, 9/PC re-executes the portion of the block that has been executed. For example, if the block requests an axis move of 20 millimeters and the axis has moved 12 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 though this may be modified by the use of the **<FEEDRATE OVERRIDE>** switch. Refer to chapter 17.

Press the **<CYCLE START>** button at any time during a block retrace to return the cutting tool to normal forward execution. Program execution returns to the normal forward direction from the currently retraced block. 9/PC executes the retraced blocks in normal order until the tool is positioned at the start point of block retrace. From this point it continues program execution in a normal fashion unless **<SINGLE BLOCK>** is active. If **<SINGLE BLOCK>** is active, 9/PC halts execution when the return from block retract is complete.

While block retrace is active, 9/PC disables all jog features with the exception of **<JOG RETRACT>**. Refer to page 7-14. MDI is not available to insert blocks during a block retrace operation.

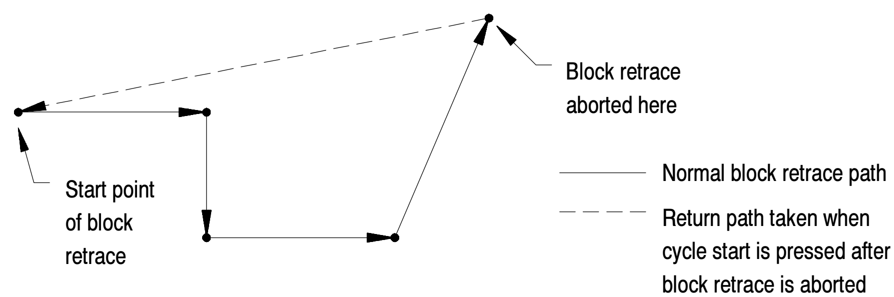
The block retrace function is unable to retrace any of these blocks and an attempt to do so results in an error message:

- Threading
- Tapping
- Boring
- Inch/Metric changes (unit conversion)
- A block that commands a tool change operation

- A block that commands a change in the coordinate system
- Any block that is followed by a Manual Jog Move except a Jog Retract
- The number of blocks retraced is already equal to the maximum number of retraceable blocks as determined in AMP
- Certain Paramacro Parameter Assignments
- Interrupt Macros

ATTENTION: If the block retrace function is deactivated during its execution (performing a control reset, E-STOP, etc.), attempting to return the tool by pressing cycle start can cause an undesired return path. The tool returns to the start point of block retrace along a linear path. This is most likely not the retracted path. To avoid possible part or tool damage, we recommend that the cutting tool be jogged to the point from which block retrace was started prior to pressing cycle start.

Figure 7.8
Pressing Cycle Start When Retract Path is Lost



END OF CHAPTER

Data Display

Chapter Overview

This chapter gives a description of the different data displays available on the control.

Selection of Axis Position Data Display

Pressing the [F12] key displays the softkeys for selecting the axis position data screens.

The control provides eight different axes position data screens as described in Table 8.A. Four of these screens may be displayed in normal or large (4 axis triple size or 8 axis double size) 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 and large.
{ABS}	Axis position in the machine coordinate system is displayed. Each time this softkey is pressed the display toggles between normal and large.
{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 and large.
{DTG} Distance to go	Distance from the current position to the end point of the commanded axis move is displayed. Each time this softkey is pressed the display toggles between normal and large.
{AXIS SELECT}	This softkey is used to select which axes are going to be displayed on normal and large displays. Normal 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 available on systems with 6 or fewer axes.
{G CODE STATUS}	G-codes that are currently active are displayed.

The screens described above may also show in addition to axis position:

- The current unit system being used (mm or in.)
- 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 **[F12]** key to display the softkeys for selecting axis position data screens. Press the **[F12]** key at any time from any softkey level. Pressing the page **{→}** softkey displays additional selections.

2. Press the softkey corresponding to the display wanted. The softkeys will toggle between large and normal display mode each time the corresponding softkey is pressed, provided that screen is available as a large display.

   The “large” display is available only for the axis position screens (Program, Absolute, Target, and Distance to Go).

   For example, immediately after power up and accessing the **[F12]** 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 four axes in triple-height characters and any eight axes in double-height characters.

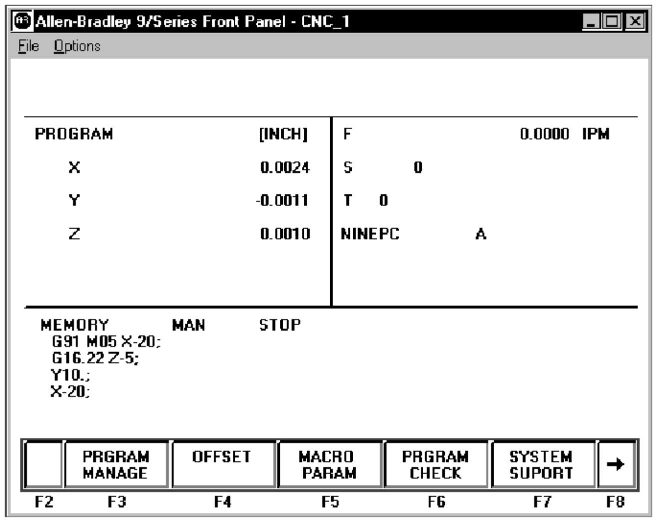
3. To return to softkey level 1, press the **[F12]** key again. The most recently selected data position screen will remain in effect for softkey level 1 until either power is turned off or a different position display screen is selected. The default screen selected at power up is the regular size program 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.

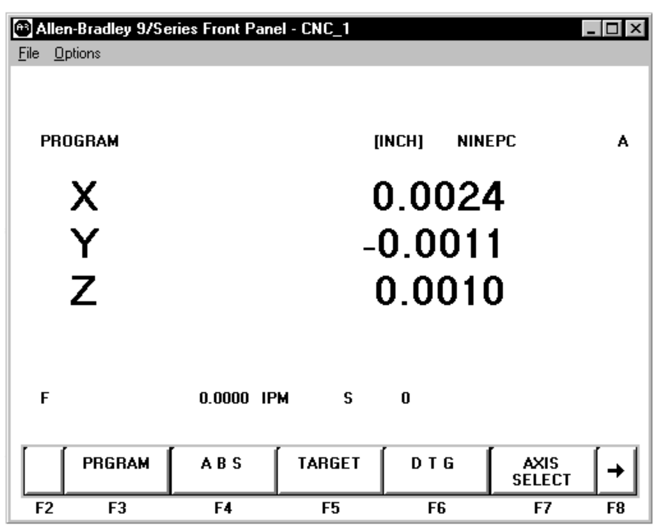
Figure 8.1
Result After Pressing {PRGRAM} Softkey



{PRGRAM} (Large Display)

Axis position in the current work coordinate system displayed in large characters.

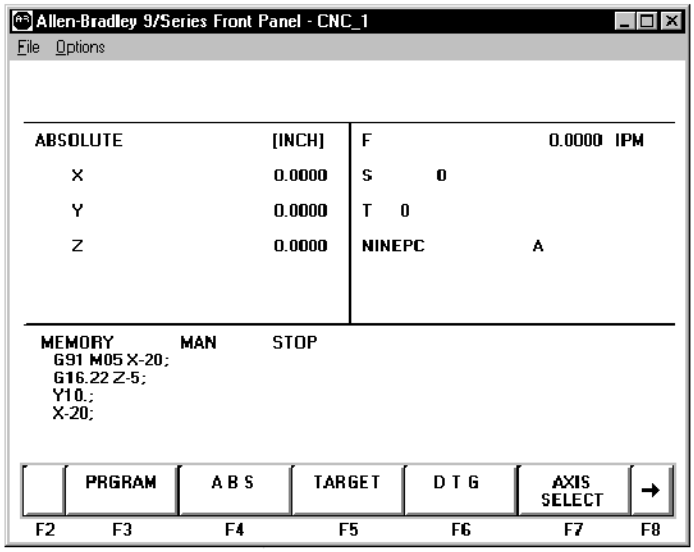
Figure 8.2
Results After Pressing {PRGRAM} (Large Display) Softkey



{ABS}

The axis position data in the machine coordinate system.

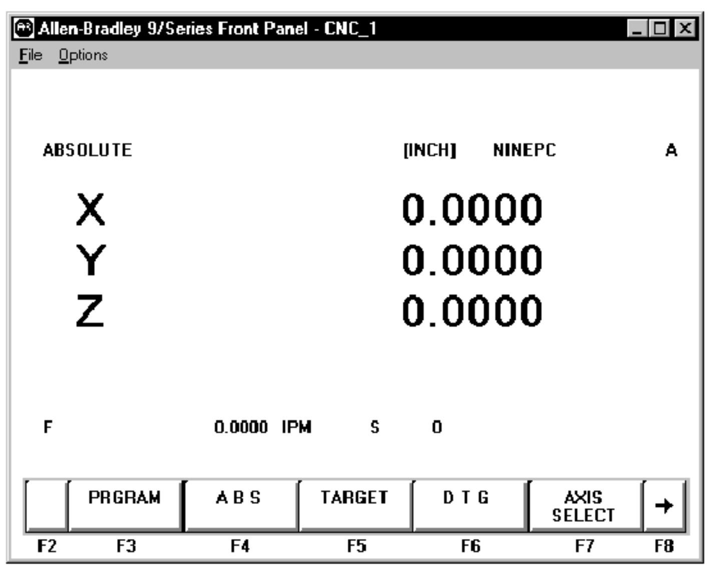
Figure 8.3
Results After Pressing {ABS} Softkey



{ABS} (Large Display)

Axis position in the machine coordinate system displayed in large characters.

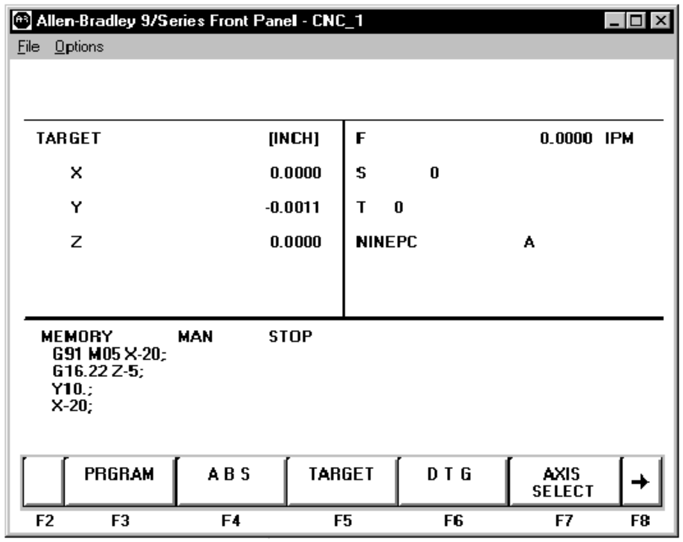
Figure 8.4
Results After Pressing {ABS} (Large Display) Softkey



{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.

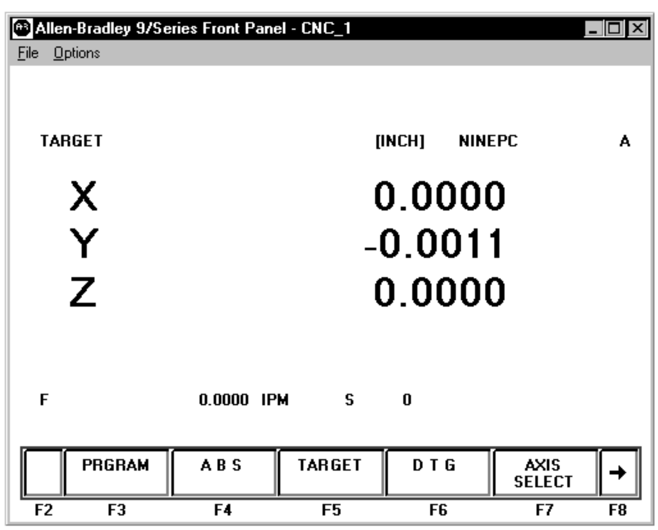
Figure 8.5
Results After Pressing {TARGET} Softkey



{Target} (Large Display)

The coordinate values in the current work coordinate system, of the end point of commanded axis moves in normal size characters.

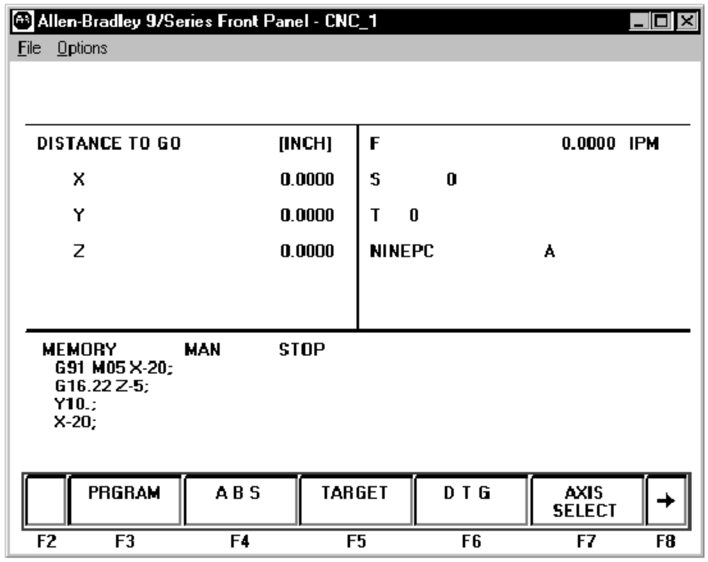
Figure 8.6
Results after Pressing {TARGET} Softkey (Large Display)



{DTG}

The distance from the current position to the command end point, of the commanded axis in normal size characters.

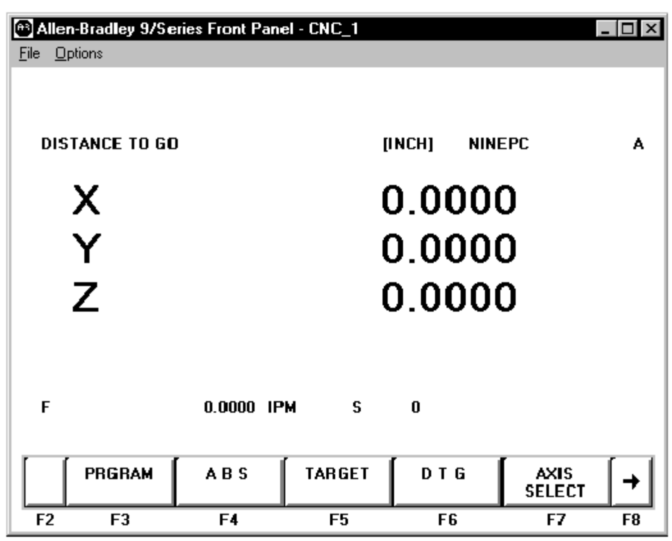
Figure 8.7
Results After Pressing {DTG} Softkey



{DTG} (Large Display)

The distance from current position to the command end point of the commanded axis move in large characters.

Figure 8.8
Results After Pressing {DTG} (Large Display) Softkey



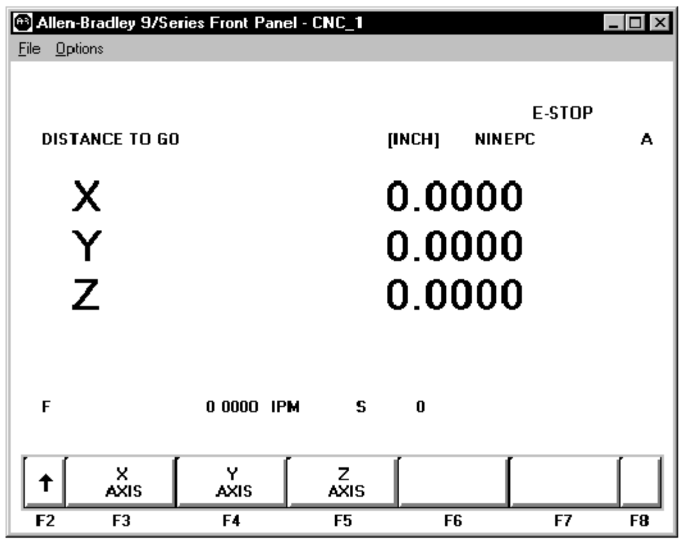
{AXIS SELECT}

Important: **{AXIS SELECT}** is available only during a large character 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 the axis select features can select the axes for these normal size displays.

Figure 8.9
Results After Pressing {AXIS SELECT} Softkey



{M CODE STATUS}

The currently active system M-codes are displayed. This screen indicates the last programmed system M-code in the modal group. It is the logic programmer's responsibility to make sure proper machine action takes place when the M-code is programmed.

To access this screen, press the **{→}** button.

(softkey level 1)

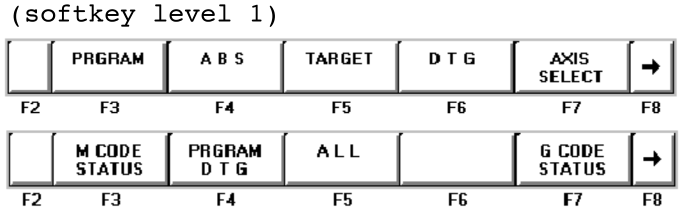
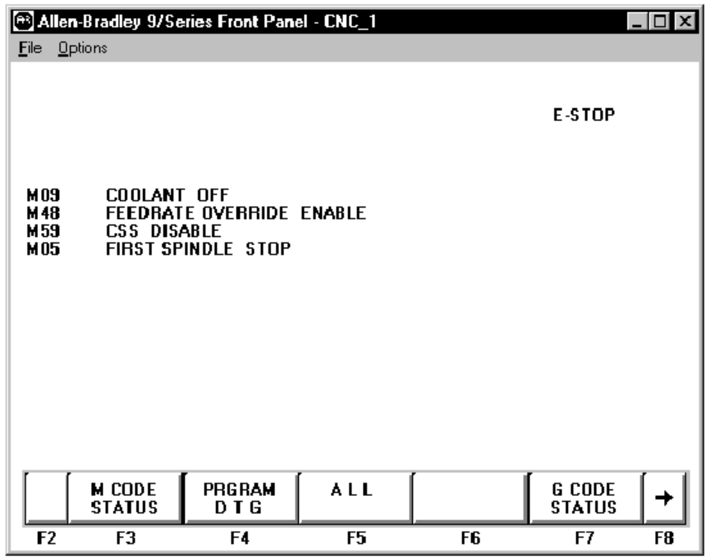


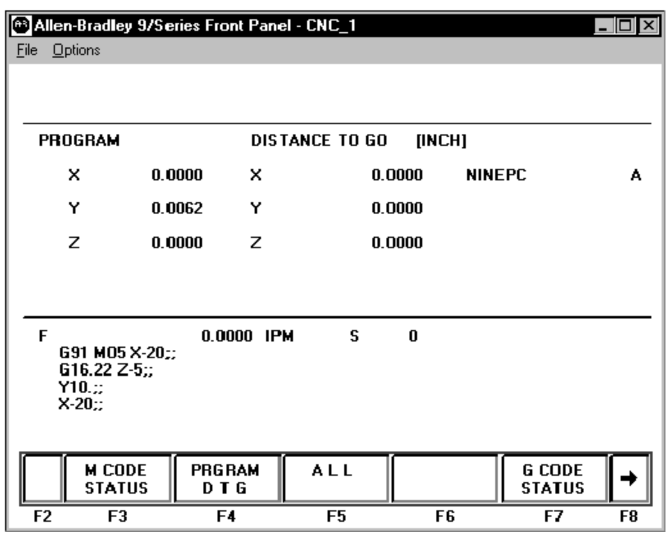
Figure 8.10
Result After Pressing {M CODE} Softkey



{PRGRAM DTG}

This screen provides a multiple display of position information from the program screen and the distance to go screen.

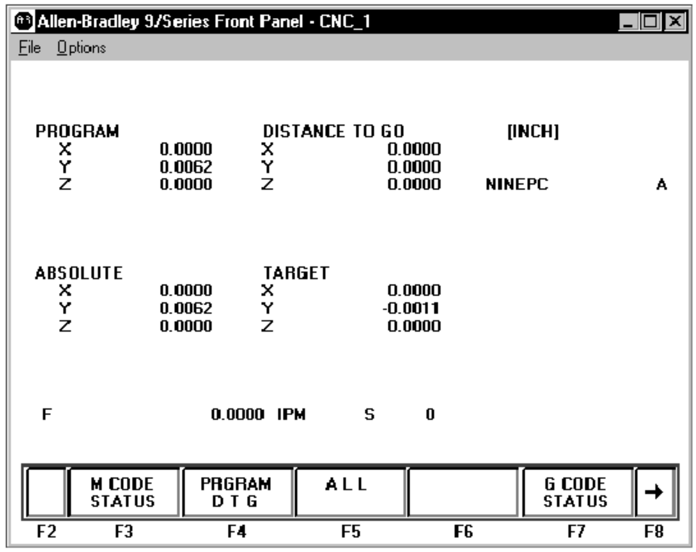
Figure 8.11
Program, Distance to Go Screen



{ALL}

This screen provides a multiple display of position information from the program, distance to go, absolute, and target screens.

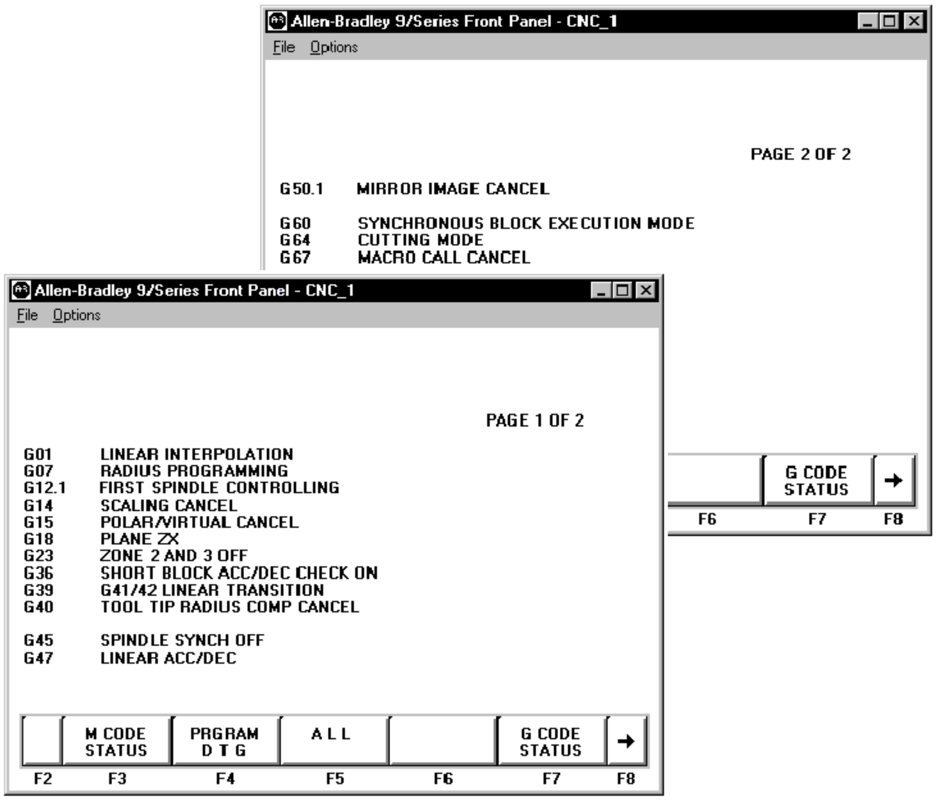
Figure 8.12
Result After Pressing {ALL} Softkey



{G CODE STATUS}

The currently active G-codes are displayed.

Figure 8.13
Results After Pressing {G CODE} Softkey

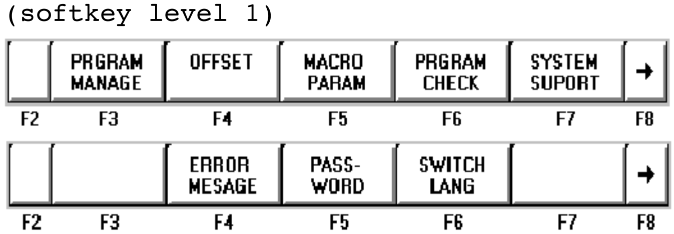


Changing Languages

The 9/PC control is equipped to display all screens, softkeys, and messages in multiple languages.

Press the **{SWITCH LANG}** softkey to access these languages.

(softkey level 1)



Each time you press the **{SWITCH LANG}** softkey, the language displayed on the screen changes. The system installer can password protect this softkey.

The 9/PC control is equipped to display five languages. The languages available and the order they are displayed are fixed in this order:

- English
- French
- German
- Spanish
- Italian

Power Turn-on Screen

When the 9/PC is started, the BDS displays the power turn-on screen.

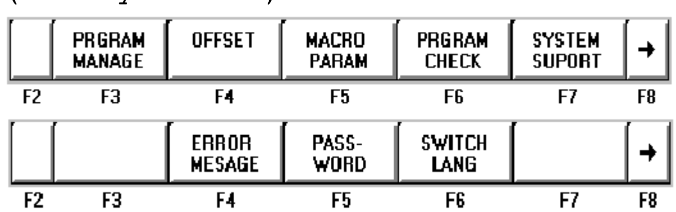
The following section discusses how to modify information displayed on this screen at power up.

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.

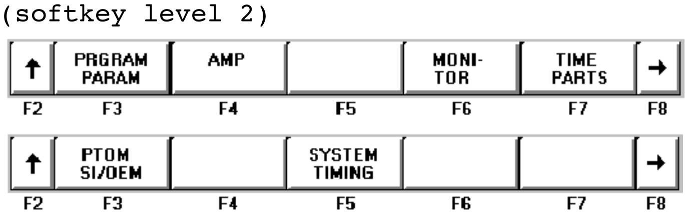
(softkey level 1)



The control brings you to softkey level 2.

2. Press the **`{PTOM SI/OEM}`** softkey.

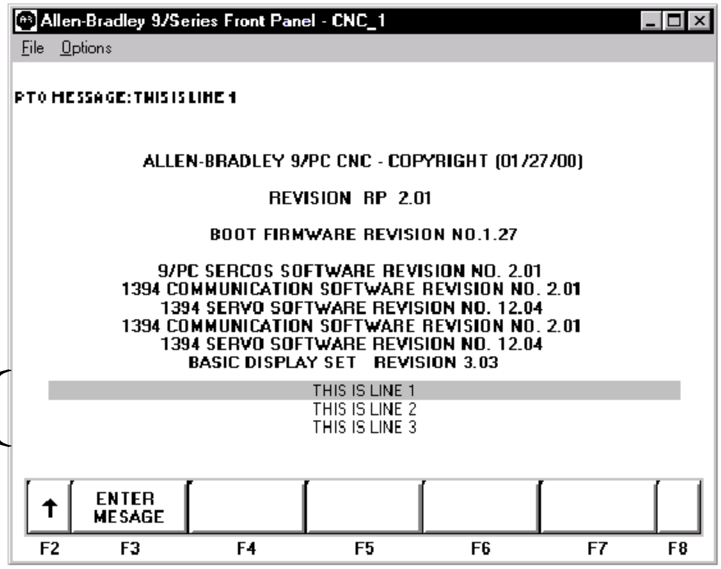
(softkey level 2)



The control changes the screen to display the PTO screen, as shown in this section.

There are three lines available for system integrator messages. The softkeys used to change these lines are **password protected.**

These three lines are available for system integrator messages.



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)

5. Once you edit the line, press the **[ENTER]** key to transfer the edited line to the PTO screen. After you press the **[ENTER]** key, you can either:

   - edit another line
   - exit the PTO screen by pressing the up arrow softkey

   Changes to the system integrator message lines are automatically saved when you exit the PTO screen.

END OF CHAPTER

Introduction to Programming

Chapter Overview

The 9/PC control performs machining operations by executing a series of commands that make up a part program. These commands are interpreted by the control which then directs axis motion, spindle rotation, tool selection, and other CNC functions.

Part programs can be executed from the control's memory or from the hard disk.

This chapter begins with an explanation of CNC part program format. The remainder of the chapter deals with the contents of a part program, including:

Program Configuration

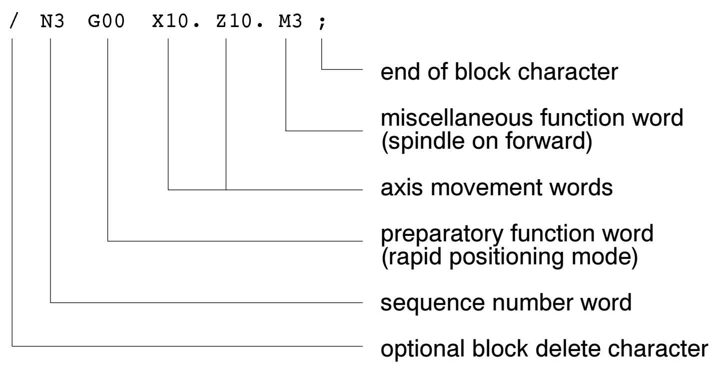
Each individual machining operation performed by the control is determined by the control's interpretation of a group of words or codes (commands) called a "block." Individual blocks in a part program define each machining process. Part programs consist of a number of blocks that define a complete operation of a part.

Part program blocks are made up of:

Component of program block:	Description:
character	a number, a letter, or a symbol that means something specific to the control. For example, 1 G ; are characters that the control recognizes as meaningful information.
address	a letter that defines the instruction for the control. Examples of addresses are: G, X, Z, F.
word	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 9-15.

code	industry standards for many of the G- and M-codes used here. For that reason, they are often referred to as G- or M- "codes."
parameter	a number of fixed cycles that are initiated by a specific G-code. Other words appearing in those G-code blocks are referred to as "parameters" because their values are relevant only to that G-code. For example, a Z-word generally refers to a Z axis move, but when it appears in a block with a G83 peck drilling cycle, its value refers to the depth of the hole to be drilled. In that case, it is a "parameter" of the G83 fixed cycle.

A block is a set of words and characters that defines the operations of the control. For example:



The 9/PC control sequentially executes blocks in a part program to conduct the required machining operation.

Important: To make jumps, loops, or calculations within an executing program or subprogram, use the paramacro features described in chapter 27.

A part program has a:

Part program section:	Description:
Beginning	sets up the control and the machine to perform the operations wanted
Middle	performs the machining operations
End	returns the machine to a safe stop position, and preparing the control for the next part program

The blocks programmed vary for each section of the program. As an example, consider the following simple example program.

Example 9.1
Sample Part Program

```
G91G21;          beginning

G00X28.;
G33Z-64.E4.;     middle
G00X5;

Z2.;             end
M02;
```

Program Names

You can enter up to 8 alphanumeric characters for program names.

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. The control does **not** consider programs with more than 5 numeric characters as subprograms.

The control lists subprograms in numerical order from lowest to highest. The main programs are listed in alphabetical order, following the subprograms.

Entering Program Names

To enter a program name:

1. Press the softkey **{PROGRAM MANAGE}**. This calls up the program directory, which lists subprograms first, then programs by alphabetical order.
2. Type in the name of a new program or one already listed. You cannot enter spaces or special characters.
3. Press **{EDIT}**. This initiates the editing mode for the program selected.

Example 9.2
Entering Subprogram Names

Name entered	Program name stored by control
O00123	O00123
O123	O00123
123	O00123
12345	O12345

Example 9.3
Legal Program Name Blocks

```
O12345;
O12345(TAPPING PROGRAM);
O333
O2;
```

Sequence Numbers

Each block in a part program can be assigned a sequence number to distinguish one block from another. Sequence numbers begin with an N address, followed by a one to five digit numeric value.

Sequence numbers can be assigned at random to specific blocks or to all blocks. If you assign sequence numbers to blocks, you can designate their sequence numbers. Sequence numbers are necessary to make program jumps and to specify a block for paramacro calls and returns.

Below is an example of two blocks with sequence numbers 10000 and 10010.

```
N10000 X5.  Z4.  ;
N10010 X2.  Z2.  ;
```

When you assign sequence numbers to blocks the N-word comes first in the block except when you designate block delete. See page 9-5. It is not necessary to program the N-word first in the block. The control still finds it for jumps; however, it will not find renumber operations.

If more than one N-word is in a block, the control uses only the first N-word encountered for that block number.

Different blocks can be assigned the same sequence number. If this number is called by a "GOTO" or some other command, the first block found by the control with the sequence number that is closest to the calling block is used. The control first searches for the sequence number in the forward direction (from the calling block), 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.

Comment Blocks

Information between the control out code "(" and the control in code ")" within a part program is regarded as a comment, and it is 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 9.4
Program Block With Comment

```
N00010G91X5.(CHANGE TO INC.  MOVE X 5);
```

Block Delete and Multilevel Delete

When you program a slash "/" followed by a numeric value (1-9) anywhere in a block, the control skips (does not execute) all remaining programmed commands. The block delete feature is turned on with an optionally installed switch on the MTB panel.

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 9.5
Block Delete in a Part Program

Program Block	Comment
`N1000 X__ Z__;`	first block
`N1010  Z__;`	second block
`/1N1020 X__;`	control skips this block if switch 1 is on
`/1/2N1030 X__;`	control skips this block if switch 1 or switch 2 are on
`/N1032 X__;`	control skips this block if switch 1 is on
`N1040 X__;`	
`/2N1050 X__;`	control skips this block if switch 2 is on

The control always reads several blocks into its buffer memory so that it can prepare for moves and commands before it executes them. The switch controlling a block delete must be set **before** that block is read into buffer memory, otherwise it will not be skipped.

The control considers a "/" without a number to mean "/1." However, "/1" must be programmed if more than one block delete number is to be used in a block.

The block delete is active for sequence number search and dry run operations.

The control ignores the block delete when you load a part program from tape or another device into control memory. The control also ignores the block delete when a part program is saved on punched tape or another device from control memory.

For details on the block delete switch(s), see the logic reference manual and the documentation prepared by your system installer.

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.

Important: When performing an EOB search, the search is executed from the beginning of the part program, NOT from the point of display.

To specify an end of block character ";" at the keyboard use the **[EOB]** key on the operator panel. If you are editing part programs off line you cannot enter the end of block character when blocks are keyed in. Refer to chapter 7. The control automatically inserts end of block ";" when the program is downloaded.

Using Subprograms

When the same series of blocks is repeated more than once it is usually easier to program them using a subprogram.

The key difference between a subprogram and a G65 paramacro is that a paramacro always gets a new set of local parameters. A subprogram uses the same set of local parameters that the main program used. See chapter 27 for details on paramacros and local parameters.

This section explains:

- Main and subprograms
- Subprogram calls

Important: To make jumps, loops, or calculations within an executing program or subprogram, use any of the paramacro features described in chapter 27.

You can call a subprogram in an MDI command; however, a MDI command cannot contain an M99 code.

ATTENTION: Any edits that you make to a subprogram or paramacro program (as discussed in chapter 5) that have already been called for automatic execution are ignored until the calling program is disabled and reactivated. Subprograms and paramacros are called for automatic execution the instant that the calling program is selected as active (as described on page 7-5).

Subprogram Call (M98)

Generally, programs are executed sequentially. When you enter an M98Pnnnnn command ("nnnnn" representing a subprogram number) in a program, the control merges the subprogram (designated by the address P) before the block that immediately follows the M98 command. The control issues the error message "CANNOT OPEN SUBPROGRAM", if it cannot find the subprogram designated by the M98 command.

For example,

```
M98 P00001 ;
```

would cause execution to transfer from the current program to the subprogram numbered 00001.

Important: For a program to be used as a subprogram it must have a program name starting with the letter O followed by up to a 5 digit numeric value. When calling the subprogram with a P-word only the numeric value is used. The letter O is omitted.

You might want to execute a subprogram more than one time. For example,

```
M98PnnnnnLmm;
```

would cause the subprogram numbered nnnnn to be merged in the main program mm times. When 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 as many times as 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, execution stops. The program resets or rewinds if an M30 code is executed, or the program ends if the M02 code is executed.

Main and Subprogram Return

M99 code acts as a return command in both sub- and main programs; however, there are specific differences:

Using M99 in a Main Program

If you use M99 in a:	M99:
Main program	executes all commands in the block, regardless if information is programmed in the block to the right of the M99 command
	clears all modal codes similar to an M02 or M30 (simulates start-up conditions)
	resets the current main program to the first block
	automatically performs a cycle start on the program after it is reset and program execution starts over.
Subprogram	tells the control the end of a subprogram
	will not merge any commands within a file that is used as a subprogram and follows a M99 code in the main program into the calling program.

Using M99 in a Subprogram

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 program:	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 9.6
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;		

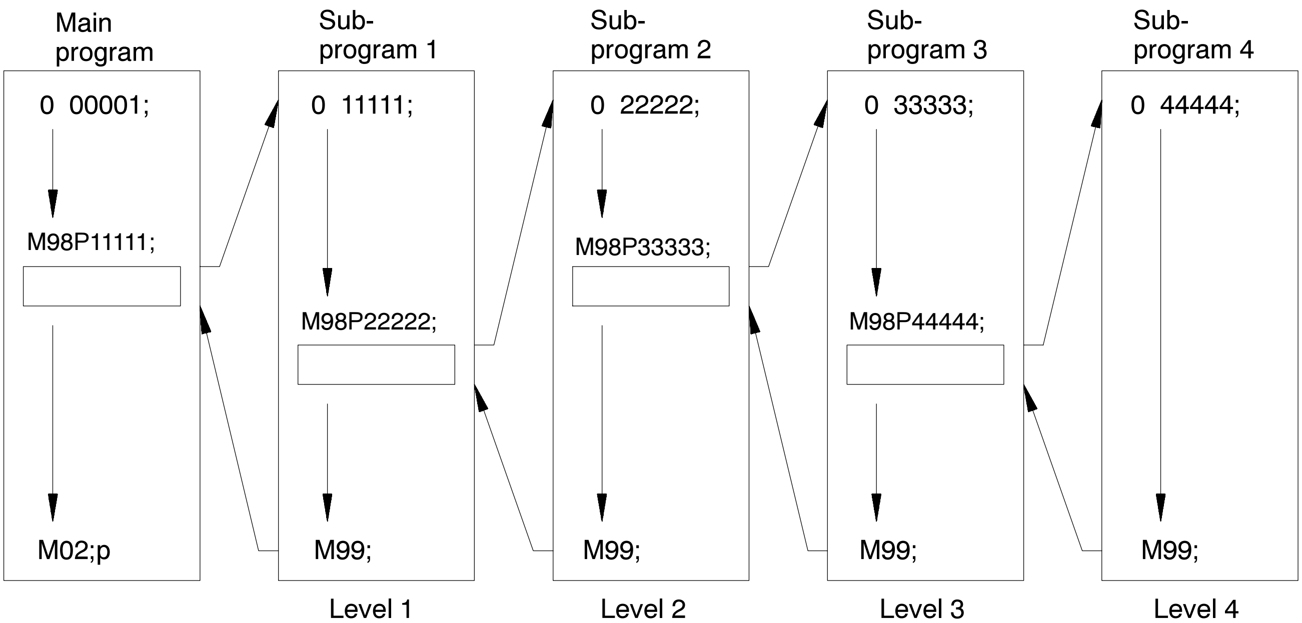
This path of execution results when you select the main program in Example 9.6 as the active program:

```
(MAIN PROGRAM);
N00010...;
N00020...;
N00030M98P1;
(SUBPROGRAM 1);
N00110;
N00120...;
N00130M99;
N00040...;
N00050...;
N00060M98P2L2;
(SUBPROGRAM 2);
N00210;
N00220...M99;
(SUBPROGRAM 2);
NOO210;
N00220...M99;
N00070M30;
```

Subprogram Nesting

We use the term nesting to describe one program calling another. The program called is a nested program. When a subprogram is called from the main program it is on the first nesting level or nesting level 1. If that subprogram in turn calls another subprogram, the called subprogram is in nesting level 2. Subprograms can be nested up to a maximum of 4 levels.

Figure 9.1
Subprogram Nesting



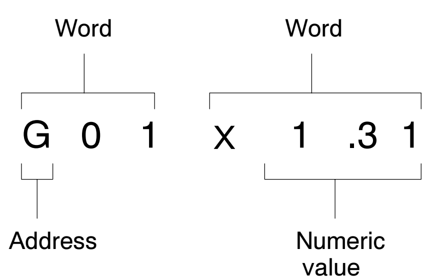
Important: Calling a macro does not add to the nesting level of any active subprograms. Up to 4 subprograms can still be nested, but the combined total of nested macros and subprograms cannot exceed 8. See chapter 27 for information on macros.

Word Formats and Functions

Words in a part program consist of addresses and numeric values.

Component:	Description:
Address	A character to designate the assigned word function.
Numeric value	A numeral to express the event called out by the word.

Figure 9.2
Word Configuration



For each word used in a part program, there is a format that designates the number of digits allowable as a numeric value for that word. The format for an M-code or word, for example, is normally M2 which indicates that an M address can be followed by only two digits.

For words that allow the use of a decimal point in a numeric value, the decimal point format is used. In this case, the numeral to the left of the decimal point indicates the number of digits acceptable as integers, and the numeral to the right of the decimal point indicates the number of fractional digits acceptable.

The format "X3.4" for an X-word, indicates that three digits to the left and four digits to the right of the decimal are acceptable as numeric values. With this format selected, the maximum programmable value for an X-word would be 999.9999.

Leading Zero and Trailing Zero Suppression

The system installer can choose from these programming format types in AMP:

- **Leading Zero Suppression** -- enable or disable
- **Trailing Zero Suppression** -- enable or disable

Table 9.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 below, but the end result would be comparable.

Table 9.A
How the Control Interprets Numeric Values

Programmed X Value	Position Interpreted by the Control		
	TZS Disabled LZS Disabled	TZS Disabled LZS Enabled	TZS Enabled LZS Disabled
X123456.	ERROR	ERROR	ERROR
X12345.6	12345.60	12345.60	12345.60
X1234.56	1234.56	1234.56	1234.56
X123.456	123.45	123.45	123.45
X12345	12345.00	123.45	12345.00
X012345	ERROR	123.45	1234.50
X123456	ERROR	1234.56	12345.60
X1234567	ERROR	12345.67	12345.67
X12345678	ERROR	ERROR	ERROR

Using LZS and TZS with G-codes

The following table illustrates how the control interprets different G-Codes in leading zero and trailing zero suppression modes.

Leading Zero Suppression Mode (decimal assumed at end if not programmed)		Trailing Zero Suppression Mode (2-digit G-code assumed unless decimal point programmed)	
Program this:	**Results in this:**	**Program this:**	**Results in this:**
G02	2	G02	2
G2	2	G2	20
G2.	2	G2.	2
G92	92	G92	92
G920	920	G920	920 or 92 (if no AMP defined macro 920)
G92.1	92.1	G92.1	92.1

Important: If backing up a table using a G10 program (such as the offset tables or coordinate system tables), keep in mind the G10 program output is generated in the current format of the control (LZS or TZS). If you intend to transport this table to a different machine it must also be using the same format.

Programming without Numeric Values

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 (control assumes G00 X0;)
M;     program stop (control assumes M00)
```

would result in the actions described in the comments following the blocks. If the feature is enabled, the error "NUMERIC MISSING" would have occurred upon execution of either of those blocks.

Word Descriptions and Ranges

Table 9.B shows, in alphabetical order, the addresses for words that are recognized by the control, their typical formats, and their general meanings. Since most of these formats are configured in AMP, refer to the documentation prepared by your system installer.

Many of the addresses can be altered in AMP. This table assumes the most common names (such as X and Z for the main axes). Alterable addresses are indicated by the note "AMP assigned."

Later sections discuss these words in more detail, including variations in their meanings when they are associated with certain G-codes. All words discussed in this manual assume that the format and addresses in the following table have not been changed by your system installer.

Important: The formats in this table indicate the maximum number of digits left and the maximum number of digits right of the decimal point for each word. In many cases, they are not valid together since the control allows a maximum of 8 total digits. Refer to your system installer's manual for specific formats.

Table 9.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	8.5	Fixed cycle parameter
E	2.6	3.7	Thread lead
F	8.6	8.5	Feedrate function (F-word)
G	2.1	2.1	Preparatory function (G-code)
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 5.3	5.0 5.3	Subprogram name Length of dwell in G04 and fixed cycles
Q	8.6	8.5	Parameter in fixed cycles
R	8.6 8.6 8.6	8.5 8.5 8.5	Arc radius Return point in fixed cycles QuickPath Plus radius designation
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	8.6 5.3	8.5 5.3	Incremental axis name (Lathe A only) Length of dwell in G04 and fixed cycles
V	8.6	8.5	Incremental axis name (Lathe A only)
W	8.6	8.5	Incremental axis name (Lathe A only)
X	8.6 5.3	8.5 5.3	Main axis (AMP assigned) Length of dwell in G04
Z	8.6	8.5	Main axis (AMP assigned)

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. 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, the smallest programmable distance of axis motion.

Table 9.C
Programming Resolutions

Formats as set in AMP	_.3	_.4	_.5	_.6
Corresponding Resolution	0.001	0.0001	0.00001	0.000001

Refer to your system installer's documentation for the programming resolutions and ranges in a specific system.

Word Descriptions

This section describes general features of the words used in programming. Later chapters in this manual describe how to use these words in detail.

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.

A_L,R_,C (QuickPathPlus Words)

To simplify programming an angle, corner radius, or chamfer between two lines, all that is necessary is the angle between the lines and the radius or chamfer size connecting them. This method of programming can be used to simplify the cutting of many complex parts.

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

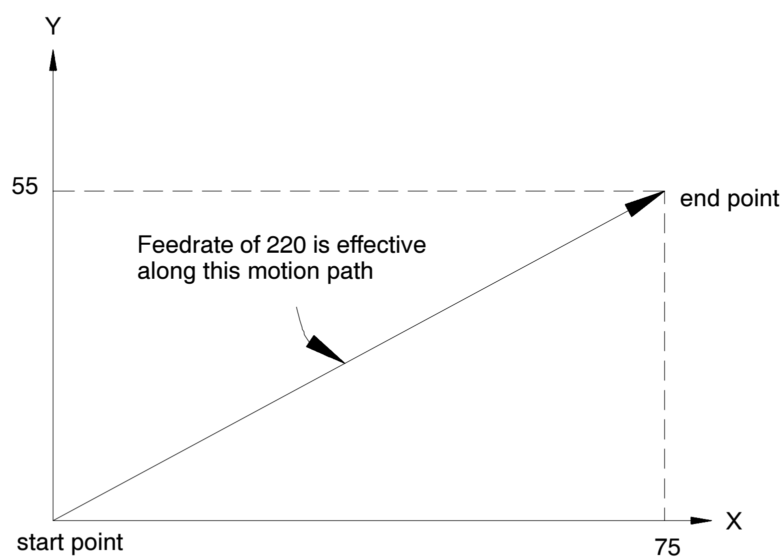
Important: A comma “,” must precede the ,R and ,C address characters for the control to recognize them as radius or chamfer words.

For more details and examples using these words, see chapters 14 and 15.

F-Words (Feedrates)

An F-word with numeric values specifies feedrates for the cutting tool in linear interpolation (G01), and circular interpolation (G02/G03) modes. The feedrate is the speed along a vector of the commanded axes, as shown in Figure 9.3.

Figure 9.3
Feedrate Vectors



The term “feed” refers to moving a tool at a specific velocity in a cutting path. “Feedrate” is the velocity programmed for the feed of a tool.

Feedrates are expressed by the distance of movement per interval. Depending on the mode of the control and the results you want, the distance can be millimeters, inches, meters, or revolutions. The interval can be minutes or revolutions.

Table 9.D
Feedrate Units

Unit/Interval	Abbreviation	Typically Used For:
millimeters per minute	mmpm	linear axis feedrates
inches per minute	ipm	linear axis feedrates
revolutions per minute	rpm	rotary axis feedrates
millimeters per rev	mmpr	threading
inches per rev	ipr	threading

In a metric part program for a linear axis, a feedrate of 100 millimeters per minute (mmpm) typically would be written as F100.; (depending on the active word format).

For details on programming feedrates by using the different feedrate modes, see chapter 17. It also describes special preassigned feedrates.

Important: Feedrates programmed in any of the feedrate modes (G94 or G95) can be overridden by use of the **<FEEDRATE OVERRIDE>** switch.

G-Codes (Preparatory Functions)

The preparatory function is designated by a G-code consisting of address G followed by a two-digit value. In some cases, the G-code may have an additional decimal digit. 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 nonmodal.

Modal	the G-code remains in effect until another G-code in the same G-code group is programmed.
Nonmodal	the G-code is in effect only in the block in which it is programmed.

Important: When you program more than one G-code from the **same** modal group in a block, the control executes the block as the G-codes occur in the block sequentially from left to right. Any mode that is being changed in a block only applies to the values to the right of the G-code. Characters to the left of the G-code still use the old operating mode.

When the control executes an M02 or M30 code the system installer determines in AMP if the modal M- and G-codes reset to their default values. These default values become active at power up, E-Stop reset, or control reset. Your system installer determines these defaults in AMP.

Important: G-codes can also be expressed in terms of a parametric expression (for example G[#12+6]). For details, refer to page 27-2.

Example 9.7 explains execution of modal G-codes, using G00 and G01, both classified into the same G-code group.

Example 9.7
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

Example 9.8 is an example of nonmodal G-code execution.

Example 9.8
Programming Nonmodal 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 9.9
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 9.E breaks down the G-codes into their modal groups. For example, G-codes in group 01 are modal only with other G-codes in group 01. G-codes in the 00 group are nonmodal, and they are effective only in the block in which they are programmed.

Table 9.E
G-code Table

A	Modal Group	Function	Type
G00	01	Rapid Positioning	Modal
G01		Linear Interpolation	
G02		Circular/Helical Interpolation (Clockwise)	
G03		Circular/Helical Interpolation (Counterclockwise)	
G04	00	Dwell	Nonmodal
G09		Exact Stop	
G10L2		Setup Work Coordinate Offset Tables	
G10L3		Setup Tool Management Table	
G10L10		Setup Tool Length Values Geometry Table	
G10L11		Setup Tool Length Values Wear Table	
G10L12		Setup Tool Radius Values Geometry Table	
G10L13		Setup Tool Radius Wear Values Wear Table	
G10.1		Setup Random Tool Table	
G11		Setup Tool Management Table (Cancel)	
G12.1	21	Primary Spindle Controlling	Modal
G12.2		Auxiliary Spindle 2 Controlling	
G14	19	Scaling (Disable)	Modal
G14.1		Scaling (Enable)	
G15	15	Polar Coordinate Programming (Cancel)	Modal
G16		Polar Coordinate Programming	
G16.1		Cylindrical Interpolation	
G17	02	Plane Selection	Modal
G18		Plane Selection	
G19		Plane Selection	
G20	06	Inch System Selection	Modal
G21		Metric System Selection	
G22	04	Programmable Zone 2 and 3, ON	Modal
G22.1		Programmable Zone 3, ON	
G23		Programmable Zone 2 and 3, OFF	
G23.1		Programmable Zone 3, OFF	
G24	00	Feed to Hard Stop	Nonmodal
G25		Adaptive Feedrate (torque mode)	
G26		Adaptive Depth	
G27		Machine Home Return Check	Nonmodal
G28		Automatic Machine Home	
G29		Automatic Return From Machine Home	
G30		Return to Secondary Home	
G31		External Skip Function 1	
G31.1		External Skip Function 1	
G31.2		External Skip Function 2	
G31.3		External Skip Function 3	

A	Modal Group	Function	Type
G31.4		External Skip Function 4	
G36	22	Short Block Acc/Dec Check (Enable)	Modal
G36.1		Short Block Acc/Dec Check (Disable)	
G37	00	Tool Gauging Skip, Function 1	Nonmodal
G37.1		Tool Gauging Skip, Function 1	
G37.2		Tool Gauging Skip, Function 2	
G37.3		Tool Gauging Skip, Function 3	
G37.4		Tool Gauging Skip, Function 4	
G38		Circle Diameter and Center Measurement	
G38.1		Parallel Probing Cycle	
G39	20	Cutter Diameter Comp (Linear Generated Block)	Modal
G39.1		Cutter Diameter Comp (Circular Generated Block)	
G40	07	Cutter Diameter Compensation (Cancel)	
G41		Cutter Diameter Compensation (Left)	
G42		Cutter Diameter Compensation (Right)	
G43	08	Tool Length Offset (Plus)	Modal
G43.1		Tool Length Offset Selection (Plus)	
G44		Tool Length Offset (Minus)	
G44.1		Tool Length Offset Selection (Minus)	
G45	23	Disable Spindle Synchronization	Modal
G46		Set Spindle Positional Synchronization	
G46.1		Set Active Spindle Speed Synchronization	
G47	24	Linear Acc/Dec in All Modes	Modal
G47.1		S-Curve Acc/Dec for Positioning and Exact Stop Mode	
G47.9		Infinite Acc/Dec (No Acc/Dec) (AMP-selectable only)	
G48	00	Reset Acc/Dec to Default AMPed Values	Nonmodal
G48.1		Acceleration Ramp for Linear Acc/Dec Mode	
G48.2		Deceleration Ramp for Linear Acc/Dec Mode	
G48.3		Acceleration Ramp for S-Curve Acc/Dec Mode	
G48.4		Deceleration Ramp for S-Curve Acc/Dec Mode	
G48.5		Programmable Jerk Value	
G49	08	Tool Length Offset Cancel)	Modal
G50.1	11	Programmable Mirror Image (Cancel)	Modal
G51.1		Programmable Mirror Image	
G52	00	Offsetting Coordinate Zero Point	Nonmodal
G53		Motion in Machine Coordinate System	
G54	12	Preset Work Coordinate System 1	Modal
G55		Preset Work Coordinate System 2	
G56		Preset Work Coordinate System 3	
G57		Preset Work Coordinate System 4	
G58		Preset Work Coordinate System 5	
G59		Preset Work Coordinate System 6	
G59.1		Preset Work Coordinate System 7	

A	Modal Group	Function	Type
G59.2		Preset Work Coordinate System 8	
G59.3		Preset Work Coordinate System 9	
G60	25	Synchronous Logic/Block Synchronization Mode	Modal
G60.1		Asynchronous Logic/Block Synchronization Mode	
G60.2		Autosynchronous Logic/Block Synchronization Mode	
G61	13	Exact Stop Mode	Modal
G62		Automatic Corner Override	
G63		Tapping Mode	
G64		Cutting Mode	
G65	00	Paramacro Call	Nonmodal
G66	14	Paramacro Modal Call	Modal
G66.1		Paramacro Modal Call	
G67		Paramacro Modal Call (Cancel)	
G68	16	Part Rotation	Modal
G69		Part Rotation (Cancel)	
G73	09	Deep Hole Peck Drilling Cycle (With dwell)	Modal
G74		Left-Hand Tapping Cycle	
G74.1		Left-Hand Solid Tapping Cycle	
G76		Boring Cycle (Spindle Shift)	
G80		Cancel or End Fixed Cycle	
G81	09	Drilling Cycle (No Dwell, Rapid Out)	Modal
G82		Drilling Cycle (Dwell, Rapid Out)	
G83		Deep Hole Peck Drilling Cycle	
G84		Right-Hand Tapping Cycle	
G84.1		Right-Hand Solid Tapping Cycle	
G85		Boring Cycle (No Dwell, Feed Out)	
G86		Boring Cycle (Spindle Stop, Rapid Out)	
G87		Back Boring Cycle	
G88		Boring Cycle (Spindle Stop, Manual Out)	
G88.1	00	Pocket Milling Roughing Cycle	Nonmodal
G88.2		Pocket Milling Finishing Cycle	
G88.3		Pocket Milling Roughing Cycle	
G88.4		Pocket Milling Finishing Cycle	
G88.5		Hemispherical Milling (Roughing Cycle)	
G88.6		Hemispherical Milling (Finishing Cycle)	
G89	09	Boring Cycle (With Dwell, Feed Out)	Modal
G89.1	00	Irregular Pocket Milling (Roughing Cycle)	Nonmodal
G89.2		Irregular Pocket Milling (Finishing Cycle)	
G90	03	Absolute Mode	Modal
G91		Incremental Mode	
G92	00	Coordinate System Offset (Using Tool Positions)	Nonmodal
G92.1		Coordinate System Offset (Cancel)	
G92.2		Selected Coordinate System Offsets (Cancel)	

A	Modal Group	Function	Type
G93	05	Inverse Time Feed Mode	Modal
G94		Feed-per-minute mode	
G95		Feed-per-revolution Mode	
G98	10	Initial Level Return in Milling Cycles	Modal
G99		R-Point Level Return in Milling Cycles	

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. These default G-codes can be seen on the status display screen after power up or control reset.

I J K Integrand Words

This section describes the axis integrand words. Integrand words define parameters that relate to a specific axis for a canned cycle, probing cycle, or circular motion block, but they are not limited to these operations. For example, in circular motion blocks the axis integrands are used to define the center point of the arc being cut.

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 makes the following assumption:

Integrand Name:	Axis:
I	integrand name for the X axis
J	integrand name for the C or Y axis
K	integrand name for the Z axis

Important: Refer to your system installers documentation to make sure the assumptions are true. If this assumption is not true, it is all examples and formats in this manual that use a I, J, or K need to have their letters replaced with your system installers integrand words accordingly.

M-Codes (Miscellaneous Functions)

The miscellaneous function is designated with an address M followed by a 2- or 3-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 logic program determines whether the M-codes:

- execute at the same time as the axis motion

- execute before the axis motion
- execute after the axis motion is completed

This order of execution can also be altered by using the paramacro feature, system parameter #3003. Refer to chapter 27.

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 default back to their power up condition, and nonmodal M- and G-codes are reset to their default values. If M- and G-codes do not reset, all modal M- and G-codes remain at their present value and nonmodal M- and G-codes remain at their present values.

Table 9.F shows the basic M-codes for the 9/PC control. A part program block can contain as many basic M-codes as you want. If you program more than one M-code from any modal group in the same block, the rightmost M-code in that block for that modal group is the active M-code for the block.

Your system installer cam define additional M-codes in logic. Up to 4 M-codes can be activated in any one block. If more than 4 are programmed in any one block, the right most 4 in that block are activated. Other user M-codes in the block are ignored. Refer to documentation provided by your system installer for details on non-basic M-codes and their operation.

Table 9.F
M-codes

M-code Number	Modal or Nonmodal	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)
			PRIMARY SPINDLE
M03	M	7	Spindle positive rotation (cw)
M04	M	7	Spindle negative rotation (ccw)
M05	M	7	Spindle stop
M19	M	7	Spindle orient
			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
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

(1) Program Stop (M00)

When you execute M00, execution stops 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. 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 logic programming are the responsibility of your system installer.

(3) End of Program (M02)

If you execute a program from control memory, the M02 code acts the same as an M30. Program execution stops and the control 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.

With some machines, the M02 code can also result in a spindle and coolant supply stop. For details, refer to 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. Program execution stops and the control enters the cycle stop state. The program is reset to the first block and a **<CYCLE START>** begins part program execution 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, refer to the instruction manual prepared by your system installer.

(5) Overrides Enabled (M48)

When your execute M48, the feedrate override, rapid feedrate override, and the spindle speed override functions become effective. 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 cancel M-code (M49) to ignore any override set by the operator on the MTB panel. When you ignore the override setting, the axis feedrate, rapid feedrate, and the spindle speed override values are all set to 100 percent. An M49 cancels an M48 and your system installer can choose which is active upon power-up. This override setting is ignored if you are using programmed motion.

(7) Constant Surface Speed Mode Disabled (M58)

M58 cancels M59 mode, and it allows the control to recognize programmed G96 constant surface speed mode and S-words to be specified. 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 cutting tool 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.

Z or the spindle speed can be directly designated using an S code.

(9) Subprogram Call (M98)

When you execute M98, a subprogram is called and executed. This word can be used in any program including an MDI program. For details on programming an M98, see page 9-6.

(10) End of Subprogram or Main Program Auto Start (M99)

M99 End of Subprogram or Paramacro program
When you execute M99, subprogram execution is completed and program execution returns to the calling program. This word is not valid in an MDI command, but it can be contained in a subprogram called by an MDI command. For details on programming an M99, refer to page 9-6 or chapter 27.

M99 End of Main Program with Auto Start
If you execute a program from memory, an M99 as the last block in a main program stops program execution at that location. The program is reset to the first block and a **<CYCLE START>** automatically starts program execution for you.

ATTENTION: The M99 code is commonly used as the end of program for fully automated systems that automatically load the next part to be machined. This code requires that some logic interface be written that assures the part is fully loaded and ready for machining before block execution is allowed to restart. Failure to do so can cause injury to operators or damage to equipment.

For these systems some logic interface should be written to assure that the part is fully loaded before program execution is restarted.

Important: You cannot use these M-codes when TTRC is active.

Other more specific M-codes are described in later sections that deal specifically with their functions.

Important: When you activate the MISCELLANEOUS FUNCTION LOCK feature, the control displays M-, B-, S-, and T-words in the part program and activates the corresponding Tool Wear Offset, with the exception of M00, M01, M02, M30, M98, and M99.

2nd Miscellaneous Function (B-Word)

Your system installer may decide to 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 may be used instead of B by setting the proper AMP parameter. For details, refer to documentation prepared by your system installer, or the AMP programming manual.

The B-word is designated by a 2- or 3-digit numeric value following address B. Unlike M-codes, each block can contain only one B-word.

Important: When you activate the MISCELLANEOUS FUNCTION LOCK feature, the control displays M-, B-, S-, and T-words in the part program and activates the corresponding Tool Wear Offset, with the exception of M00, M01, M02, M30, M98, and M99. This feature is described on page 7-1.

Important: If the system cannot locate an N or O word when conducting an N or O search, the cursor positions itself directly below the last block in the program (M02).

N-Words (Sequence Numbers)

Each block in a part program can be assigned up to a 5-digit numeric value following an N address. These numbers are referred to as sequence numbers and are used to distinguish one block from another.

Sequence numbers can be assigned at random to specific blocks or to all blocks if desired. Blocks assigned sequence numbers can be called later by designating their sequence number. Sequence numbers are necessary to make program jumps and to specify a block for subprogram calls and returns. For details on sequence number, refer to page 9-4.

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. An O-word can have up to 5 numeric characters following it.

P, L Words (Main Program Jumps and Subprogram Calls

When the same series of blocks are repeated more than once it is usually easier to program them using a subprogram.

This section explains:

- Main and subprograms
- Subprogram calls

Important: To make jumps, loops, or calculations within an executing program or subprogram, use any of the paramacro features described in chapter 27.

P-words in a subprogram call (M98) or paramacro call are used to designate the specific program being called. The P address is followed by the program name being called.

L-words in a subprogram call (M98) and some cycles are used 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, refer to page 9-6.

S-Words (Spindle Speed)

Program spindle speeds (in RPM) using an S-word with up to five integer digits and three decimal digits. The actual legal format is defined in AMP by the system installer. A common S-word is used to program all of the spindles AMPed to be in the system.

Important: Your system installer sets a maximum speed in AMP for each gear range for each spindle configured in AMP. If an S-word is programmed requesting a spindle speed that exceeds this limit. The spindle speed holds at the AMP-defined maximum. A new value may be set for this maximum RPM by programming a G92 code followed by an S-word.

When programming an S-word in a block that contains axis motion commands, the logic program has the option to temporarily suspend the axis motion commands until the spindle reaches speed. The control has the ability to take the programmed spindle speed and automatically search for the gear range that is AMPed to allow the necessary RPM. The operation of gear changing and how it is implemented is very logic dependant. Refer to your system installer's documentation for details on how a gear change operation is performed.

Override spindle speeds designated in a program with the **<SPINDLE SPEED OVERRIDE>** switch on the MTB panel. This switch can be positioned in five percent increments within a range of 50 - 120 percent. For details, refer to your system installer's instruction manual.

Use the override cancel M-code (M49) to ignore any override set on the MTB panel. When the override setting is ignored, the axis feedrate, rapid feedrate, and the spindle speed values are all set to 100 percent. For more information on spindle functions, refer to chapter 16.

Important: When you activate, the MISCELLANEOUS FUNCTION LOCK feature, the control displays M-, B-, S-, and T-words in the part program and activates the corresponding Tool Wear Offset, with the exception of M00, M01, M02, M30, M98, and M99. This feature is activated as described on page 7-1.

Cutting Speed

The term "cutting speed" refers to the velocity of the surface of the revolving cutting tool relative to the workpiece. Cutting speeds are determined by the spindle speed in revolutions per minute (rpm) and the diameter of the cutting tool in the following equation:

```
Metric Units                     English Units

    3.14159 x D x N                  3.14159 x D x N
V = ---------------            V = -----------------
         1000                               12
```

Where :	Is :
V	cutting speed in meters or feet per minute
D	diameter of the tool in millimeters or inches
N	rpm of the spindle

Or, stating the equations in terms of required spindle rpm:

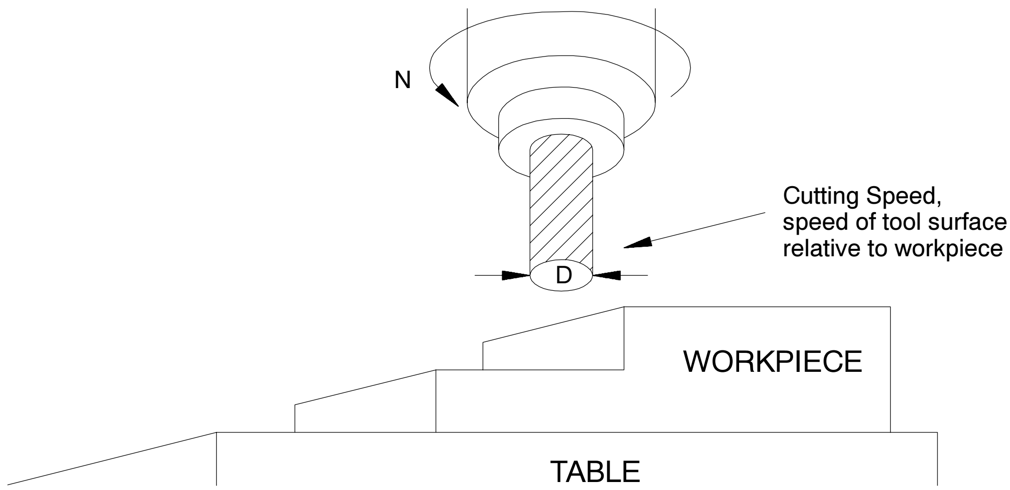
```
Metric Units                  English Units

               V                        V
N = 318.30989 ---          N = 3.81972 ---
               D                        D
```

From the metric equation, if a desired cutting speed is known, for example V = 100 surface meters per minute using a cutting tool with a diameter, D = 100 millimeters, the spindle speed, N is equal to approximately 318 rpm.

Spindle speed is programmed with an S-word. In this above example the S-word would be S318.

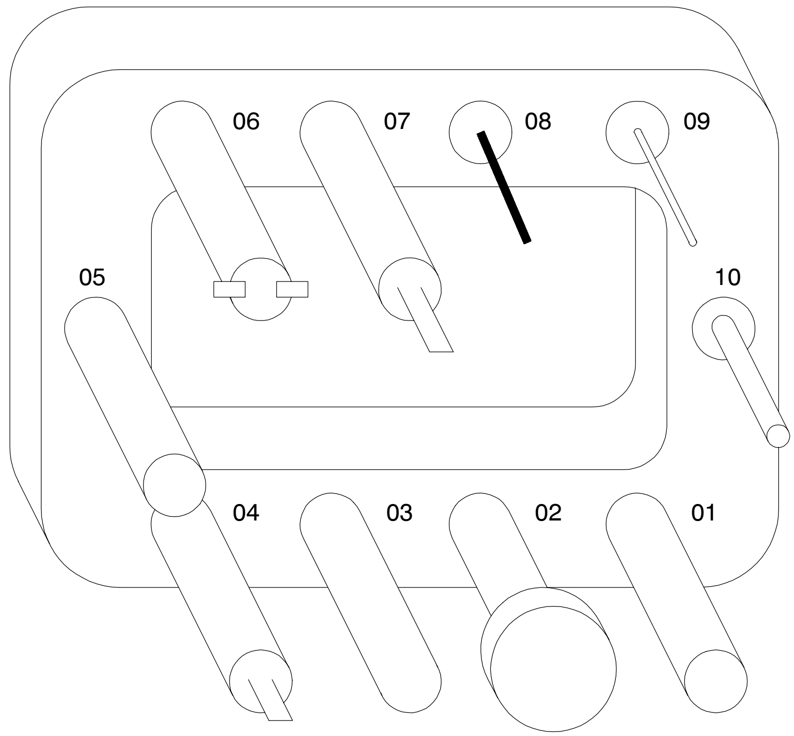
Figure 9.4
Cutting Speed



T-Words (Tool Selection)

A workpiece usually requires different kinds of cutting processes, and usually there are cutting tools that correspond to each process. The cutting tools are typically stored in a tool magazine and are assigned tool numbers (see Figure 9.5).

Figure 9.5
Tool Magazine



A T-address followed by a numeric value programs a tool selection. When the control executes the T-word, it outputs a tool selection signal to a tool changer. The tool changer should perform a sequence of operations to deliver the proper tool in response to the tool selection signal. For example, to select a cutting tool that is assigned tool number "03", write "T03" in the part program. The system installer may require a M06 in the program to cause a tool change.

Since tool changers vary in style, size and function, the system installer is responsible for specific implementations through logic. Refer to your *9/PC Logic Reference Manual* and the manual supplied by the system installer for more details.

Important: When changing cutting tools it is usually necessary to change the tool offset at the same time. This is done with an H- or a D-word. For details refer to page 9-16.

Important: When you activate the MISCELLANEOUS FUNCTION LOCK feature, the control displays M-, B-, S-, and T-words in the part program and activates the corresponding Tool Wear Offset, with the exception of M00, M01, M02, M30, M98, and M99. This feature can be activated through the front panel screen as described in chapter 2.

END OF CHAPTER

Basic Control Operation

Chapter Overview

This chapter covers the control of the coordinate systems on the 9/PC 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. This chapter describes:

A thorough understanding of this group makes programming easier by allowing full control of the coordinate systems.

Machine (Absolute) Coordinate System

The 9/PC control has two types of coordinate systems.

Coordinate System:	Description:
work coordinate system	defined based on the coordinate system used in the part drawing of a part to be cut by the machine. Programs are usually written based on the work coordinate system.
machine coordinate system (often referred to as the absolute coordinate system)	unique to the individual machine tool.

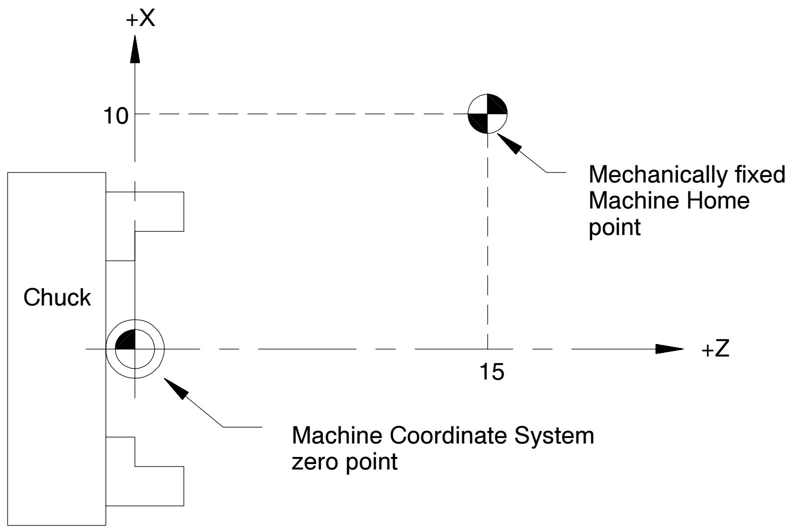
The machine coordinate system is the basic coordinate system set for every machine. It is established after completion of the machine-homing operation. It cannot be offset or shifted in anyway. Its position is determined in AMP by your system installer.

Important: Before you can activate any coordinate system, the machine must be homed. The homing operation refers to the positioning of the axes to a machine-dependent, fixed position which is called the machine home. For more on machine homing, refer to page 4-7.

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 you establish, the machine coordinate system is not affected by a control reset operation or any other programming or operator operation.

Figure 10.1
Machine Coordinate System, Home Coordinate Assignment



In Figure 10.1, your system installer defined the machine coordinate system zero point by assigning the machine home point to have the coordinates X=10 and Z=15.

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.

Important: Normally, the control displays the current axes positions in respect to active work coordinate system. The position in the machine coordinate system can be displayed by selecting the absolute screen as described in chapter 8.

Motion in the Machine Coordinate System (G53)

Although axis motion is usually commanded in the work coordinate system, axis motion is possible when a G53 is programmed in a block if you reference coordinate values in the machine coordinate system.

```
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 tool travels to this position in either G00 or G01 mode, depending on which is active when the G53 block is executed. Any attempt to execute a G53 block in G02 or G03 mode generates an error.

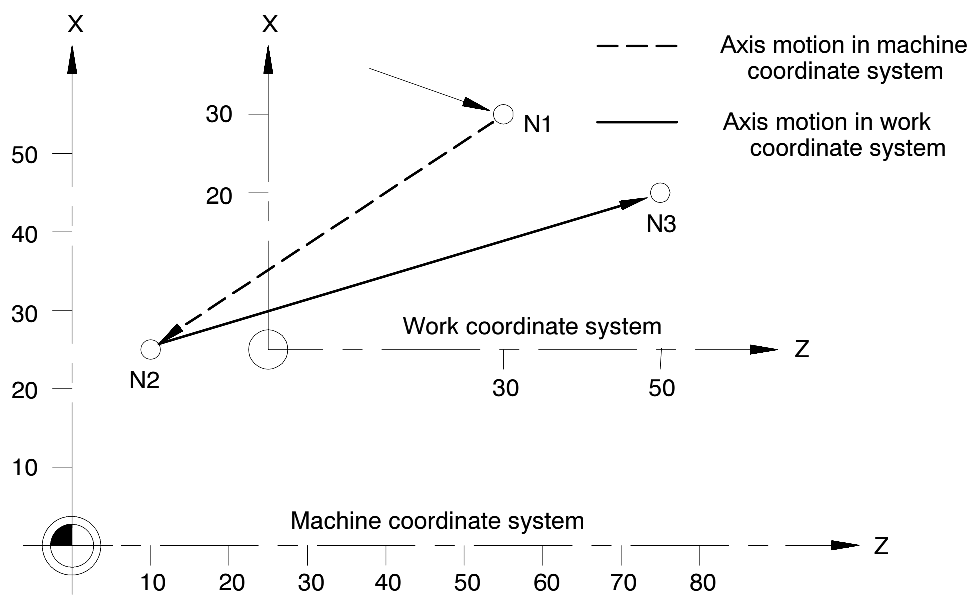
The G53 code is not modal. It is effective only in the block in which it is called. After a G53 block, the control returns to the coordinate system that was in effect prior to the G53 blocks execution.

Important: The control must be in absolute mode (G90) when the G53 command is executed. If a G53 is executed while in incremental mode (G91), the control ignores the G53 code and any axis words in the G53 block.

Example 10.1
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 10.2
Results of Example 10.1

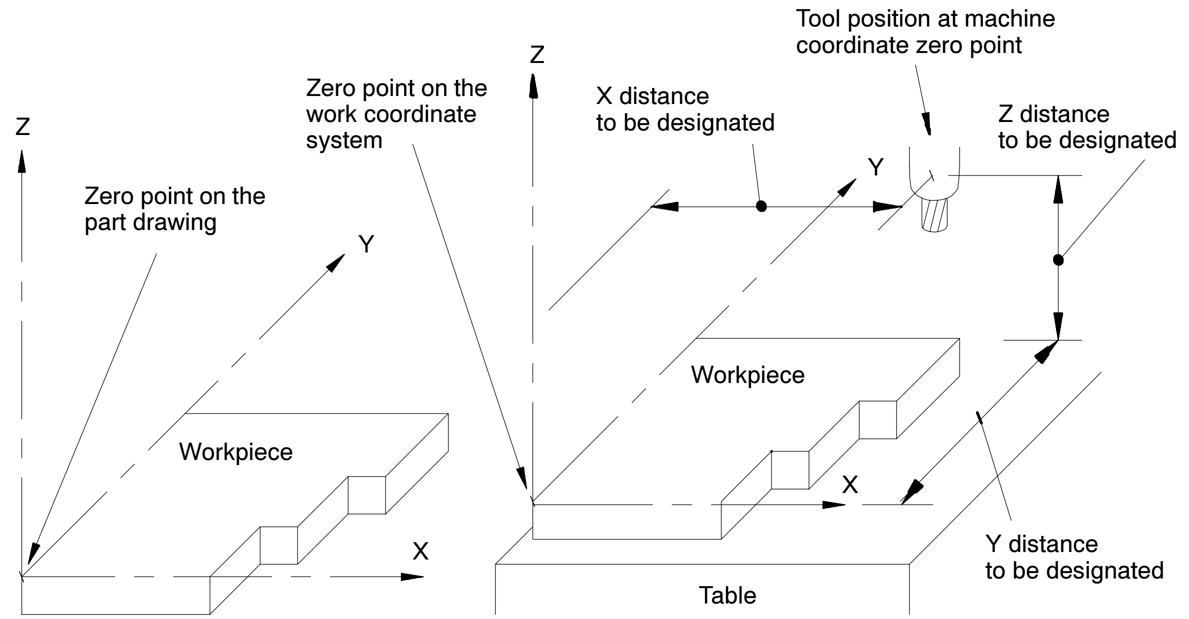


Preset Work Coordinate System (G54 - 59.3)

When you cut a workpiece using a part program made from a part drawing, you want 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 10.3, you establish the work coordinate system by programming the distance between the desired zero point of the work coordinate system and the zero point of the machine coordinate system.

Figure 10.3
Work Coordinate System

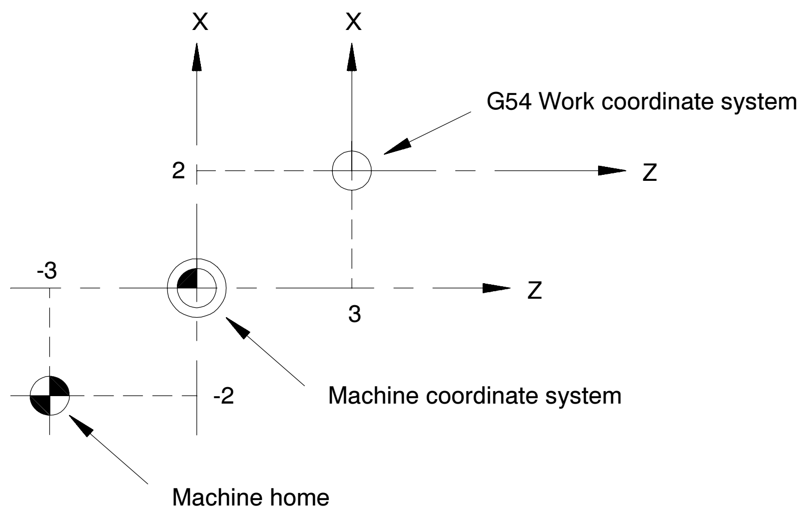


There are nine preset work coordinate systems selectable using G54 - G59.3. The required work coordinate system can be selected by specifying any of these G-codes in the program.

Work coordinate systems called out by G54 - G59.3 have zero points that you enter in a work coordinate system table (refer to chapter 4). 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 you complete 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 established when you execute a control reset operation, E-Stop, G92.1, or power up. The default work coordinate system is the sum of the external offset value (if any), and the offsets of the default coordinate system selected in AMP (G54-G59.3 or none). If the default coordinate system is selected as none, the default work coordinate system is simply the external offset (if any). This manual assumes G54 to be the default coordinate system and no external offset has been entered.

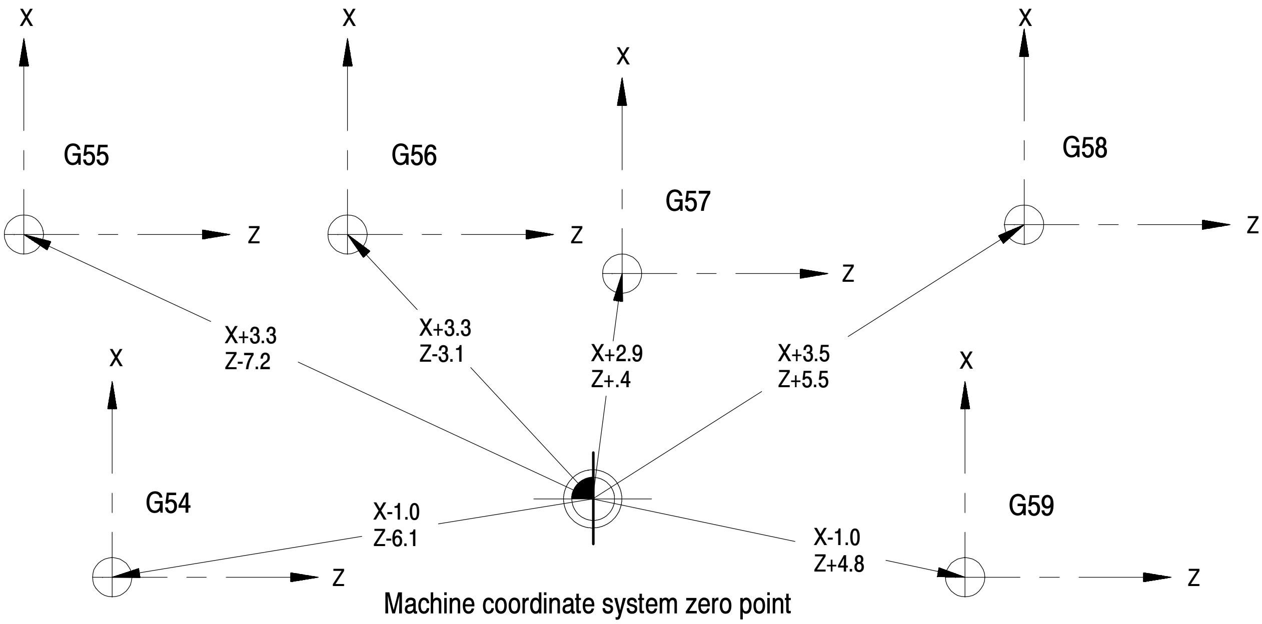
Figure 10.4
Work Coordinate System Definition



In Figure 10.4, the machine coordinate system was defined by declaring the fixed position machine home as the point X=-3., Z=-2. Then the G54 work coordinate system zero point was 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 10.5
Examples of Work Coordinate System Definition

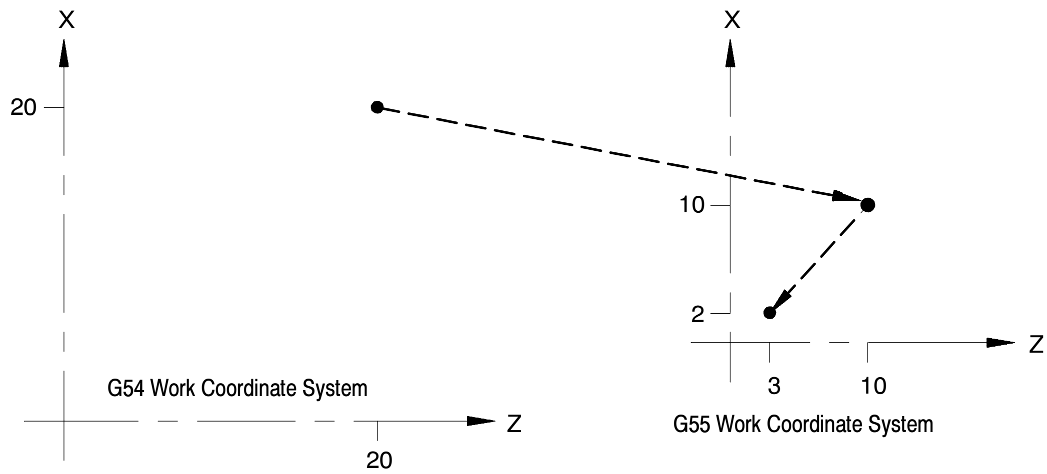


To change work coordinate systems, specify the G-code corresponding to the work coordinate system you want in a program block. Any axis motion commands in a block that contains a change from one work coordinate system to another is executed in the work coordinate system specified in that block.

Example 10.2
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 10.6
Results of Example 10.2



Altering Work Coordinate System (G10L2)

There are 4 methods to change the value of a work coordinate system zero point in the work coordinate system table. Three methods can be found in the following sections:

- Manually alter the work coordinate system table as described in chapter 3.
- Alter the paramacro system parameter values 5221- 5386 as discussed in chapter 27.
- Alter through some external means defined by the system installers logic program (refer to your *9/PC Logic Reference Manual*).

The fourth method, and the one discussed in this section, alters the work coordinate system table through G10 programming. Changing the values in the table using any of these methods does not cause axis motion; however, it does immediately shift the active coordinate system by the amount entered. The format for altering the work coordinate systems using G10 is as follows:

```
G10L2P__X__Y__Z__;
```

Important: The order of the words in this program block is important. The L and P words must be programmed before any axis words are programmed in the G10 block. Failing to follow this order can result in data being misinterpreted and loaded into the table incorrectly.

Where :	Is :
L2	tells the control that you want to alter the coordinate system tables.
P	specifies which coordinate system (G54 through G59.3) you want to work on. P1 through P9 correspond to the work coordinate systems G54 through G59.3. P1 = G54 work coord. system P6 = G59 work coord. system P2 = G55 work coord. system P7 = G59.1 work coord. system P3 = G56 work coord. system P8 = G59.2 work coord. system P4 = G57 work coord. system P9 = G59.3 work coord. system P5 = G58 work coord. system
X_Y_Z_	specify the location of the zero point of the specified work coordinate system relative to machine coordinate system.

Important: G10 blocks may not be programmed when TTRC is active.

Incremental/Absolute Mode and the G10L2 Command

When you program in:	Then:
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.
absolute mode (G90)	any values entered into the work coordinate system table using the G10 command **replace** the currently active work coordinate system values.

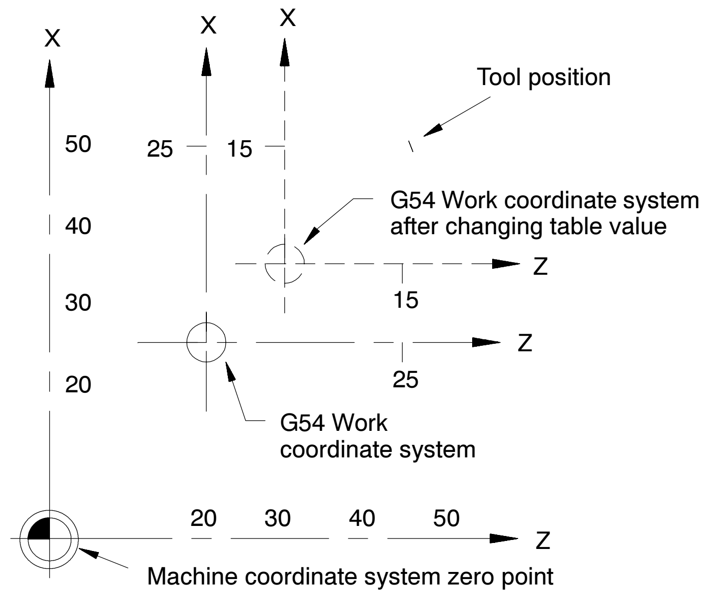
Example 10.3 and Figure 10.7 illustrate how the work coordinate system is shifted by using G10.

Example 10.3
Work Coordinate System Shift Using G10

Program block	Work coordinate Position	Absolute coord. 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 new table values for the work coordinate systems are saved even when control power is turned off.

Figure 10.7
Results of Example 10.3



Work Coordinate System External Offset

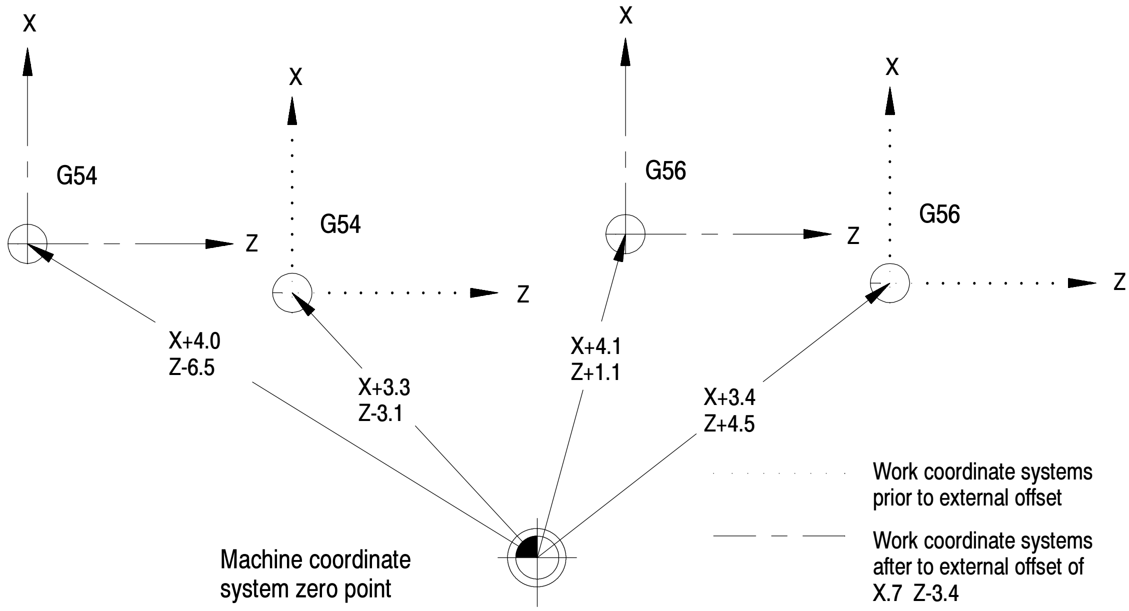
The external offset allows all work coordinate system zero points to be shifted simultaneously, relative to the machine coordinate system. This offset can compensate for part positioning shifts that result when a different tool is installed. It can also compensate for tool position shifts that result from a different tool fixture.

The external offset can also be used to match the work coordinate systems on mechanically different machines. The machines can then use the same part program with the same G54-G59.3 coordinate values. This allows part programs to be less machine dependant.

Four methods are available to modify the external offset:

Method:	Active Immediately after block execution	Active next Cycle Stop	Active next re-home or power cycle
Programming a G10	X		
System installer logic offset		X	
Manually through table		X	
Paramacro Programming			X

Figure 10.8
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.

Altering External Offset (G10L2)

There are four methods used to change the value of an external offset in the work coordinate system table. Three methods can be found in the following sections:

Method:	Refer to:
manually alter the external offset value in the work coordinate system table	Chapter 3
alter the paramacro system parameter values 5201- 5206	Chapter 27
alter through some external means defined by the system installer's logic program	9/PC Logic Reference Manual

The fourth method, and the one discussed in this section, alters the external system table through G10 programming. Changing these values in the table using any of these methods does not cause axis motion; however, it does immediately shift the active coordinate system by the amount entered.

The values entered into the external offset are added to the work coordinate system zero point values each time a work coordinate system is called. The format for altering the external offset using G10 is:

```
G10 L2 P0 O__ X__ Z__;
```

Where :	It :
`L2`	tells the control that you want to alter the coordinate system tables.
`P0`	designates the external offset as the offset to update.
`O__`	specifies whether the value entered for the diameter axis is a radius or diameter value. (O is non-modal.) `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_`	specifies the location of the zero point of the specified work coordinate system relative to machine coordinate system.

When you execute this block, the control immediately shifts the currently active work coordinate system by the new external offset amount.

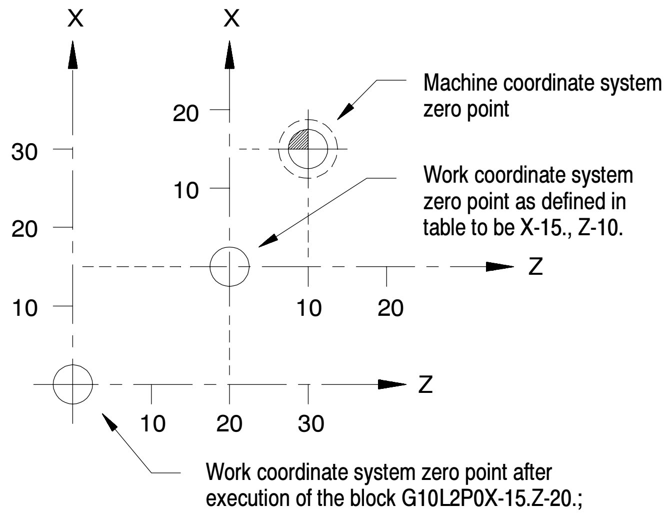
Example 10.4 and Figure 10.9 illustrate how the work coordinate system is shifted using G10.

Example 10.4
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 new table values for the work coordinate systems are saved even when control power is turned off.

Figure 10.9
Results of Example 10.4



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 they are canceled when you remove power to the control. They may also be cancelled by an M02, M30, or control reset, depending upon the selections made in AMP by your system installer.

Important: All of these offsets are global in nature. This means that they apply to all work coordinate systems. When you change work coordinate systems (programming G54-G59) consider the effects of these offsets on the new work coordinate system.

Tool geometry and wear offsets **are not** effected by an offset made to the work coordinate system.

Important: We recommend that tool offsets for geometry and wear be canceled before you execute any work coordinate system offsets. If tool offsets are not canceled, the work coordinate system offset is added to the active tool offset. This can cause confusion when you change tool offsets later in the program. Refer to page 10-18 on canceling tool offsets.

Coordinate Offset Using Tool Position (G92)

Use the G92 command in a part program to offset the currently active work coordinate system relative to the current tool position. A G92 block in a program offsets the zero point of the work coordinate system a specified distance from the current tool position.

G92.2 cancels G92 without canceling any other work coordinates. This differs from G92.1, which cancels all coordinate system offsets. A control reset may cancel this offset, depending upon the selections made in AMP by your system installer.

When a G92 command is executed in a program, it cancels any other active work coordinate system offsets that may have been in effect including G52 offsets, set zero, or jogged offsets. External offsets are not affected. When the logic flag BW_INHR is set, it cancels G92.

Important: A tool offset is not automatically canceled when you execute a G92 block. This can result in undesired effects on the work coordinate system when tool offsets are changed later.

The following G92 block offsets the work coordinate system so that the current tool position takes on the coordinate values programmed in the G92 block.

```
G92 X___ Z___;
```

For example specifying values of zero for all axes in a G92 block causes the current tool position to become the zero point of the current work coordinate system.

Execution of a G92 block does not produce any axis motion.

Important: Any axis not specified in the G92 block is not offset, and the current coordinate position for that axis remains unchanged.

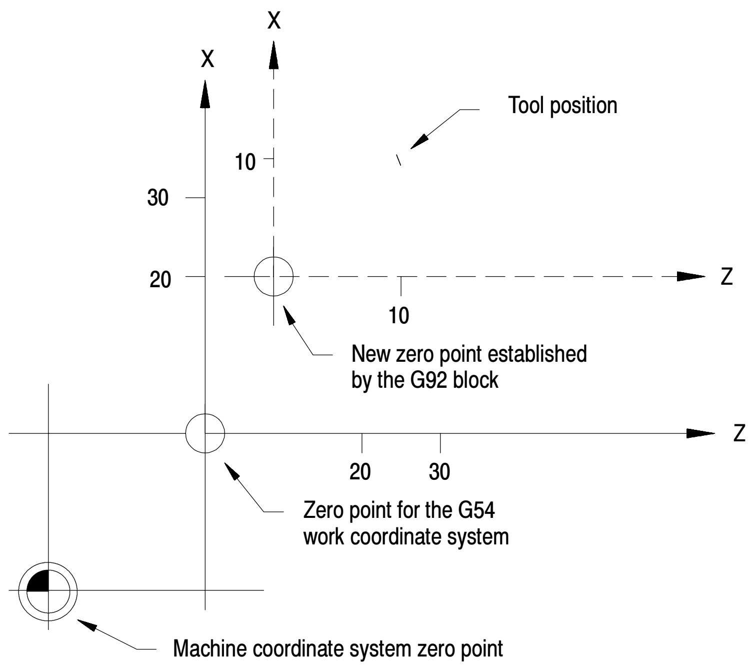
Once the work coordinate system is offset, all absolute positioning commands in the program are executed as coordinate values in the offset coordinate system.

Example 10.5
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 tool 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 10.10
Results of Example 10.5



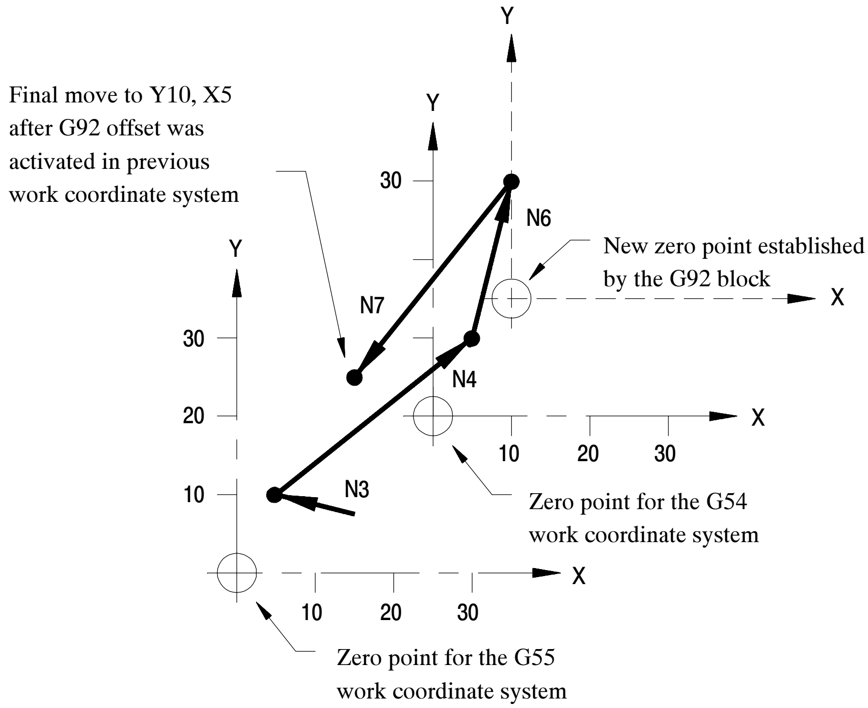
ATTENTION: G92 offsets are global. Changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59.3) unless the effects of the offset have been considered.

Example 10.6 shows the effect of changing work coordinate systems while the G92 offset is active.

Example 10.6
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 tool 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 10.11
Results of Example 10.6



In Figure 10.11, the offset entered for the G54 work coordinate system has also shifted the G55 coordinate system. Any offsets described in this section alter all of the work coordinate system (G54 - G59) at the same time.

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___ ;
```

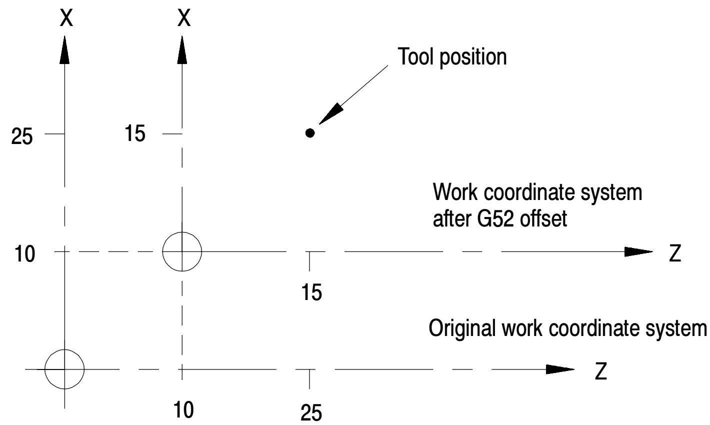
This command offsets the current work coordinate system by the axis values that follow the G52 command.

Example 10.7
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 this example no axis motion takes place when the G52 block is executed. The work coordinate system position values change. See Figure 10.12.

Figure 10.12
Results of Example 10.7



The G52 work coordinate system zero point offset can be canceled by programming a G52 block with zero values for the axes to be cancelled. The following block would cancel the work coordinate system offset for the X axis only.

```
G52 X0;
```

A G52 offset can also be canceled by executing a G92 or G92.1, performing a control reset or an E-STOP reset operation, or executing an end of program M30 or M02. A G92 command only cancels a G52 offset if one is active when the G92 block is executed. A G52 offset can be activated at some time after the G92 block is executed even if a G92 offset is still in effect.

ATTENTION: G52 offsets are global. Changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59) unless the effects of the offset have been considered.

Set Zero Offset

When you perform a Set Zero operation, the control shifts the current work coordinate system so that the current tools position is the zero point of the coordinate system. The axis where you want to perform a set zero on is selected through logic (refer to your system installer's documentation).

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 **not** cancel a G52 offset. The G52 remains active and continues to offset the current tool position in the work coordinate system. When the G52 offset is canceled later, the coordinate system shifts.

The Set Zero offset can be canceled by programming a G92.1 command, executing a control reset operation, executing an E-STOP reset operation, or programming an end of program M30 or M02 command. A control reset may cancel the Set Zero offset, depending upon the selections made in AMP by your system installer.

ATTENTION: Set Zero offsets are global. Changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59) unless the effects of the offset have been considered.

Example 10.8
Typical Set Zero Offset Application

Operation	Comment
`-Manual jog-`	axes are manually jogged to a location where the operator has determined that a special operation must be performed.
`-Set Zero-`	operator performs a Set Zero offset to establish the work coordinate system zero point at the current axis location
`-Run program-`	a generic special operation program can now be executed from the axis coordinate position that resulted from the manual jog and Set Zero

The set zero offset can be performed through an optional switch installed by your system installer.

Jog Offset

The jog offset feature lets you manually create a desired offset by jogging the axes during an automatic or MDI operation.

Important: This feature functions only if your system installer has supplied a special switch and the appropriate logic programming. See the “Jog Offsets” and “Jog-on-the-fly” logic flags in the logic reference manual or refer to the documentation supplied by your system installer.

Press a special switch after interrupting an automatic or MDI operation to activate this feature. Any manual jog moves you make are added to the current work coordinate position as an offset. When you press cycle start to continue execution, the jogged distance for each axis remains as a coordinate offset for that axis.

ATTENTION: Jog offsets are global. Changing from one coordinate system to another does not cancel the offset. Do not specify a change in coordinate systems (G54-G59) unless the effects of the offset have been considered.

You can cancel the jog offset by programming a G92.1 command, executing a control reset operation, executing an E-STOP reset operation, or programming an end of program M30 or M02 command.

To use this feature, follow these steps:

1. Press **<CYCLE STOP>** or **<SINGLE BLOCK>** on the MTB panel to interrupt automatic or MDI operation.
2. Turn on the switch to activate the jog offset feature (refer to documentation provided by your system installer).
3. Change to manual mode, unless the control is equipped for the "Jog-on-the-Fly" feature which allows jogging in automatic or MDI modes (refer to documentation prepared by your system installer).
4. Jog the axes using any of the available jog types (with the exception of homing) as described on page 4-1. The control adds the amount of the jog move as an offset for each jogged axis.
5. Return to Automatic or MDI mode. When you press the **<CYCLE START>** button, execution continues from the new tool location at the jogged offset.

Important: When you move the jog offset, 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.

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 system to default condition

It does not cancel an external offset. Refer to page 10-10.

The G92.1 block also reestablishes 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 generate an error. When you execute the G92.1 block, all G92, G52, set zero, and Jog offsets are canceled on all axes. You cannot cancel the offsets on only one or more of the axes.

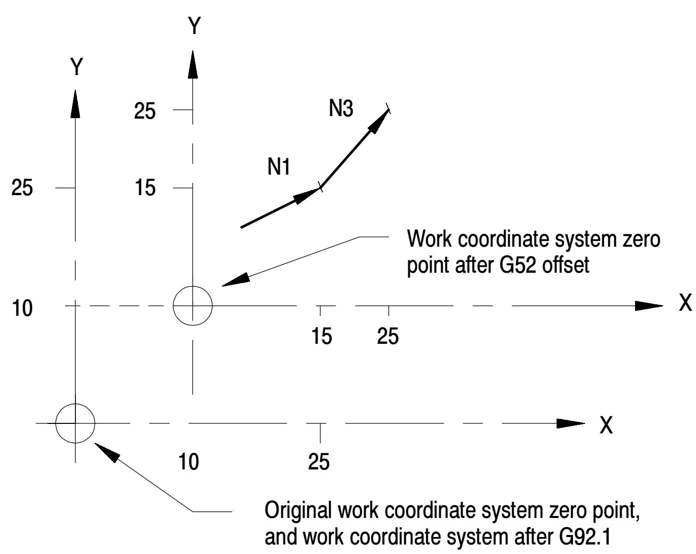
No axis motion takes place during execution of a G92.1 block. Axes remain at their last programmed positions while the work coordinate system adjusts to remove all offsets.

Example 10.9 demonstrates the G92.1 offset cancel.

Example 10.9
G52 Offset Cancelled by a G92.1

Program Blocks	Comment
`N1 G01Y25.X25.;`	move to Y25, X25
`N2 G52Y10.X10.;`	work coordinate system is offset by Y10, X10
`N3 Y25.X25.;`	move to Y25, X25 in the offset coordinate system
`N4 G92.1;`	G52 offset is cancelled, program position displays axis position at X35Y35.

Figure 10.13
Results of Example 10.9



Canceling Selected Coordinate System Offsets (G92.2)

The G92.2 command cancels these offsets:

G92 work coordinate system offset
Set zero offset
Jog offset

It does not:

- cancel an external offset
- reset the current work coordinate system (G54-G59.3)
- cancel a G52 offset

The G92.2 block must be programmed with no axis words. Axis words in a G92.2 block generate an error. When you execute the G92.2 block, all G92, set zero, and Jog offsets are canceled on all axes. You cannot cancel the offsets on only one or more of the axes.

No axis motion takes place during execution of a G92.2 block. Axes remain at their last programmed position while the work coordinate system adjusts to remove these offsets.

Logic Offsets

Your system installer has the option of activating, deactivating, or altering the value of these offsets through logic:

- Work coordinate systems
- External offset
- Tool length offsets (geometry and wear)
- Tool tip radius offsets (geometry and wear)
- Tool orientation

These offsets can be modified through a custom display page created by your system installer or through some other input to logic.

There can be an impact on 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. If a cutter compensation offset has been modified by logic, the control does not update the look-ahead buffer unless the offset is currently active. Refer to documentation supplied by your system installer for details on specific logic offset operations.

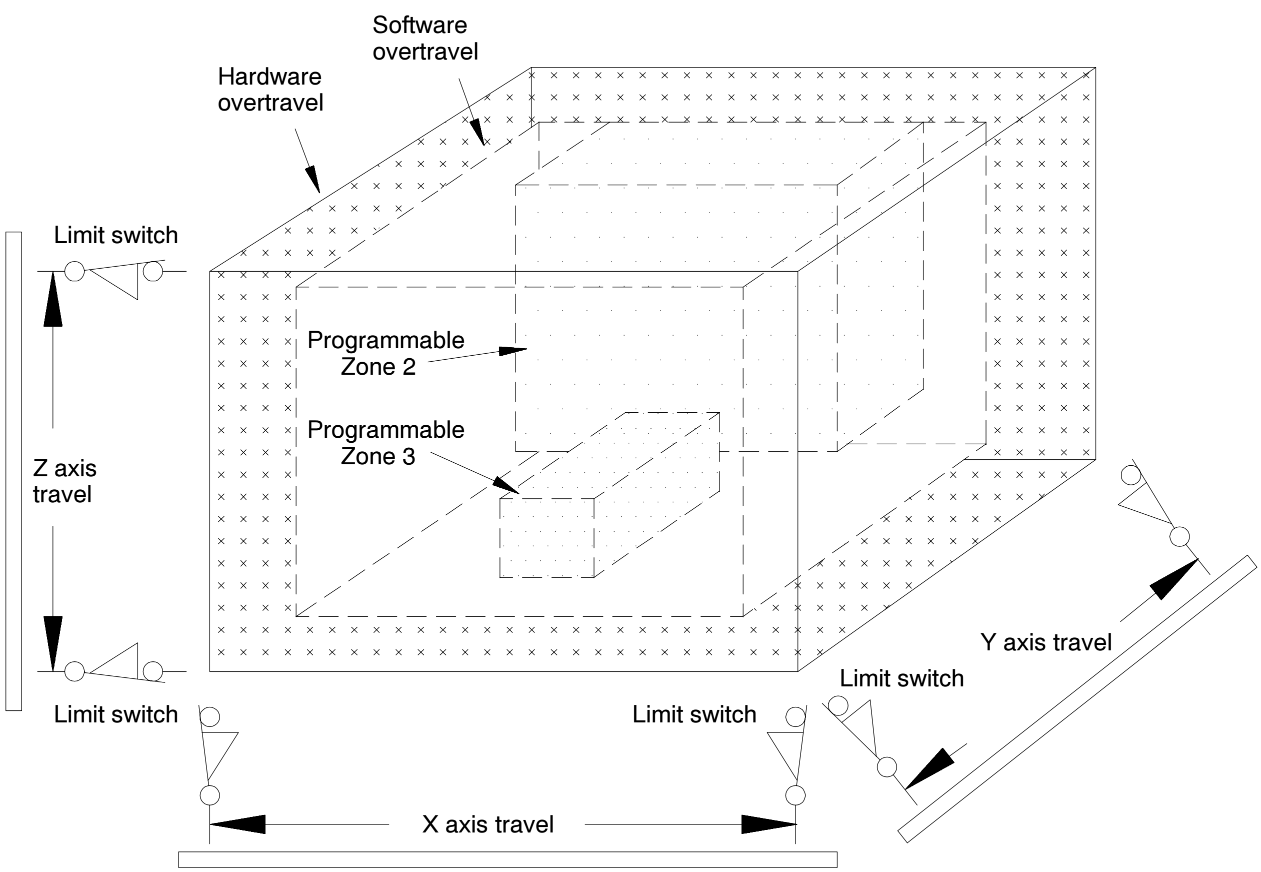
END OF CHAPTER

Overtravels and Programmable Zones

Chapter Overview

Overtravels and programmable zones define areas that restrict the movable range of the cutting tool. The 9/PC control is equipped to establish two overtravel areas and two programmable zones as illustrated in Figure 11.1.

Figure 11.1
Overtravels and Zones



There are two types of overtravels:

1. **Hardware overtravels** -- Established by your system installer by mounting mechanical limit switches on the movable range of the axes
2. **Software overtravels** -- Established in AMP by your system installer designating coordinate values in the machine coordinate system

There are two types of Programmable Zones:

1. **Programmable Zone 2** -- Established by the operator, or person in charge of job setup. The machine coordinate system boundaries for this zone are entered in a table. Programmable zones may be turned on and off in the part program.
2. **Programmable Zone 3** -- Established by the operator, programmer, or person in charge of job setup. The machine coordinate system boundaries for this zone are entered in a table **or through programming.** Programmable zones may be turned on and off in the part program.

Hardware Overtravels

When the machine tool 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 so that when the machine attempts to move beyond a range determined by your system installer the limit switch is tripped. When the limit switch is tripped axis motion stops. The area defined by these limit switches is referred to as the hardware overtravel.

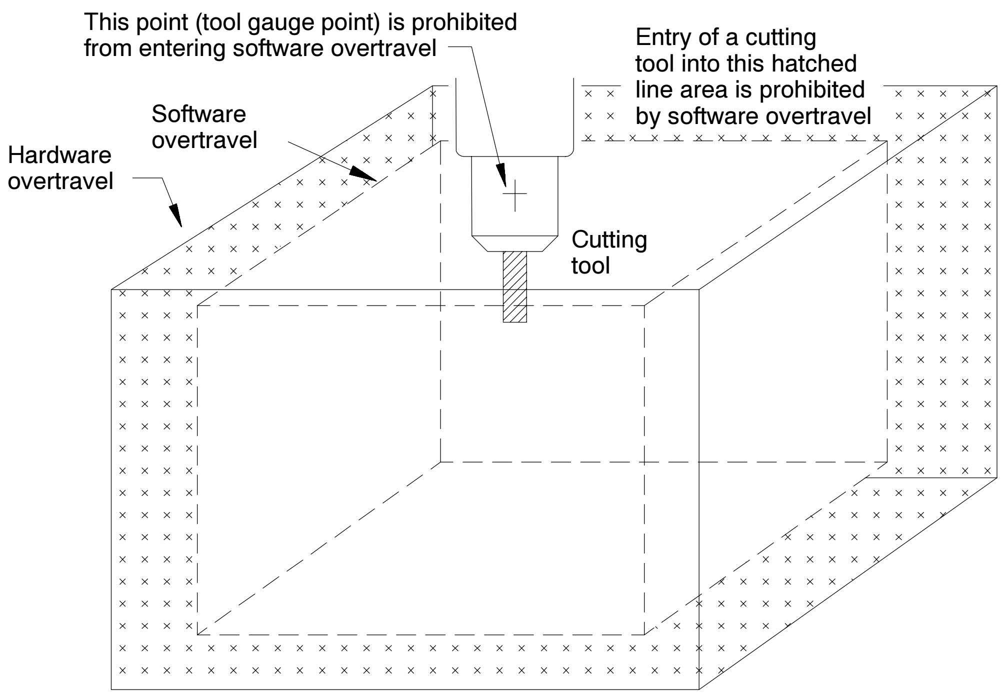
ATTENTION: The area defined by a hardware overtravel does **not** take into account any tool offsets. This can allow the actual tool to enter the restricted area without the axes entering it.

These switches are installed to prevent the machine from motion that exceeds a range that can cause damage to the machine. Frequently your system installer wires the hardware overtravel directly into the E-STOP string. This stops all motion and disables the axis drives. Refer to the literature provided by your system installer for instructions on moving axes out of hardware overtravel.

Software Overtravels

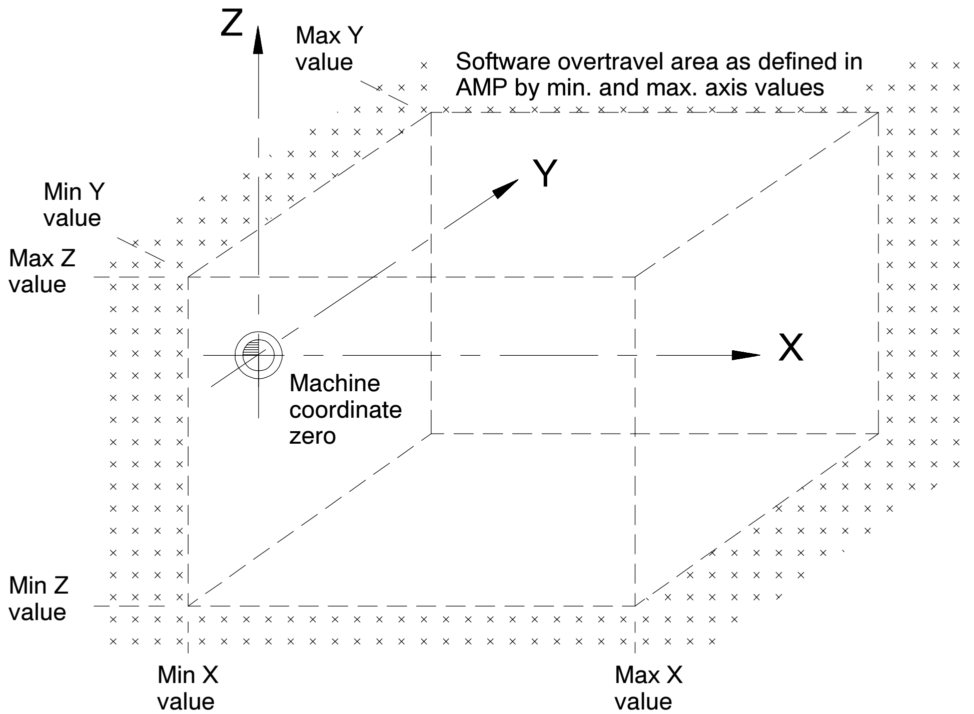
The coordinate values of the points defining the software overtravels are set in AMP by your system installer. This overtravel can only be disabled by your system installer in AMP. If your system installer has enabled the software overtravels, the control is not allowed to exit the area defined by the software overtravels.

Figure 11.2
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 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 11.3
Area Defining Software Overtravel



Typically the software overtravels are located within the hardware overtravels (maximum axis travel defined by the limit switches on each axis), and they 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, refer to page 11-12.

ATTENTION: The area defined by a software overtravel does not take in to account any offsets. This allows the actual tool to enter the restricted area without the axes absolute position entering it. Make sure this is considered when the software overtravel is established.

Programmable Zone 2

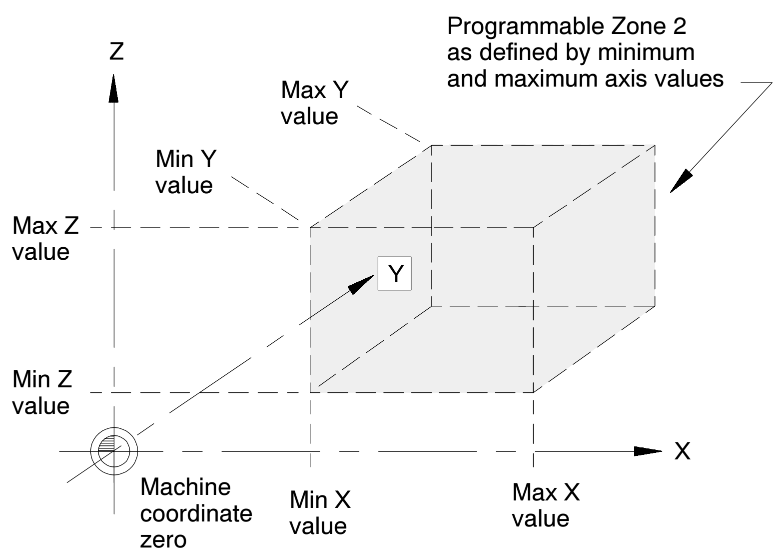
Programmable zone 2 defines an area which the tool **cannot enter**. Generally, zones are used to protect some vital area of the machine or part located within the software overtravels.

Important: Programmable zones are defined using coordinates in the machine coordinate system. They are **not** affected by any changes in the work coordinate system, including external offsets.

ATTENTION: Programmable zones only protect the tool tip from entering the zone (as determined with the currently active tool length offset). They do not protect other moving members from collision with objects in the programmable zone's boundary.

Values for programmable zone 2 are entered in the programmable zone tables as described in chapter 3. These values represent a maximum and a minimum value in the form of machine coordinate values for each axis. The area defined by these points establishes the boundaries for programmable zone 2.

Figure 11.4
Area Defining Programmable Zone 2



Programmable zones 2 and 3 become active when a G22 block is executed and are cancelled when a G23 is executed. Both G22 and G23 are modal commands.

Important: You must home your axes first before the control will enable the programmable zones.

Important: When changing a tool offset or activating a programmable zone 2, the current tool tip location must be outside of the area defined by programmable zone 2.

G22 programmable zone 2 and 3 active
G23 programmable zone 2 and 3 inactive

G23 is normally automatically made active at power up, though this is ultimately determined by the system installer in AMP. Your system installer also determines in AMP if an M02 or M30, control reset, or E-Stop reset cancels programmable zones that you have turned on or off while executing your program.

Important: If you program a G22, any axis words included in the block are stored as the coordinates for programmable zone 3. Refer to page 11-7.

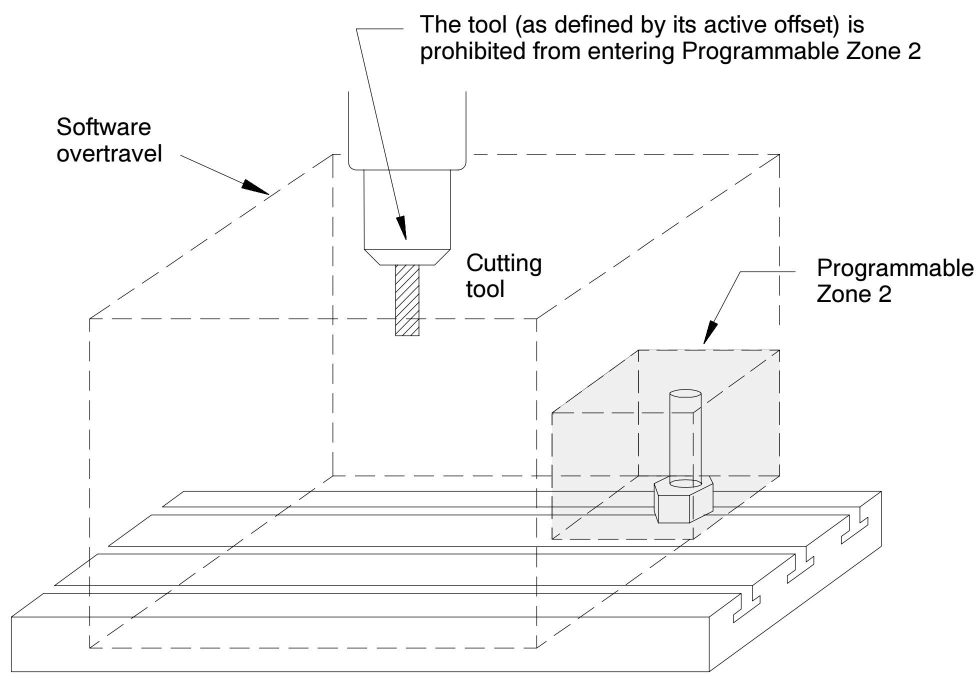
If you attempt to program some other command in a G22 or G23 block, for example:

```
G22 G01 X12.;
```

the control issues the error message:

"UNUSABLE WORDS IN ZONE BLOCK"

Figure 11.5
Programmable Zone 2



Programming this G-code:	turns Zone 2:	turns Zone 3:
G22	On	On
G22.1	Off	On
G23	Off	Off
G23.1	No Change*	Off

* A G23.1 turns on programmable zone 2 if it is the default power up condition configured in AMP (also activated at a control reset). G23.1 does not turn on programmable zone 2 when it is activated in a part program.

Your system installer can also turn zones on and off with PAL. Refer to your system installer's documentation for more information.

For details on how the control reacts to entry into a prohibited area refer to page 11-12.

Programmable Zone 3

Important: G22.1, G23, and G23.1 must be programmed in blocks without other commands. If programming a G22, any axis words included in the block will be stored as the coordinates for programmable zone 3.

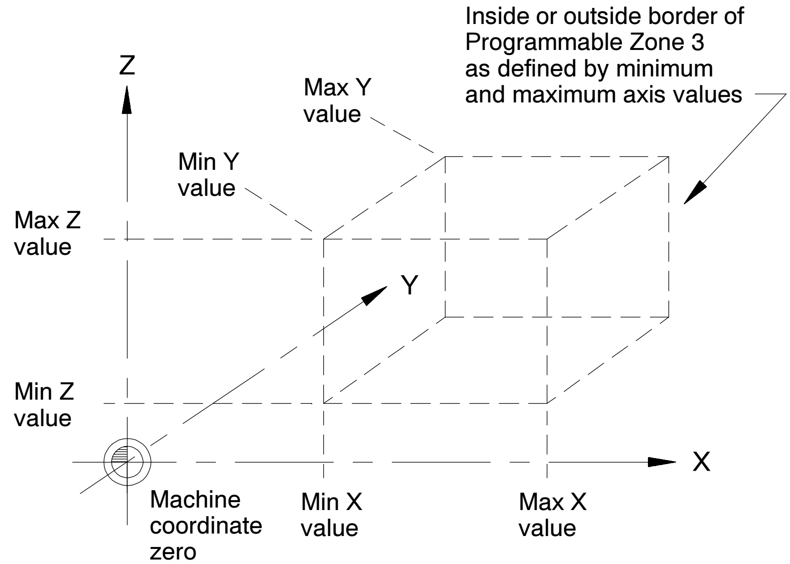
Programmable zone 3 can define an area which the tool **cannot enter or an area the tool cannot exit**. The current tool location determines when programmable zone 3 is made active. Generally, zones are used to protect some vital area of the machine or part located within the software overtravels.

Important: Programmable zones are defined using coordinates in the machine coordinate system. They are **not** affected by any changes in the work coordinate system, including external offsets.

ATTENTION: Programmable zones only protect the tool tip from entering the zone (as determined with the currently active tool length offset). They do not protect other moving members from collision with objects in the programmable zone's boundary.

Values for programmable zone 3 are entered either in the programmable zone table (described on page 3-18) or through a G22 program block. A maximum and a minimum coordinate value (in the machine coordinate system) are assigned for each axis. The resulting coordinates define the boundaries for programmable zone 3.

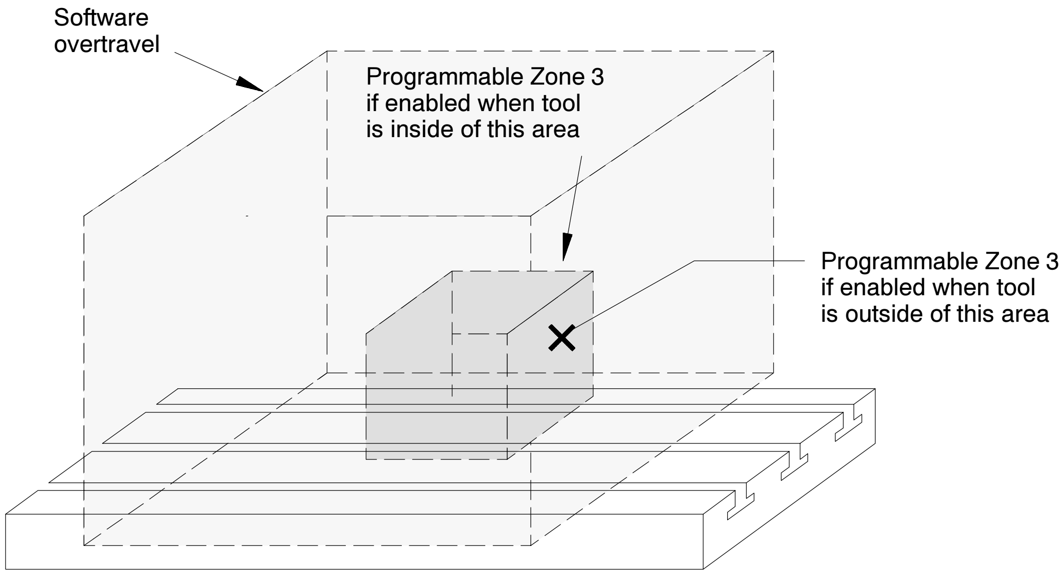
Figure 11.6
Area Defining Programmable Zone 3



Unlike the software overtravels and programmable zone 2, programmable zone 3 can define either an area that the cutting tool can not exit or an area that the cutting tool can not enter. This is determined by the current tool location when programmable zone 3 is made active.

- It defines an area that the cutting tool <u>may</u> <u>not</u> <u>exit</u> if the tool is currently inside the zone when the zone is activated.
- It defines an area that the cutting tool <u>may</u> <u>not</u> <u>enter</u> if the tool is currently outside the zone when the zone is activated.

Figure 11.7
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.

Important: You must home your axes first before the control will enable the programmable zones.

Program G-code:	To turn on these zones:	To turn off these zones:
G22	2 and 3	not applicable
G22.1	3	2
G23	not applicable	2 and 3
G23.1	2*	3

* A G23.1 only turns on programmable zone 2 if it is activated via a control reset or power up condition. G23.1 does not turn on programmable zone 2 if it is programmed. To turn on programmable zone 2 and turn off programmable zone 3, first program a G22 followed by a G23.1 to turn off programmable zone 3. Programming G23.1 has no affect on zone 2.

G22.1 and G23.1 are modal (G22.1 cancels G23.1 and G23.1 cancels G22.1). G22 and G23 belong to a different modal group than G22.1 and G23.1. This means that programmable zone 2 may be activated without activating programmable zone 3 if a G23.1 is executed.

G23 is automatically active at power up, control reset, or E-STOP reset as the default G code for this modal group.

Your system installer can also turn zones on and off with PAL. Refer to your system installer's documentation for more information.

Programming Zone 3 Values (3 or fewer axes)

You can reassign values for the parameters that establish programmable zone 3 by programming axis words in a G22 program block. Two methods are available. This section discusses programming values for zone 3 when three or fewer axes have been configured on the system (this does not include any spindle).

Define values for programmable zone 3 using the G22 command followed by axis words in the following format:

G22 X__ Z__ U__ I__ K__ J__;

Where:	Defines:
Primary axis words (normally X, Z, and U)	maximum zone limits
Integrand words (normally I, K, and J)	minimum zone limits

These axis words can vary. Refer to your system installer's documentation.

The zone values entered in a G22 block always reference coordinate values in the machine coordinate system.

If a value for a maximum axis parameter is less than the value set for an axis current minimum parameter, or if a value for a minimum axis parameter is set greater than the value set for an axis current maximum value, the control displays the message:

"INVALID VALUE (MAX < MIN) FOR ZONE 3 AXIS (X)"

This message displays the name of the axis that has been set incorrectly. It does not indicate if it is the minimum or maximum value that is incorrect.

If the same integrand word is assigned in AMP by the system installer to more than one axis, that integrand word will set the lower zone 3 limit for all axes with that integrand.

Programming Zone 3 Values (4 or more axes)

You can reassign values for the parameters that establish programmable zone 3 by programming axis words in a G22 program block. Two methods are available. This section discusses programming values for zone 3 when four or more axes have been configured on the system (this does not include any spindle).

This method differs from the three axis method in that the same integrands can be used again for different axes (necessary since the control only supports three integrand words). Assume the following AMP configuration:

Primary Axis name	X	Y	Z	U	V	W	A	B	C
Axis Integrand	I	J	K	I	J	K	I	J	K

These axis words can vary. Refer to your system installer's documentation.

Define values for programmable zone 3 using the G22 command followed by axis words in the following format:

G22 X__ Y__ Z__ I__ J__ K__;
G22 U__ V__ W__ I__ J__ K__;
G22 A__ B__ C__ I__ J__ K__;

Where:	Defines:
Primary axis words	maximum zone limits
Integrand words (normally I, J, and K)	minimum zone limits

Using this method, the same integrand word assigned in AMP to more than one axis correspond only to the absolute axis words programmed in the G22 block. Integrand words cannot be programmed alone (without a absolute axis word in the G22 block). The following example assumes a machine with axes configured as shown above.

These blocks:	Results in:
G22 X10 I-10 Y14 J-14 Z1 K-1; G22 U5 I-5 V13 J-2 W11 K10; G22 A3 I2 B7 J-7 C12 K11;	upper and lower zone 3 limits for all 9 axes are changed. Zones 2 and 3 are both activated when the first block in this series of blocks is executed.
G22 X1 Y2 Z3 U4 V5 W6 A7 B8 C9;	upper zone 3 limits are changed for all 9 axes. Zones 2 and 3 are both activated.
G22 X1 Y2 Z3 U4 V5 W6 A7 B8 C9 I-1 J-2 K-3;	upper and lower zone 3 limits for all 9 axes are changed. (I sets lower for X, U, and A: J sets lower for Y, V, and B: K sets lower limits for Z, W, and C). Zones 2 and 3 are both activate.
G22 K-10;	error is generated. Current status of zones remains in current state (on or off).

ATTENTION: When using multiple blocks to set the zone 3 limits, keep in mind zone 3 is activated after the first G22 block. This will result in zone 3 being activated before you have completed changes to the zone 3 values. This can cause the control to miss-interpret zone 3 as an internal or external zone, depending on the tool location at the time of the zone activation.

The zone values entered in a G22 block always reference coordinate values in the machine coordinate system.

If a value for a maximum axis parameter is less than the value set for an axis current minimum parameter, or if a value for a minimum axis parameter is set greater than the value set for an axis current maximum value, the control displays the message:

"INVALID VALUE (MAX < MIN) FOR ZONE 3 AXIS (X)"

This message displays the name of the axis that has been set incorrectly. It does not indicate if it is the minimum or maximum value that is incorrect.

Resetting Overtravels

Tool motion stops during overtravel conditions that occur from 3 causes:

Cause:	Description:
Hardware overtravel	the axes reach a travel limit, usually set by a limit switch or sensor mounted on the axis. Hardware overtravels are always active.
Software overtravel	commands cause the axis to pass a software travel limit. Software overtravels are active only after the axis has been homed provided the feature has been activated in AMP by the system installer.
Programmable zone overtravel	The tool reached a travel limit established by independent programmable areas. Programmable Zones are activated through programming the appropriate G-code.

In all cases, the control issues an error message.When an overtravel condition occurs, all axis motion stops, the control goes into cycle stop and one of the these error messages appears.

Error Message:	Description:
HARDWARE OVERTRAVEL (-) BY AXIS (X)	indicates that the specified axis has tripped either the + or - hardware limit switch mounted on the machine.
SOFTWARE OVERTRAVEL (+) BY AXIS (X)	indicates that 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) BY AXIS (X)	indicates that a tool has reached the specified axis overtravel area defined by either programmable zone 2 or 3.

When an overtravel of any type occurs, axes cannot move in the same direction as the feed causing the overtravel. Only axis motion in the reverse direction is possible.

How a hardware overtravel condition is reset depends on the E-Stop circuit design and the way logic 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 an 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.
4. Make sure it is safe to move the axis away from the overtravel limit.
5. Use any of the jog features described on page 4-1 except homing, to manually move the axis away from the limit.

END OF CHAPTER

Coordinate Control

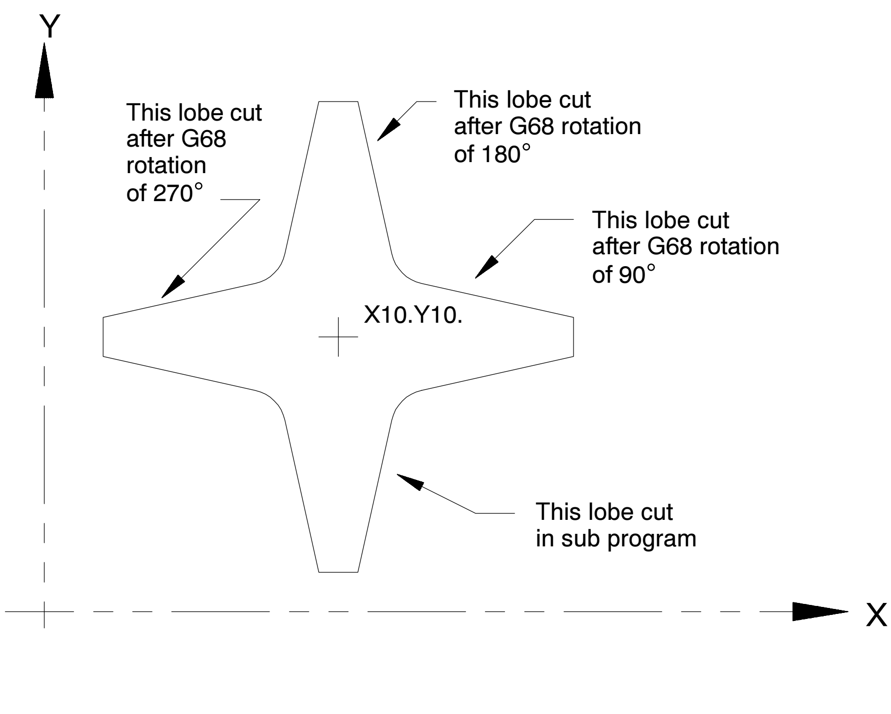
Chapter Overview

This chapter describes 9/PC coordinate control.

Rotating the Coordinate Systems

The 9/PC has a feature (G68) that can rotate the work coordinate system. There is also a feature called External Part Rotation which rotates all work coordinate systems by simulating a rotation of the machine coordinate system. Rotating the coordinate systems can prove to be useful when cutting a part that contains symmetrical geometries (see Figure 12.1).

Figure 12.1
Part With Symmetrical Geometry



Rotating the Current Work Coordinate System (G68, G69)

To rotate the current work coordinate system, program the following command.

```
G68 X__ Y__ Z__ R__;
```

Where :	Is :
X, Y, Z	Specify the center of rotation using 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	Specify the angle of rotation that the coordinate system is to be rotated at. Enter a value in units of degrees. R is always measured parallel to the first major axis in the current plane. Positive R is measured counter clockwise, and a negative R is measured clockwise. If you do not specify an angle using an R-word in the G68 block, the control uses the value of the “Programmable Part Rotation” angle field from the rotation table shown in figure Figure 12.6.

Important: If the first motion command following the G68 command is an incremental move, the center of rotation as described in the G68 command is ignored and the coordinate system is rotated about its zero point (see Example 12.1). The first motion command following a G68 block cannot be a circular move.

Any unnecessary parameters in the G68 block are ignored. Any G-codes other then a plane selection (G17, G18, or G19) or a change from absolute or incremental mode (G90, or G91) specified in a G68 block will result in an error.

The G68 block will not create any axes motion. Position displays change due to the alteration of the work coordinate system.

Any rotation of the work coordinate system by programming a G68 command will rotate only the currently active work coordinate system. When changing to a different work coordinate system, the rotation will not be applied to the new work coordinate system. When changing back to the rotated coordinate system the rotation will still be in effect.

If you do not program an R-word in the G68 block, the value of the angle for programmed rotation is taken from the part rotation screen. Access this screen as described for external part rotation on page 12-8. The last field on this screen is the programmable part rotation angle. This angle is only used when the R-word is excluded in the G68 block.

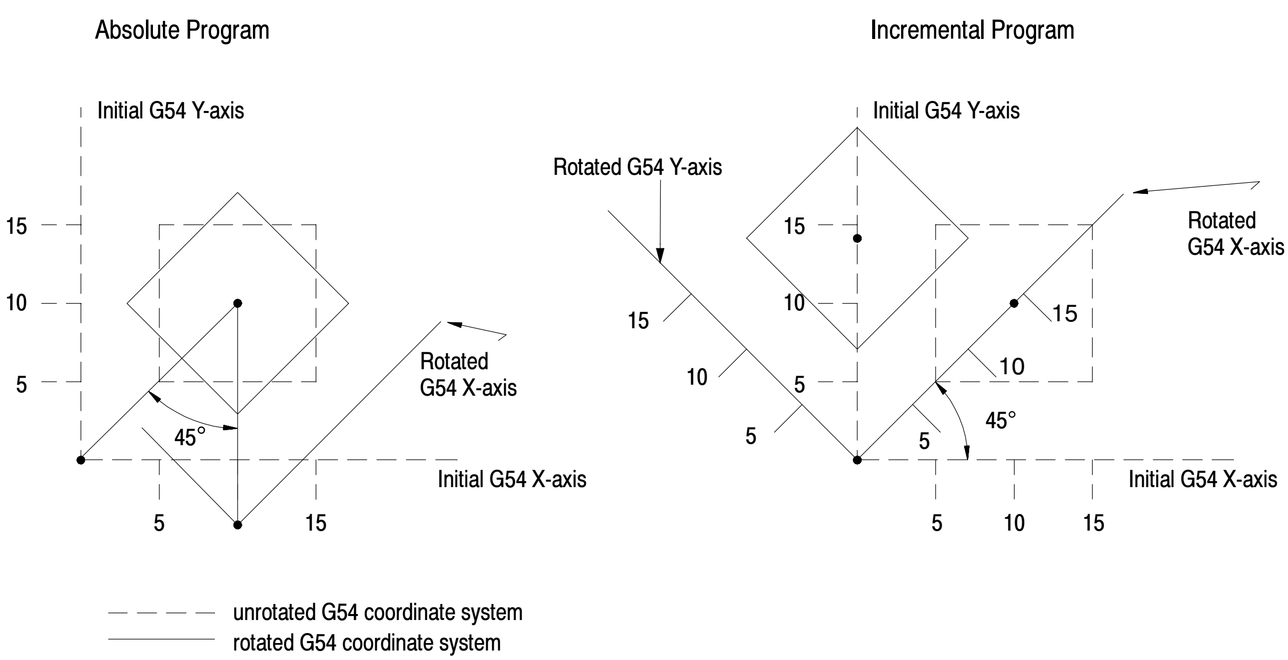
These program blocks cause the rotation of the active work coordinate system as shown in Figure 12.2.

Example 12.1
Rotating the Active Work Coordinate System (G68)

ABSOLUTE PROGRAM	INCREMENTAL PROGRAM
`N1 G54 G17 G00;`	`N1 G54 G17 G90;`
`N2 G90 X0. Y0. F500;`	`N2 G00 X0. Y0.;`
`/N3 G68 X10 Y10 R45;`	`/N3 G68 X10 Y10 R45;`
`N4 G90 G00 X5. Y5.;`	`N4 G91 G00 X5. Y5.;`
`N5 G01 X15. F100;`	`N5 G01 X10 F100;`
`N6 Y15.;`	`N6 Y10;`
`N7 X5.;`	`N7 X-10;`
`N8 Y5.;`	`N8 Y-10;`
`N9 M30;`	`N9 G69;`
	`N10 M30;G54 G00;`

If optional block delete 1 is set “ON”, the control will cut the part shown with a dashed line in Figure 12.2. If optional block 1 is set “OFF” the control will cut the part shown with a solid line in Figure 12.2.

Figure 12.2
Results of Example 12.1



Note that in the preceding figure the center of rotation programmed in the G68 block is ignored when the block immediately following the G68 is an incremental motion block.

Angles and centers of rotation for G68 blocks are modal and remain in effect for following G68 blocks until a new center of rotation or angle is specified with a G68 command.

Important: It is possible to rotate all of the work coordinate systems at once by using the external part rotation.

If rotating the coordinate system again in the same plane using another G68 command:

- while in incremental mode, the angle of rotation is taken from the current rotated coordinate position (see Figure 12.3)
- while in absolute mode, the angle of rotation is taken from the original position

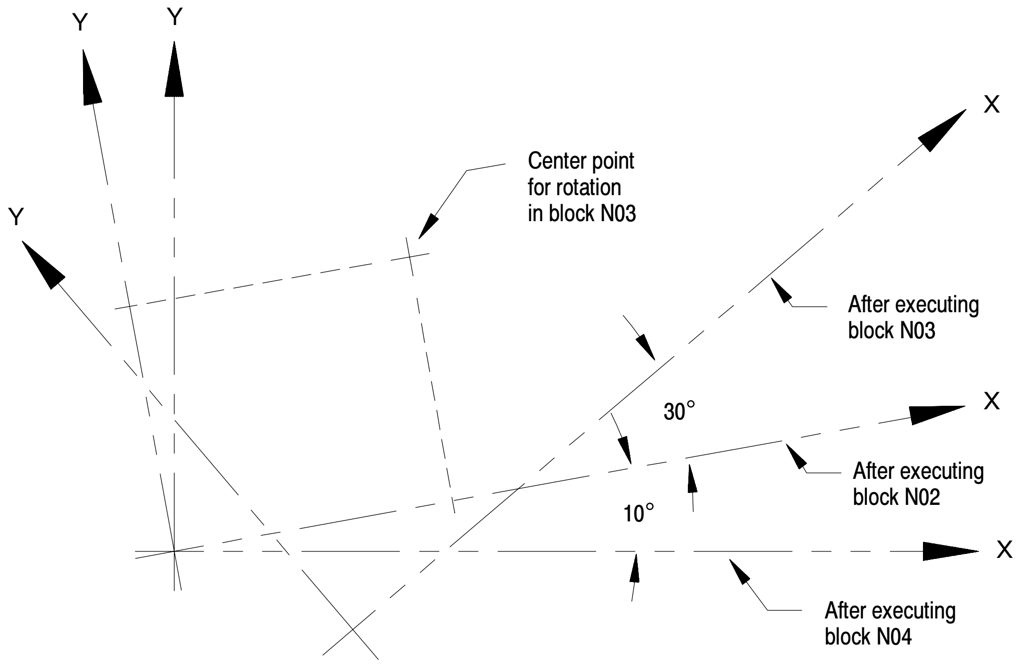
Rotating the coordinate system again in a different plane using another G68 is not allowed.

Executing a G69 cancels all G68 rotations and returns the coordinate system back to its original orientation. Local rotation of a work coordinate system using the G68 command is also canceled when the control executes an M30 or M02 code in a program.

Example 12.2
Multiple Rotation of the Coordinate System While in Incremental Mode

Program Block	Comment
N01 G54 G91;	Incremental mode
N02 G68X0Y0R10;	Rotates the current work coordinate system 10 degrees.
N03 G68X5.Y4.R30;	Rotates the current work coordinate system 30 degrees about a center point of X5., Y4. for a total rotation from its original position of 40 degrees.
N04 G69;	Returns the work coordinate system to its original position of 0 degrees.

Figure 12.3
Results of Example 12.2

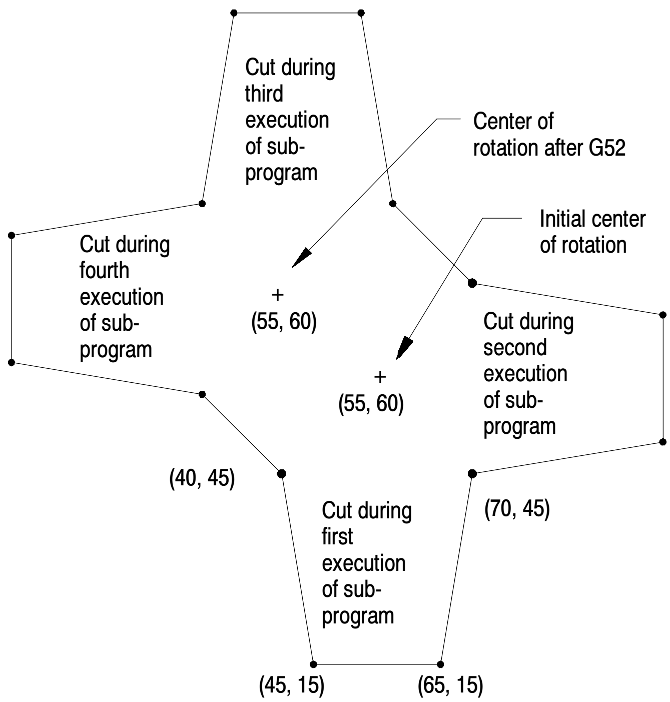


Rotating the work coordinate system can be helpful anytime a part has a repetitive shape. This feature combined with the G52 work coordinate system shift can reduce the size of a part program appreciably. The following program is an example of this.

Example 12.3
G68 Coordinate Rotation With G52 Coordinate System Shift

Main Program	Subprogram 1000
`G17 G90 G00 X0 Y0;`	`G01 X45. Y15. F500.;`
`G00 G90 X40 Y45.;`	`X65.;`
`M98 P1000 L4;`	`X70. Y45.;`
`M30;`	`G68 X55. Y60. R90.;`
	`M99;`

Figure 12.4
Results of Example 12.3



External Part Rotation

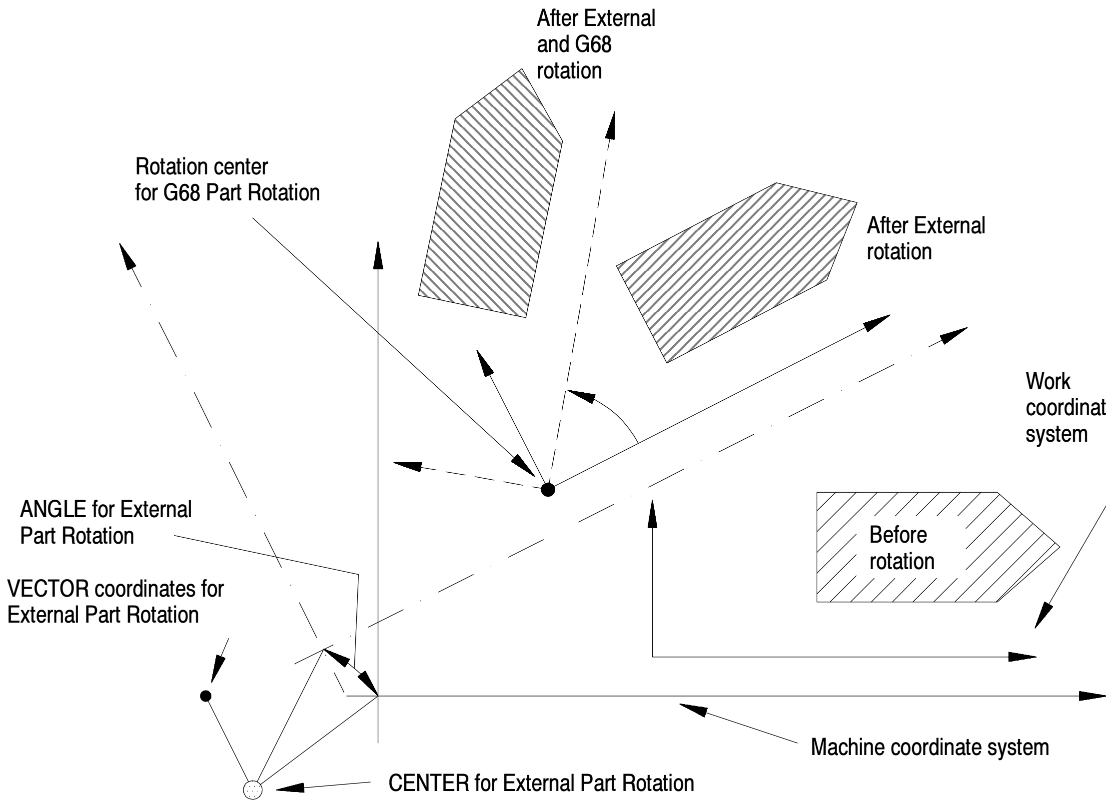
The external part rotation feature simulates a rotation of the machine coordinate system. Since all work coordinate systems are referenced from the machine coordinate system, rotating it would effectively rotate all work coordinate systems. However, software overtravels, programmable zone limits, homing, and positioning in the machine coordinate system are not affected.

When external part rotation is activated all work coordinate systems are rotated together by a specified amount, about a specified point.

External Part Rotation can be executed before or after rotation of the work coordinate system using the G68 command (as described on page 12-2). If a G68 is programmed to rotate the current work coordinate system, an additional rotation of coordinates will result as shown in Figure 12.5.

Any work coordinate system rotation that is to be done using the external rotation feature must be performed before program execution begins. Program execution may not be interrupted to perform a external part rotation. If an attempt is made to interrupt a program to perform an external part rotation the rotation will not become effective until the end of program (M02 or M30) command is read, a control reset, or E-Stop reset is performed.

Figure 12.5
External Part Rotation Followed By G68 Work Coordinate Rotation



Important: This feature only simulates rotation of the machine coordinate system for the purpose of establishing the preset work coordinate systems. Software overtravels, programmable zones, and any other positioning referenced on the machine coordinate system will remain unaffected by this rotation, but a program originally written within the limits can now be outside these limits.

Activating the External Part Rotation Feature

To activate the External Part Rotation feature, follow these steps:

1. Place the control in E-Stop and press the **{OFFSET}** softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
F2	F3	F4	F5	F6	F7	F8

		ERROR MESAGE	PASS-WORD	SWITCH LANG		→
F2	F3	F4	F5	F6	F7	F8

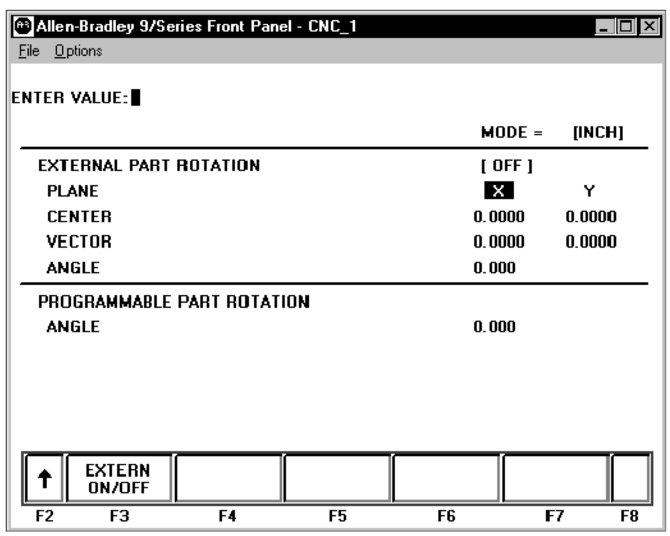
2. Press the **{COORD ROTATE}** softkey. This will display the external part rotation parameters screen as shown below.

(softkey level 2)

↑	WORK CO-ORD	TOOL WEAR	TOOL GEOMET	TOOL MANAGE	RANDOM TOOL	→
F2	F3	F4	F5	F6	F7	F8

↑	COORD ROTATE	BACKUP OFFSET	SCALNG			→
F2	F3	F4	F5	F6	F7	F8

Figure 12.6
Typical External Part Rotation Parameter Screen



3. Move the cursor to the desired parameter to be changed by pressing the up, down, left, right cursor keys. The selected parameter will be shown in reverse video.

4. Enter the new value for the desired parameter using the keys on the keyboard. The entered value will be shown on the input line (lines 2 and 3) of the CRT. The value may be modified on the input line as described on page 2-10.

5. When the desired value is displayed on the input line of the CRT, press the key. The value on the input line will replace the old parameter value.

6. Repeat until all parameters display the desired values.

7. Activate the external part rotation feature by pressing the **{EXTERN ON/OFF}** softkey. The external offset feature will toggle between on and off each time this softkey is pressed.

```
(softkey level 2)
```

↑	EXTERN ON/OFF					
F2	F3	F4	F5	F6	F7	F8

The work coordinate systems are all rotated as soon as the external rotation feature is activated. The current work coordinate system can be changed while an External Part Rotation is active. If changed, the new work coordinate system will be rotated as described by the External Part Rotation parameters.

The “PROGRAM” and “TARGET” position displays (as discussed in chapter 8) will not reflect an external part rotation since these values are relative to the active work coordinate system.

Since “ABSOLUTE” position displays always show coordinates relative to the machine coordinate system, they will reflect an external part rotation.

External Part Rotation Parameters

This table describes the parameters you use for external part rotation:

If you want to:	Use this parameter:	In this way:
select a plane	plane	Enter the axis names that define the plane to be rotated. Two separate values must be entered on this line. The first axis entered here is the axis that the angle of rotation is measured from. This parameter must match the active plane. If the external and program-selected planes do not match, the control generates a block decode error.
define the center of rotation	center	Enter a coordinate value for each axis in the selected plane. The center of rotation is a point on the machine coordinate system about which all the work coordinate systems will be rotated. The default value for the center of rotation is (0, 0).

define the vector	vector	Enter a coordinate value to define the “head” of the vector. The “tail” of this vector is the center of rotation. This parameter is optional. If you use the angle parameter, that value overrides this value. The resulting angle between this vector line and the first plane axis is the angle of rotation for the work coordinate system All work coordinate systems rotate about the point defined by the center parameter. The default values are (0, 0). With the values of this parameter, the control calculates the angle of rotation and displays this value in the angle parameter.
define the angle of rotation	angle	Enter the angle at which that you want to rotate the work coordinate system. This value overrides the value of the vector parameter. Enter a positive value for clockwise rotation from the primary axis of the selected plane. If you do not enter a value the control displays the angle defined by the vector parameter.

Plane Selection (G17, G18, and G19)

The control has a number of features that operate in specific planes. For that reason it is frequently necessary to change the active plane using a G17, G18, or G19.

Some of the features that are plane dependant are:

- Circular interpolation
- Cutter compensation
- Work Coordinate system rotation
- Many fixed cycle operations

Important: The system installer determines the planes defined by G17, G18, and G19 in AMP. Axes may not be assigned to the planes exactly as listed below. Refer to the documentation prepared by the system installer.

Typical axis names and their corresponding plane assignment are shown below:

- G17 -- plane defined by the X and Y axes (or axes parallel to X and Y)
- G18 -- plane defined by the Z and X axes (or axes parallel to Z and X)
- G19 -- plane defined by the Y and Z axes (or axes parallel to Y and Z)

Planes can be altered to accommodate additional axes parallel to the principle axes by programming those axes in the G17, G18, or G19 block. See Example 12.4.

Example 12.4
Altering Planes for Parallel Axes

Assuming the system installer has made the following assignments in AMP:

```
G17    -- the XY plane.
U axis -- parallel to X axis
V axis -- parallel to Y axis
```

Program block	Plane selected	Axis Motion
`G17;`	selects XY plane	None
`G17 U0;`	selects UY plane	U axis moves to zero
`G17 V0;`	selects XV plane	V axis moves to zero
`G17 U0V0;`	selects UV plane	U & V axes move to zero

Important: Any axis word in a block with plane select G-codes (G17, G18, G19) causes axis motion on that axis. If no value is specified with that axis word, the control assumes a value of zero or generates an error depending on how your system is AMPed.

Absolute/Incremental Modes (G90, G91)

There are two methods for programming axis positioning commands, absolute positioning and incremental positioning.

In the absolute mode, coordinates are referenced from the zero point of the active coordinate system. Absolute mode is established by programming a G90.

```
G90X40.Z20.;
```

In the above block the control will move the axes to a position X40, Z20 as referenced on the active coordinate system.

G90 is a modal G-code and remains active until cancelled by a G91.

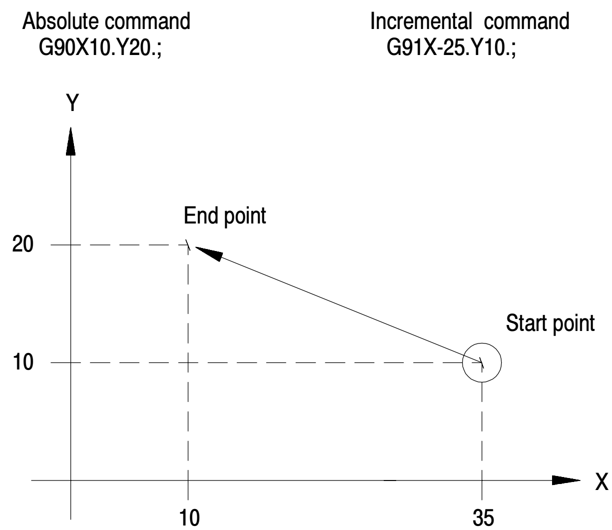
In the incremental mode, coordinates are referenced from the current axis position. Incremental mode is established by programming a G91.

```
G91X40.Z20.;
```

In the above block the control will move the cutting tool a distance of 40 units on the X axis and 20 units on the Y axis away from the current axis position.

G91 is a modal G-code and remains active until cancelled by a G90.

Figure 12.7
Incremental and Absolute Commands.



Inch/Metric Modes (G20, G21)

The selection of a unit system (inch or metric) can be done by programming either G20 for the inch system or G21 for the metric system. These unit system G-codes should be among the first blocks written in a program.

Both G20 and G21 are modal, and cancel each other. The default unit system selected by the control at power-up is determined in AMP by the system installer.

The currently active unit system is usually displayed on the screen for softkey level 1 in lines 3 or 4 between the [] symbols. If the screen selected for display of softkey level 1 is the status screen the active system G-code (G21 or G20) will be displayed among the active system G-codes.

These functions are affected by the active unit system (inch or metric).

- Position commands
- Feedrate commands
- Axis feed amount for fixed amount feed operation
- Unit system for hand pulse generator (HPG)

Scaling

Use the Scaling feature to reduce or enlarge a programmed shape. Enable this feature by programming a G14.1 block as shown below:

```
G14.1  X__Y__Z__P__;
```

Where :	Is :
X, Y, Z	the axis or axes to be scaled and the center of scaling for those axes
P	the scaling magnification factor for the specified axes.

The axes programmed in the G14.1 block determine which axes will be scaled. The corresponding axis word values specify the center of scaling for each axis. This position is the axis position around which the scaling operation is performed.

The scaling magnification factor (P) is the amount of scaling to apply to the programmed axes. Each scaled axis may have a different scale factor by programming them in separate G14.1 blocks. The scaling range is from 0.00001 to 999.99999. A scale factor less than one will reduce a programmed move while a scale factor greater than one will enlarge a programmed move.

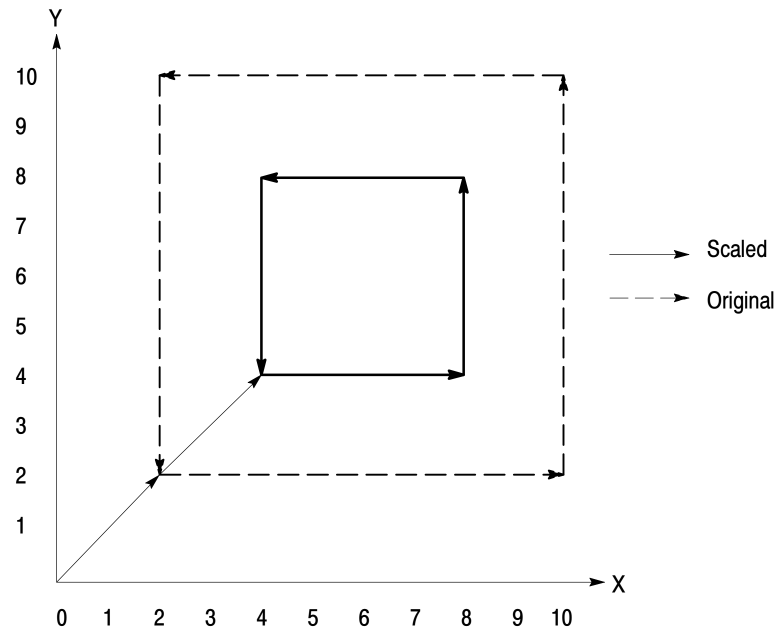
If no P-word is programmed or if P0 is programmed in the G14.1 block, the default magnification factor is used. If the programmed P-word value is out of range, an error message will be displayed on the CRT.

When absolute mode (G90) is active, scaling moves are referenced from the programmed center of scaling.

Example 12.5
Scaling with Absolute Mode Active

Program
N01 G14.1 X6 Y6 P0.5;
N02 G90 X2 Y2 F100;
N03 X10;
N04 Y10;
N05 X2;
N06 Y2;
N07 M30;

Figure 12.8
Results of Example 12.5



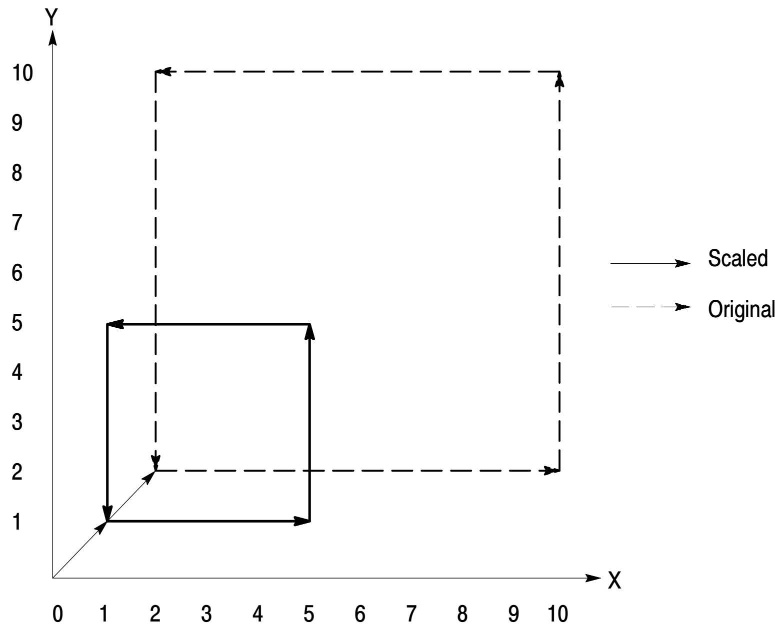
When incremental mode (G91) is active, the control ignores the programmed centers of scaling. The control performs scaling on the axes programmed in the G14.1 block, but the scaling moves are referenced from their current axis positions not the programmed center of scaling or the active coordinate zero point.

Important: The center of scaling may be specified in either incremental or absolute mode (G90/G91) in the G14.1 block. But, unlike other features in the control, both modes may not be programmed in the same block.

Example 12.6
Scaling with Incremental Mode Active

Program block	Comment
`N01 G91;`	incremental mode
`N02 G14.1 X6. Y6. P0.5;`	scale X and Y by .5 (X, Y values ignored)
`N03 G91 X2 Y2 F100;`	axis positioning move
`N04 X8.;`	feedrate move X
`N05 Y8.;`	feedrate move Y
`N06 X-8.;`	feedrate move X
`N07 Y-8.;`	feedrate move Y
`N08 G14;`	cancel scaling

Figure 12.9
Results of Example 12.6



G14 disables scaling on all axes. When scaling is disabled, the center of scaling and any scaling magnification factors are cleared. The next time scaling is enabled these values must be reset. In addition to G14, M99 in the main program, M02, M30, and a control reset operation will disable scaling. The system will power up with scaling disabled.

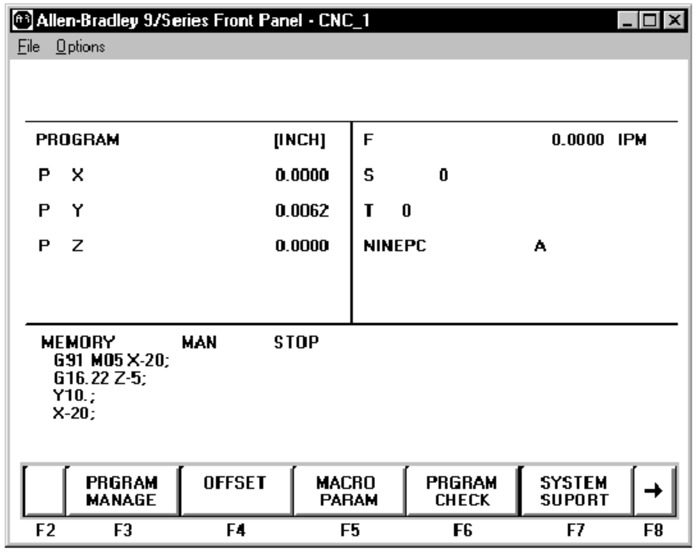
The system installer specifies in AMP, on an axis by axis basis, whether scaling is allowed. Refer to the literature provided by your system installer for additional information.

The control provides the logic program with the option of monitoring which axes are currently being scaled, on an axis by axis basis, through the logic flag BR_SCAX. Refer to your *9/PC Logic Reference Manual* for additional information.

Scaling and Axis Position Display Screens

When scaling is enabled for a particular axis, the letter "P" will be displayed next to the axis name on all axis position display screens. The following screen shows scaling enabled on all axes.

Figure 12.10
Axis Position Display Screen Showing Scaling Enabled



Scaling Magnification Data Screen

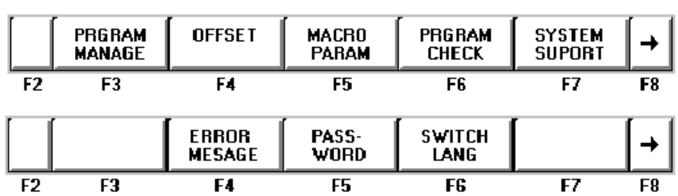
The scaling magnification data screen lists the currently active scaling magnification amount, the current center of scaling, and the default scaling magnification amount for all axes. The currently active scaling magnification amount and the current center of scaling for the axes can be monitored through this screen. The default scaling magnification amount for the axes can be monitored and/or changed through this screen.

The default scaling magnification values should only be changed when the control is in a stopped state. If the default values are changed, the new default values will not become active until the next G14.1 block is executed.

The scaling magnification data screen is accessed through these steps:

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

(softkey level 1)



2. Press the **{SCALNG}** softkey to display the scaling magnification data screen.

(softkey level 2)

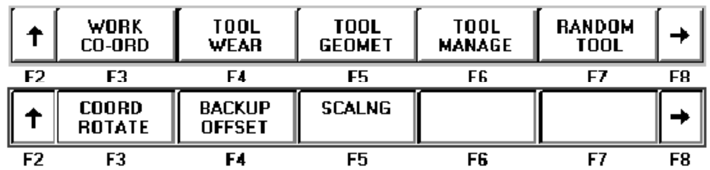
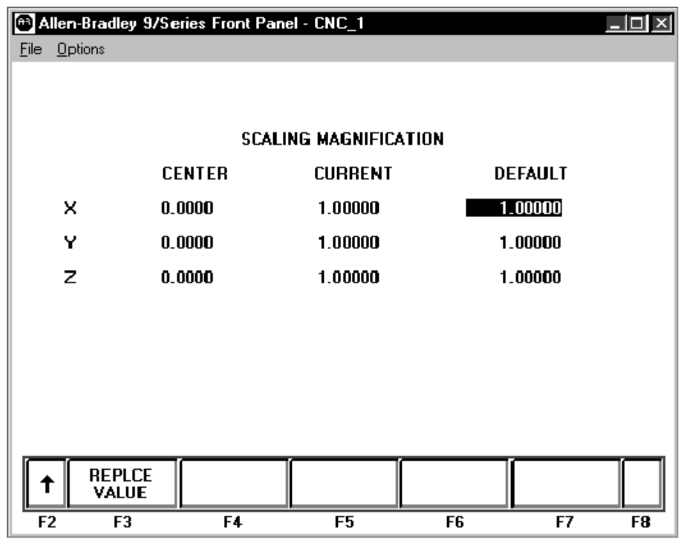


Figure 12.11
Scaling Magnification Data Screen



Important: If an axis is configured as a rotary axis, the scaling magnification display screen will display dashes instead of numbers for that axis. Rotary axes cannot be scaled.

The left column lists the current center of scaling for each axis. When scaling is cancelled, the current center of scaling for each axis is set to zero. The format of this value is determined by the word format of the selected axis.

The middle column lists the currently active scaling magnification value for each axis. When scaling is cancelled, the current scaling magnification value for each axis is set to 1.00000.

The right column lists the current scaling magnification default value for each axis. This value is used if P is not programmed or if P0 is programmed in the G14.1 block. The range of the default value is 0.00001 to 999.99999 with a word format of 3.5. The default values are stored in RAM or flash memory when the control is powered down. When the control is powered up, these values are restored.

3. Use the up or down cursor keys to move the block cursor to the default value to be changed. The selected default value will be shown in reverse video.

4. To replace stored default scaling magnification value, key-in the new default value and press the {REPLCE VALUE} softkey.

Scaling Restrictions

While scaling is enabled, these restrictions apply:

- Scaling affects only programmed axis motion. All manual axis motions and logic axis mover motions are performed at full scale
- Scaling does not affect M-, F-, S-, T-, and B-word functions. The F-word will be scaled if the control is in inverse time mode (G93). Scaling while in inverse time mode will be applied as follows:

```
                                          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 tool position in the old work coordinate system to the tool position in the new work coordinate system is not scaled
- Scaling is applied to G52 and G92 offsets. The center of scaling will be shifted when the work coordinate systems are shifted by a G92 offset or by changing coordinate offset values. When using a G52 offset, the center of scaling will be adjusted to the new local coordinate system
- Scaling is not applied to external offsets, tool wear, tool geometry, tool radius, or tool length offsets.
- Scaling will not be applied to blocks containing dwells (G04), data settinG-codes (G10, G10.1), or macro calls (G56, G66, G66.1). In the case of macro calls, the data passed via local parameters will not be scaled unless the data is used inside of the macro for motion.
- G22, programmable zone 2 check on and data setting, will not be scaled
- G53, absolute positions moves, will not scaled
- Rotary axes cannot be scaled
- Polar coordinates are not scaled

- In circular mode, the scale factors for the axes of the active plane have to be the same. The control generates an error if the scale factors of the axes are not equal
- Scaling will be applied to these fixed cycles as shown below:
  - G31, G31.1 - G31.4

    Gxx X__Y__Z__
    X (scaled)
    Y (scaled)
    Z (scaled)
  - G37, G37.1 - G37.4

    Gxx Z__
    Z (scaled)
  - G73, G74, G76, G82, G83, G84G85, G86, G87, G88, G89
- Gxx X__Y__Z__R__I__Q__K__P__F__L__
  X (scaled)
  Y (scaled)
  Z (scaled)
  R (scaled)
  I (not scaled)
  Q (not scaled)
  K (not scaled)
  P (not scaled)
  F (not scaled)
  L (not scaled)

Important: R uses the scale factor associated with the axis that is perpendicular to the active plane

- G38

  G38 H__R__D__E__F__
  H (scaled)
  R (scaled)
  D (scaled)
  E (not scaled)
  F (not scaled)

Important: The active plane scale factors must be equal. H, R, and D use the scale factor associated with the active plane

- G38.1

  G38.1 X__Y__I__J__R__D__E__F__
  X, Y (scaled)
  I (scaled)
  J (scaled)
  R (scaled)
  D (scaled)
  E (not scaled)
  F (not scaled)

Important: The active plane scale factors must be equal. R and D use the scale factor associated with the active plane

- G88.1, G88.2

  G88.x X_Y_Z_I_J_(,R or,C)_P_H_D_L_E_F_
  X, Y (scaled)
  Z (scaled)
  I, J (scaled)
  ,R ,C (scaled)
  P (not scaled)
  H (not scaled)
  D (scaled when scale factor is less than 1)
  (not scaled when scale factor is greater than or equal to 1)
  L (scaled when scale factor is less than 1)
  (not scaled when scale factor is greater than or equal 1)
  E (not scaled)
  F (not scaled)

Important: The active plane scale factors must be equal. ,R and ,C use the scale factor associated with the active plane

- G88.3, G88.4

  G88.x X_Y_Z_I_J_Q_(,R or,C)_P_H_D_L_E_F_
  X, Y (scaled)
  Z (scaled)
  I, J (scaled)
  Q (scaled)
  ,R ,C (scaled)
  P (not scaled)
  H (not scaled)
  D (scaled when scale factor is less than 1)
  (not scaled when scale factor is greater than or equal to 1)
  L (scaled when scale factor is less than 1)
  (not scaled when scale factor is greater than or equal 1)
  E (not scaled)
  F (not scaled)

Important: The active plane scale factors must be equal. Q, ,R, and ,C use the scale factor associated with the active plane:

- G88.5, G88.6

  G88.x X_Y_Z_R_Q_P_H_D_L_E_F_
  X, Y (scaled)
  Z (scaled)
  R (scaled)
  Q (not scaled)
  P (not scaled)
  H (not scaled)
  D (scaled when scale factor is less than 1)
  (not scaled when scale factor is greater than or equal to 1)
  L (scaled)
  E (not scaled)
  F (not scaled)

Important: The Irregular Pocket Milling Cycles feature (G89.1 and G89.2) is only available prior to system software release 12.xx. Any attempt to program a G89.1 or G89.2 in release 12.xx or later will result in the error message, “Illegal G-code”.

Important: The active plane scale factors must be equal. R uses the scale factor associated with the active plane. L uses the scale factor associated with the axis that is perpendicular to the active plane:

- G89.1, G89.2

  G89.x X_Y_Z_P_Q_H_E_F_L_
  - X, Y (scaled)
  - Z (scaled)
  - Q (not scaled)
  - P (not scaled)
  - H (not scaled)
  - E (not scaled)
  - F (not scaled)
  - L (not scaled)

END OF CHAPTER

Axis Motion

Chapter Overview

This chapter covers the group of G-words that generates axis motion or dwell data blocks. Major topics include:

Positioning Axes

Use these four basic G-codes to produce axis motion:

- G00 Rapid Positioning
- G01 Linear interpolation
- G02 Circular interpolation (clockwise)
- G03 Circular interpolation (counterclockwise)

After the execution of a positioning command the program proceeds to the next block only after an in-position check function confirms that all commanded axes have reached the in-position band. Your system installer sets the in-position band width in AMP. Refer to chapter 17 for details on the G-codes that you can use to modify the in-position band check.

Rapid Positioning Mode (G00)

The format for the rapid positioning mode is as follows:

```
G00X__ Y__ Z__ ;
```

Where :	Is :
`G00`	The G00 code establishes the positioning mode. In positioning mode, the cutting tool is fed along a straight line at the rapid feedrate determined in AMP by your system installer.
`XYZ`	The end point of the move generated by the G00 block in the current work coordinate system.

The G00 code establishes the positioning mode. In positioning mode, the cutting tool is fed along a straight line to a location designated by the programmed axis words.

The axes to be moved are determined by the axis names in the G00 block. The end point of the move to be generated is determined by the values programmed with the axis names.

Rapid positioning can be performed in the absolute mode (G90) or the incremental mode (G91).

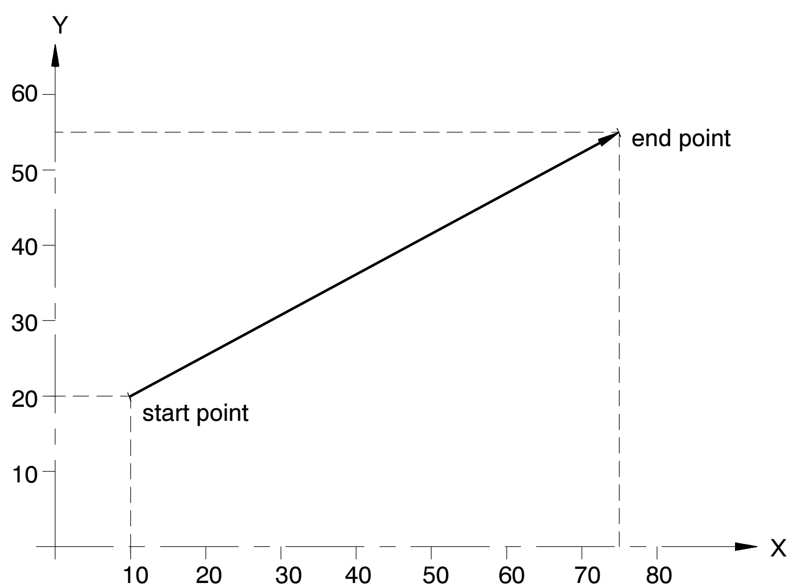
The system installer specifies a rapid feedrate individually for each axis in AMP. The feedrate of a positioning move that drives more than one axis is limited by the rapid rate set for the slower axis. The slower axis is driven at its rapid rate while the feedrate for other axes is reduced to maintain a linear move. This also assures that all axes start and stop at the same time.

G00 is a modal command and remains in effect until cancelled by a G-code of the same group. For a listing of G-code groups refer to Table 9.E.

Example 13.1
Positioning (G00)

Absolute Command	Incremental Command
`G90G00X75.Y55.;`	`G91G00X65.Y35.;`
`M30;`	`M30;`

Figure 13.1
Results of G00 Positioning Example



Important: Any F-word designated in the positioning mode is stored as the active feedrate in control memory but is ignored during positioning mode (G00).

Linear Interpolation Mode (G01)

The format for linear interpolation mode is:

```
G01X__ Y__ Z__ F__ ;
```

Where :	Is :
G01	G01 establishes the linear interpolation mode. In linear interpolation mode, the cutting tool is fed along a straight line at the currently programmed feedrate.
XZ	This is 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.

G01 establishes the linear interpolation mode. In linear interpolation mode, the cutting tool is fed along a straight line at the currently active or programmed feedrate.

The axes to be moved are determined by the axis names in the G01 block. The end point of the move to be generated is determined by the values programmed with the axis names.

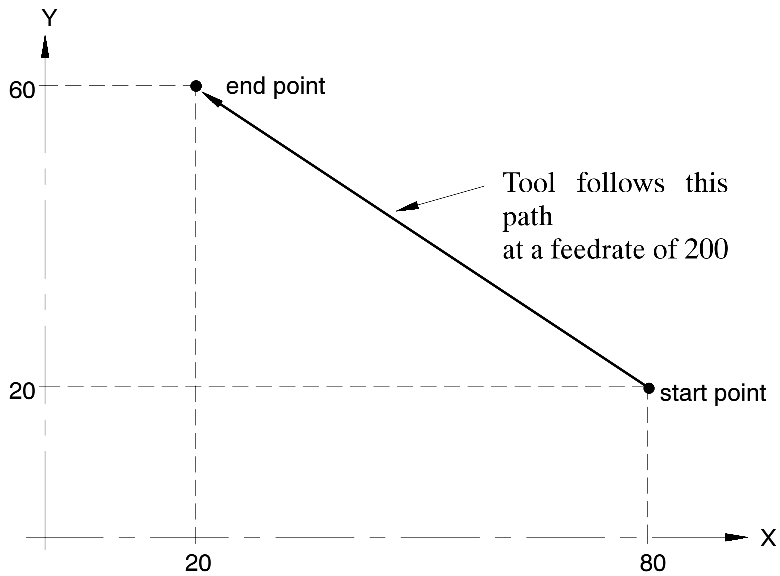
The F-word represents the feedrate for axis moves that take place in the G01, G02 and G03 modes. The F-word does not have to be programmed in the G01 block, however, if not programmed a feedrate must have been made active in some previous block.

Linear interpolation can be performed in the absolute mode (G90), or the incremental mode (G91).

Example 13.2
Absolute Versus Incremental Interpolation

Absolute Command	Incremental Command
G90G01X20.Y60.F200;	G91G01X-60.Y40.F200;
M30;	M30;

Figure 13.2
Results of Linear Interpolation (G01) Example



Once the feedrate, F, is programmed it remains effective until another feedrate is programmed (F is modal). It is possible to override programmed F-words. For information on overriding feedrates, see chapter 17.

Example 13.3
Modal Feedrates

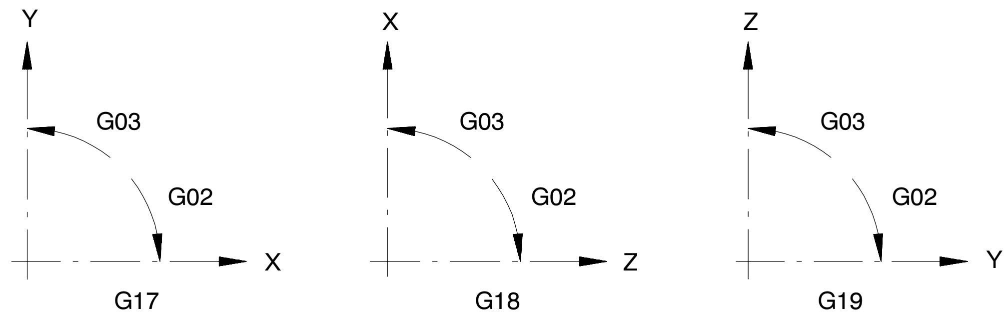
Program Block	Comment
G91G01X10.Y20.F200;	F200 is effective until another feedrate is programmed
Y35.;	
X40.Y35.;	
Y44.F50;	F50 is effective
M30;	

The feedrate for a multiaxis move is specified as the vectorial feedrate. The control will adjust the individual axis feeds to obtain the programmed feedrate. For information on feedrates, refer to chapter 17.

Circular Interpolation Mode (G02, G03)

G02 and G03 establish the circular interpolation mode. In G02 mode, the cutting tool moves along a clockwise arc; in G03 the tool moves along a counterclockwise arc. Figure 13.3 shows clockwise and counterclockwise orientation relative to the positive X, Y, and Z axes.

Figure 13.3
Circular Interpolation Direction



Circular interpolation can be performed in the absolute (G90) or incremental (G91) mode.

Important: S-Curve Acc/Dec mode is not available with circular interpolation mode.

A plane must first be established before the control will perform the correct arc.

The system installer selects a default plane that the control assumes when power is turned on, E-Stop is reset, or when the control is reset. In order to change planes, it is necessary to program either G17, G18, or G19. G17, G18, and G19 are modal and remain in effect until cancelled by each other. For details on plane selection, see chapter 12.

The system installer determines which axes are assigned to each plane in AMP. This manual assumes the axes are assigned to the planes as indicated below:

Circular Interpolation in XY plane

```
G17{G02} X__ Y__ {I__ J__} F__ ;
    G03             R__
```

Circular Interpolation in ZX plane

```
G18{G02} Z__ X__ {K__ I__} F__ ;
    G03             R__
```

Circular Interpolation in YZ plane

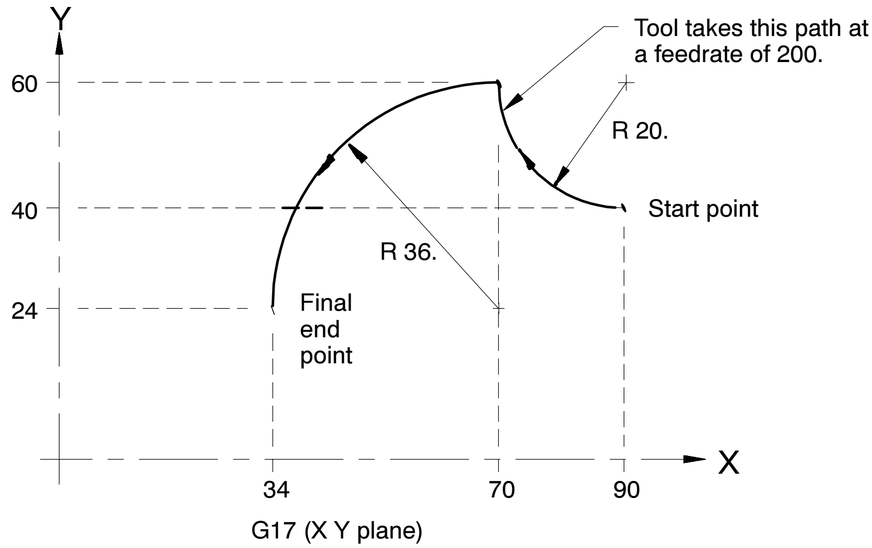
```
G19{G02} Y__ Z__ {J__ K__} F__ ;
    G03              R__
```

Where :	Is :
X, Y, Z	In absolute (G90) mode, these are the coordinate values of the end-point. In incremental (G91) mode, these are the positions of the end-point in reference to the start-point **CAUTION:** If no axis word is specified for the end-point of one of the axes in the currently defined plane, the control will assume a value for that axis equal to the current tool location. Specifying the wrong plane or wrong axis word for a circular move can inadvertently generate a helical move (refer to page 13-9).
I, J, K	These determine the position of the arc center in reference to the start-point. These values are always incremental regardless of the established positioning mode (absolute or incremental). I is parallel to X axis, J is parallel to Y axis, and K is parallel to Z axis in this manual. Refer to the system installer's documentation for integrand words.
R	Rather than defining a center with I, J, K, the option exists to define an arc radius using R. The sign of this entry determines the arc center-point location. If R is programmed as a positive value, the center-point will be located such that an arc less than or equal to 180° is generated. If R is programmed as a negative value, the center-point will be located such that an arc greater than 180° is generated. Refer to Figure 13.5 for an example.
F	Another option is to enter a feedrate tangent to the arc. If omitted, the control will use the feedrate active prior to this block.

Example 13.4
Circular Interpolation

Absolute Mode	Incremental Mode
G17;	G17;
G00X90Y40;	G91G02X-20.Y20.J20.F200;
G02X70.Y60.J20.F200;	G03X-36.Y-36.J-36.;
G03X34.Y24.J-36.;	M30;
M30;	
or	or
G17;	G17;
G00X90Y40;	G91G02X-20.Y20.R20.F200;
G90G02X70.Y60.R20.F200;	G03X-36.Y-36.R36;
G03X34.Y24.R36.;	M30;
M30;	

Figure 13.4
Results of Circular Interpolation Example

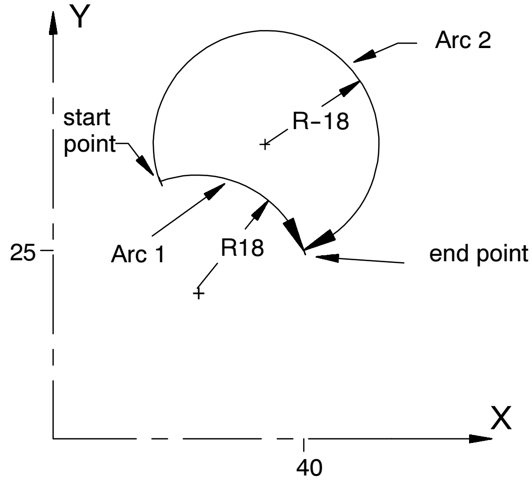


When programming an arc using the radius (R) value, two arcs are possible (Figure 13.5). Program the R word with a positive or negative value to distinguish between these arcs.

Example 13.5
Arc Programmed Using + or - Radius

Arc 1 **center angle less than 180 degrees**	**Arc 2** **center angle greater than 180 degrees**
G00X15Y30;	G00X15Y30;
G90G02X40.Y25.R18.F200;	G90G02X40.Y25.R-18.F200;
M30;	M30;

Figure 13.5
Results of Arc Programmed Using Radius Example



If the end point of the arc is not specified, or if the end point is the same as the start point, do not use R. Only J, I, and K can specify the center point in these cases.

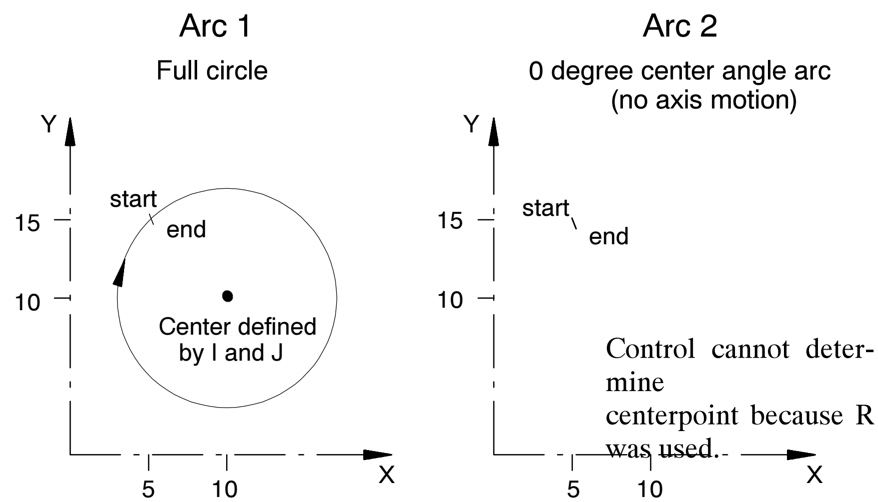
Important: Any axis in the current plane that is not specified when programming a circle defaults to the current axis position values. This results in the end point of an arc having the same coordinate value as the start point of the arc for that axis.

- If I, J, and/or K is used to program the arc center the control will cut a full circle.
- If R is used to program the radius of the arc the control will not move the axis. This is because the control defines an arc with a 0 degree center angle.

Example 13.6
Arc End Points Same As Start Points

Arc 1 - Full Circle	**Arc 2** - No Motion
G00X5.Y15;	G00X5.Y15;
G02X5.Y15.I5.J-5.F100;	G02X5.Y15.R7.07.F100;
M30;	M30;

Figure 13.6
Arc with End Point Equal To Start Point



If programming a radius command, R, in the same block as I, J, and/or K, the control gives the R priority. The I, J, and/or K words are then ignored.

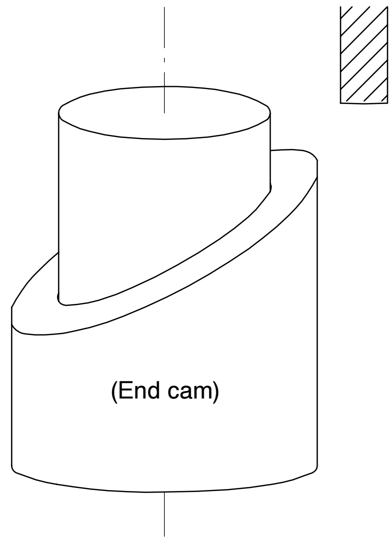
When programming I, J, and/or K words for the arc center, the words that have a zero value can be omitted.

Important: The system installer can specify the maximum allowed difference between the starting radius of the arc and the ending radius of the arc. If the difference exceeds the allowed value set in AMP, an error message occurs.

Helical Interpolation Mode (G02, G03)

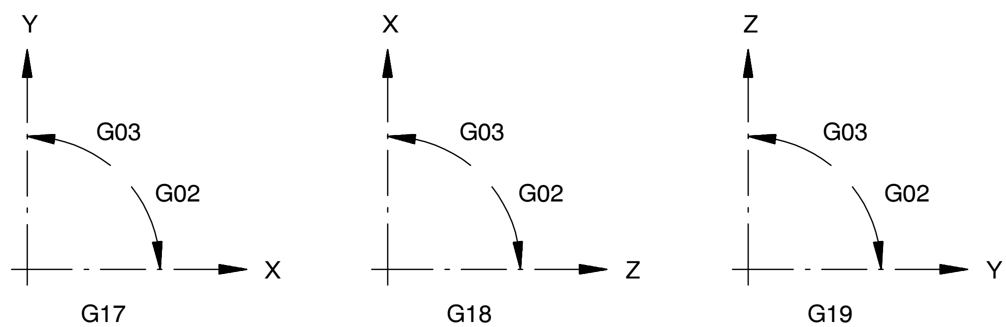
G02 or G03 may also be used to perform helical interpolation. Figure 13.7 shows how a part may be cut with helical interpolation.

Figure 13.7
Helical Interpolation (G02, G03)



Use G02 or G03 to add a third axis to the circular interpolation command block. The direction of the helical interpolation that results depends on whether a G02 or G03 was used. Refer to Figure 13.8.

Figure 13.8
Helical Interpolation Direction



Helical Interpolation in the XY Plane with the Z axis normal.

```
G17{G02} X__ Y__ Z__ {I__ J__} F__ ;
    G03                R__
```

Helical Interpolation in the XZ Plane with the Y axis normal.

```
G18{G02} X__ Z__ Y__ {I__ K__} F__ ;
    G03                R__
```

Helical Interpolation in the YZ Plane with the X axis normal.

```
G19{G02} Y__ Z__ X__ {J__ K__} F__ ;
    G03                R__
```

Where :	Is :
X, Y, Z	In absolute (G90) mode these are the coordinate values of the end point. In incremental (G91) mode these are the positions of the end point in reference to the start point The axis which is normal to the circular interpolation plane produces the "lead" of the helix. Again, all axes start and stop at the same time to produce helix motion.
I, J, K	These determine the position of the helix center in reference to the start point. These values are always incremental, regardless of the established positioning mode (absolute or incremental)
R	Rather than defining a center with I, J, K, the option exists to define an arc radius using R. The sign of this entry determines the arc centerpoint location. If R is programmed as a positive value, the centerpoint will be located such that an arc less than 180o is generated. If R is programmed as a negative value, the centerpoint will be located such that an arc greater than 180° is generated. Refer to Figure 13.5 for an example.
F	Another option is to enter a feedrate tangent to the tool path. If omitted the control will use the feedrate active prior to this block.

In helical interpolation, the feedrate is the same as in circular interpolation, that is, the feedrate is tangent to the tool path.

Important: Cutter diameter compensation is effective only for the arc portion of helical interpolation. Tool length offsets may be active during a helical move, however, changes to the tool length offset are allowed only if it does not affect either of the two circular axes in the move.

Positioning Rotary Axes

A rotary axis is a non-linear axis that typically rotates about a fixed point. A rotary axis is not the same as a spindle which uses an M19 to orient to a specific angle. A rotary axis is a fully positionable axis that is capable of interpolated motion when programmed in a block with other axes.

The system installer determines which axes are rotary axes in AMP, and determines the address that is used to command those axes. This manual assumes that the C-word is used to program a rotary axis. Refer to the system installer's documentation for the rotary axis words used in a specific system.

A rotary axis is programmed in units of degrees. This manual assumes that the system installer has configured the rotary axis to "rollover" at 359.99 degrees. Rollover means that after the rotary axis exceeds 359.99 degrees of rotation, its position displays rollover to 0 degrees and starts increasing. If the axis rotates to a position less than 0 degrees its position displays rollover to 359.99 degrees and starts decreasing.

Typically a rotary axis is programmed in a block by itself or with linear moves (rapid G00 or cutting G01 moves). If necessary it is possible, however, to program a rotary axis in a block that contains circular moves (G02 or G03).

Programming in Absolute or Incremental

Rotary axes may be programmed in absolute or incremental mode.

In absolute mode (G90), the rotary axis is programmed to angular positions. These positions are programmed between 0 and 359.99 degrees. The sign given to this angular position determines the direction that the rotary axis travels to reach the programmed angle. For example, programming:

```
G90C25;
```

in a part program causes the rotary axis C, to rotate to an angle of 25 degrees (referenced from a position 0 determined by your system installer) and rotate the axis in the positive direction to reach this position. Programming:

```
G90C-25;
```

in a part program causes the rotary axis C, to rotate to an angle of 25 degrees and rotate the axis in the negative direction to reach this position.

In incremental mode (G91), the rotary axis is programmed to move in 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 what direction the rotary axis rotates. For example, if the current C axis position is 25 degrees and this block is programmed:

```
G91C50;
```

the C axis would rotate 50 degrees in the positive direction. The new C axis position would be 75 degrees.

If the current C axis position is 25 degrees and this block is programmed:

```
G91C-50;
```

the C axis would rotate 50 degrees in the negative direction. The new C axis position would be 335 degrees.

In this mode:	you:
incremental (G91)	program a value greater than the rollover amount results in the rotary axis making one or more complete revolutions.
absolute (G90)	cannot program a rotary axis move greater than the rollover amount.
circular interpolation (G02 or G03)	cannot program a rotary axis move unless these conditions are met:: • the rotary axis cannot be in the active plane • the rotary axis must be programmed in the same block as a valid circular move made with the axes in the active plane

Important: You can program the largest move with a rotary axis is equal to the rollover amount. Any attempt to program a move that generates more motion than the rollover amount is truncated and moved to the position that has the same numerical endpoint as the programmed position. For example if this incremental move is programmed from a position of 10 degree:

```
G91C370;
```

the actual endpoint of the above move is still 20 degree; however, the rotary axis did not get there by revolving one revolution. Instead, it positioned itself directly to 20 degrees without passing 20 once as expected.

Determining Rotary Axis Feedrates

The feedrate for a rotary axis is determined in much the same way as linear axes.

When the control is in rapid mode (G00), the feedrate for the rotary axis is the rapid feedrate for that axis as set in AMP. Remember that if other axes are moving in the same block, the feedrate for the block is limited by the axis that takes the longest time to complete its programmed move at its rapid speed. (refer to chapter 17 for details).

When the control is in one of the cutting modes (G01, G02, or G03), the control uses the programmed feedrate to calculate the angular velocity of the rotary axis. This feedrate is still limited to the maximum cutting feedrate (feedrate clamp) as determined in AMP.

When you program in this mode:	The rotary feedrate units are in:
G94 feed per minute	degrees per minute.
G95 feed per revolution	degrees per revolution of the spindle.

In any event, if a rotary axis is programmed in a block with other axis moves in either rapid (G00) or cutting (G01, G02, or G03) modes, all axes reach their destinations at the same instant.

ATTENTION: When programming a rotary axis remember that the programmed feedrate is in units of angular velocity. This means that the actual cutting feedrate depends on the tools distance from the center of rotation of the rotary axis.

Cylindrical Interpolation

The cylindrical interpolation feature coordinates the motion of a rotary axis with the linear machine axes to machine contours on the side of a cylindrical workpiece as shown in Figure 13.9. Cylindrical interpolation mode is turned on using a G16.1 block and turned off with a G15 block.

A mill control with a minimum of two linear and one rotary axes is required for cylindrical interpolation. Typically there will be three linear and one rotary axis.

Cylindrical interpolation requires that axes be defined in AMP as a cylindrical interpolation rotary axis, a linear axis, a park axis, and a feed axis. The coordinates of the park and feed axes that define the rotary axis center-line must also be specified. Refer to the information provided by your system installer.

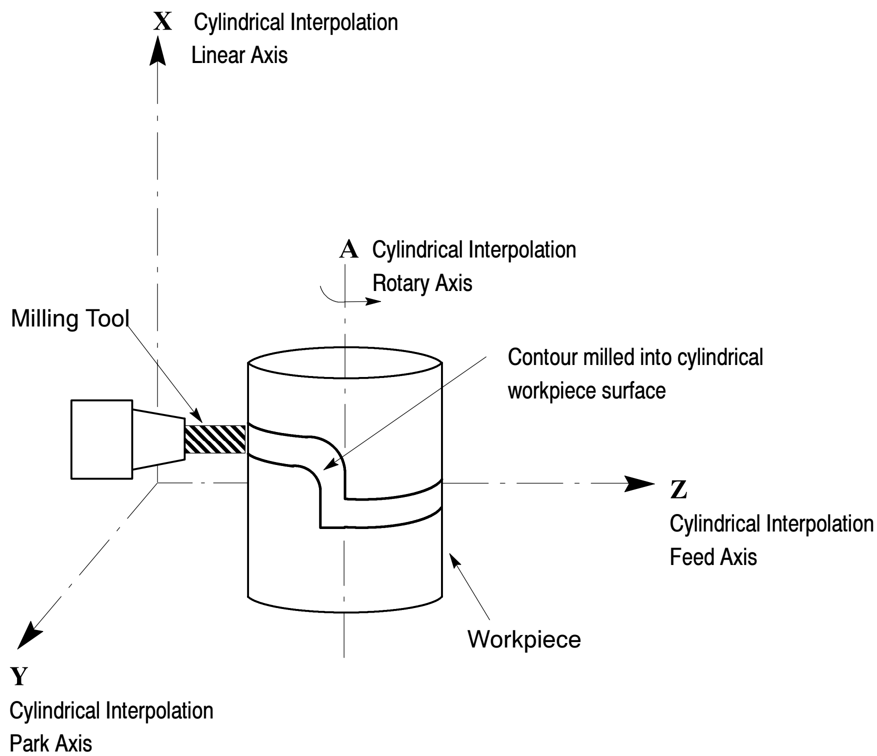
Important: Cylindrical interpolation requires that the cylindrical interpolation rotary axis rollover value be 360 degrees.

This discussion assumes the following AMP axis name assignments. Refer to the literature provided by your system installer for the axis names used by your machine.

- Feed axis is Z
- Park axis is Y
- Linear axis is X
- Rotary axis is A

Figure 13.9 shows a typical mill configuration for cylindrical interpolation.

Figure 13.9
Typical Mill Configuration for Cylindrical Interpolation



Important: The center of the rotary axis must coincide with the center-line of the workpiece on which contours are machined during cylindrical interpolation.

Cylindrical Interpolation Block Format

The block used to activate cylindrical interpolation has the following format:

```
G16.1 R__ X__ Z__ A__ F__
```

Where :	Is :
R	The radius at which the feed axis (typically the Z axis) will be positioned at the start of cylindrical interpolation. Can be used to alter the feed axis depth if programmed in a G16.1 block during cylindrical interpolation.
X	The coordinate (if in G90 absolute mode) or the linear distance (if in G91 incremental mode) to which the X axis is to move.
Z	The coordinate (if in G90 absolute mode) or the linear distance (if in G91 incremental mode) to which the Z axis (feed axis) is to move.
A	The angular coordinate (if in G90 absolute mode) or the angular distance (if in G91 incremental mode) to which the A rotary axis is to move.
F	The feedrate to be used by the X and Z axes when commanded to move while G16.1 is active. It also controls the A rotary axis speed as. Refer to chapter 17.

These parameters and their application are described in detail in the paragraphs that follow:

Important: R must be programmed in the initial G16.1 block. If R is not programmed in the initial G16.1 block, the error message “CYLINDER RADIUS IS ZERO” will appear. At power turn-on, program-end (M02, M30, or M99) or control reset the cylindrical interpolation feature is turned off and the R value is set to zero. It must then be re-entered in the next G16.1 block.

The radius specified by the R parameter is modal and does not need to be included in subsequent cylindrical interpolation blocks. Programming a G16.1 block with a different R value will modify the feed depth to the new radius. Feed depths cannot be changed using the Z parameter when G16.1 is active. Programming a Z will generate the error message “FEED AXIS MOTION NOT ALLOWED”.

Figure 13.9 illustrates the tool position if the AMP parameter **Feed Axis Park Location** is selected as “Nearest to Machine Zero”. If “Farthest from Machine Zero” were selected, then the tool would be positioned for cutting into the part from the positive side of the Z axis. Refer to the information provided by your system installer.

An A or X axis position may be programmed with the R parameter in the initial G16.1 block. However, once G16.1 mode is established, only the X parameter can be programmed in the same block as the R parameter. When it is, the X axis motion will be executed first followed by feed axis motion to radius R.

If an A axis position is programmed, the A axis will be rotated to the specified angle. If the A and X axes are programmed together in the same block, then a vector motion will result. around the circumference of the part.

If G02 or G03 circular interpolation is made active while in G16.1 cylindrical interpolation mode, a circular cut can be made around the circumference of the part (such as the contour cut in Figure 13.9). This is accomplished by programming the A and X axis endpoints along with the desired circle radius R as described on page 13-5. Note that the R parameter now defines the radius of the circular path to be cut, not the feed axis position.

Important: When programming circular interpolation in G16.1 mode, only radius programming (using R) may be used. Integrand programming (using I, J, K) is not allowed and will generate the error message “CIRCLE PROGRAMMING ERROR”. Refer to page 13-5.

Important: A axis motion is programmed as an angular value. When programming circular interpolation in G16.1 mode, this angular value has to be derived from an A axis arc length (based on the cutting radius). Refer to Example 13.7.

To perform G02/G03 circular interpolation while in G16.1 mode, the linear axis (X) and the rotary axis (A) must move to the endpoint of the arc of radius R made on the side of the cylinder.

In incremental mode (G91) the A axis arc length along with the programmed X move length, must position the A and X axes at a legal endpoint for the arc radius defined by the R value in the G02/G03 block.

In absolute mode (G90) the coordinate defined by the A axis arc along with the coordinate programmed for the X axis, must position the A and X axes at a legal endpoint for the arc radius defined by the R value in the G02/G03 block.

When cylindrical interpolation is activated, the circle plane is set to XA. The A and X axes become the two axes of the circle plane and remain so, as long as the G16.1 mode is active. If the active plane is changed, the change will not become effective until the G16.1 mode is cancelled, and will be superceded if the G16.1 plane is reactivated.

Canceling Cylindrical Interpolation

Cancel cylindrical interpolation by programming a G15. The G15 program block can not contain axis words. Note that the G15 program block can cancel other modal group 15 functions such as polar programming (refer to appendix C for a complete listing of modal group 15 G-codes).

Cylindrical Interpolation Operation

When cylindrical interpolation is activated, the control will position the tool on the cylindrical work surface with two distinct moves. In the first move, all programmed axis moves in the initial G16.1 block (including the A axis) will be executed. At the same time, the park axis (Y) is positioned to the park axis coordinate as specified in AMP (refer to the documentation provided by your system installer).

Once the tool is positioned at the AMP specified park coordinate, the control locks or “parks” the park axis at it’s current position. This prevents additional commands from moving the tool off the rotary axis center-line. This first move takes place at the rapid feedrate for the axes.

In the second move, the feed axis (Z) moves at the active cutting feedrate to the radius specified by R.

The blocks following the G16.1 block determine the contour to be machined on the side of the cylindrical workpiece. The moves of the rotary axis (A) and the machine axes are interpolated to produce the programmed contours.

The following example makes a series of circumferential and circular cuts into the side of a cylindrical workpiece. The A axis angle in the G02 block of this program was derived from the equation that follows this example. Figure 13.10 illustrates the results.

Example 13.7
Cylindrical Interpolation Example

Program Block	Comment
`N01 G01 X0 Y0 Z0 F100;`	
`N02 G16.1 X100 R100;`	set cylindrical cutting radius at 100
`N03 G01 A40;`	make circumferential cut of 40 degrees
`N04 G02 X80 A51.459 R20;`	make arc cut of radius 20
`N05 G01 X60;`	make linear cut of 60
`N06 G16.1 R90;`	change cylindrical cutting radius to 90
`N07 G01 A160;`	make circumferential cut of 160 degrees
`N08 G01 X100 A270;`	make linear and circumferential cut
`N09 G01 A0`	finish with circumferential cut
`N10 G15;`	cancel cylindrical interpolation
`N11 M30;`	

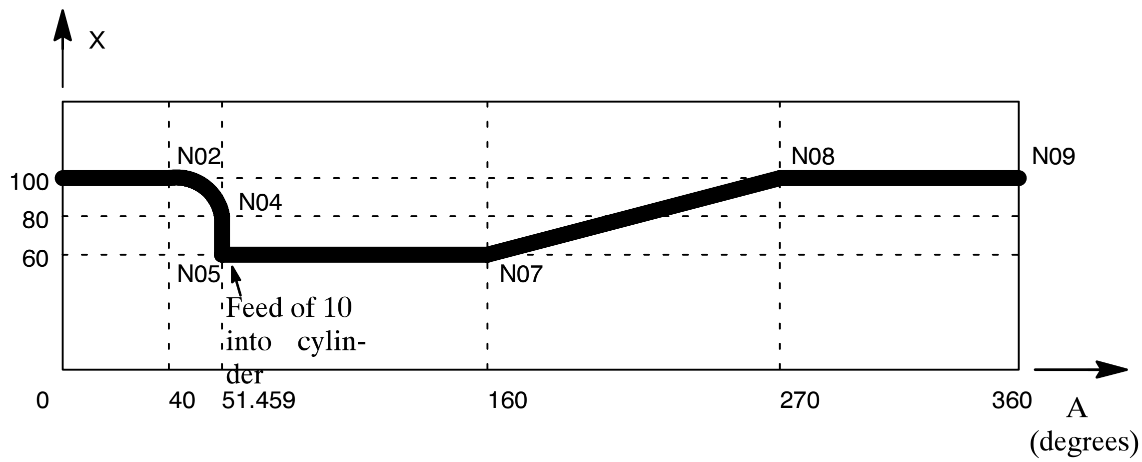
The angle for the A move in the G02 block above was determined using the following equation, with L = 20 and R = 100.

$$\theta = \frac{360\,(L)}{2\,\pi\,(R)}$$

Where :	Is :
θ	The angle to be programmed for the A axis.
L	The length of the arc along the circumference of the cylinder, as required to define a legal endpoint for the arc programmed in the G02/G03 block.
R	The radius at which the feed axis is positioned. This is the active R value programmed in the initial G16.1 block, not the R radius for the G02/G03 block.

Figure 13.10 illustrates the results of Example 13.7 with a two-dimensional plot of the circumferential tool path.

Figure 13.10
Results of Cylindrical Interpolation, Example 13.7



Cylindrical Interpolation Programming Restrictions

When the cylindrical interpolation feature is enabled the following programming restrictions apply:

- Work coordinate system offsets (G52, G54-G59, and G92) for the park and feed axes (Y and Z) will be temporarily cancelled when in G16.1 mode. Offsets for other axes will not be affected
- Tool offsets and cutter compensation/TTRC offsets are allowed on the cylindrical circle plane during cylindrical interpolation programming
- Activation of offsets through logic is disabled
- Jog on the fly is disabled
- Cylindrical interpolation cannot be activated during part rotation G88/G89
- Integrand circle/arc programming is not permitted during G02/G03 blocks. Only direct radius (R) programming is allowed
- Automatic motion to and from home G27, G28, G29, or G30 is not permitted.
- Work coordinate changes and shifts G53, G54-G59, G59.1, G59.2, G59.3, G50/G92, G52, G92.1 are not permitted
- Cavity and Irregular pocket cycles cannot be used
- Only the primary spindle (selected via G12.1) may be used in coordination with cylindrical interpolation. If the system uses Auxiliary Spindle(s), and the auxiliary spindle is the controlling spindle when cylindrical interpolation is selected, an error message appears.

Logic Axis Mover

Your system installer has the option of controlling selected axes through the logic program. When an axis is under logic control, the operator and part program have no control on that axis. Jog commands, as well as part program commands, are typically ignored unless logic has been written to manipulate these values in some manner.

Be aware that it is possible to disable axis position displays on the BDS for an axis under logic control. Refer to the documentation provided by your system installer for details on an axis controlled by logic.

Important: S-Curve Acc/Dec mode is not available with Logic Axis Mover.

Polar Coordinate Programming (G15, G16)

Polar programming allows a programmer to use polar coordinates (using angles and distance specified with a radius) as a means of establishing the end point of a move rather then specifying the normal cartesian coordinates of the end point. G16 and G15 are modal G-codes used to start and stop polar coordinate programming respectively.

After a G16 block in a part program the control will interpret the axis words as polar programming commands. Cancel polar programming with a G15 block in a part program. The G15 program block can not contain any axis words. Note that the G15 program block can cancel other modal group 15 functions such as cylindrical interpolation (refer to appendix C for a complete listing of modal group 15 G-codes).

Axis words in the current plane (selected by G17, G18, and G19) are used to program angle and radius values. The order in which they are assigned to a plane in AMP is significant in determining their use.

Specifying the Radius:

The first axis word that is used to describe the current plane is used to specify radius values. Negative radius designations are measured 180 degrees from the current angle designation.

Specifying the Angle:

The second axis word that is used (to describe the current plane) is used to specify angle designations. The angle is specified in units of degrees. Positive angles are measured counter clockwise and negative angles are measured clockwise.

For example, if the current plane is G17 (defined in AMP to be the X, Y plane) during polar programming:

Any X word in a program is used as a radius value for all following moves until re-specified using another X word or polar programming is cancelled with a G15 block.

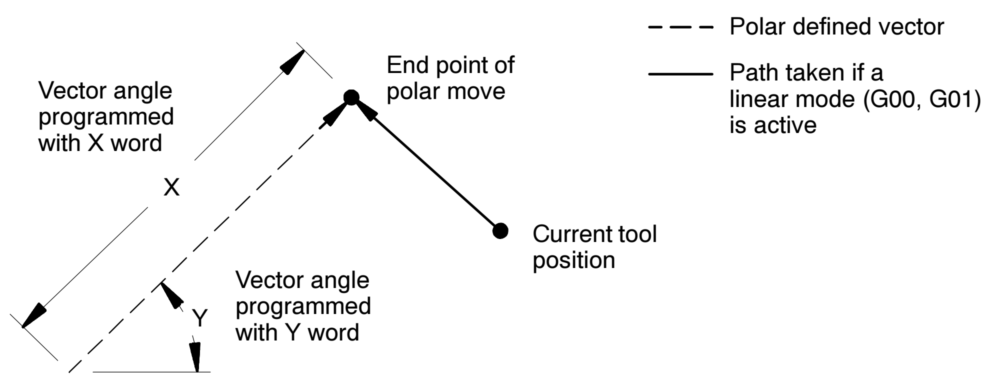
Any Y word in a program is interpreted to be the angle for all following moves until re-specified using another Y word or polar programming is cancelled with a G15 block.

For the purpose of explanation this section assumes that the X,Y plane (G17) is the currently active plane for all polar programming examples and figures. Any other axis word that is specified in a polar block and is not in the current plane is interpreted by the control as a normal cartesian coordinate value.

Polar positioning is done by defining a vector using a radius and angle value. The head (or end) of the vector defined by the radius and angle values is used as the end point of a polar move.

In both incremental and absolute mode the cutting tool will follow a path starting at the end point of the last move and ending at the head of the vector defined by the radius and angle. How the tool reaches that endpoint is determined by the current positioning mode (G00, G01, G02, or G03).

Figure 13.11
Polar Vector Defining End Point Of Tool Path



The control's interpretation of the specified angle and radius values used to define the vector is dependant on whether programming is in incremental mode or absolute mode. It is possible to mix incremental and absolute modes in polar programming, though for clarity this practice is not recommended.

If programming in incremental mode (G91):

- The radius is measured from the current tool position at the specified angle to define a vector.
- The angle is referenced relative to the last programmed angle (note the last programmed angle defaults to zero degrees upon entry to polar programming mode).

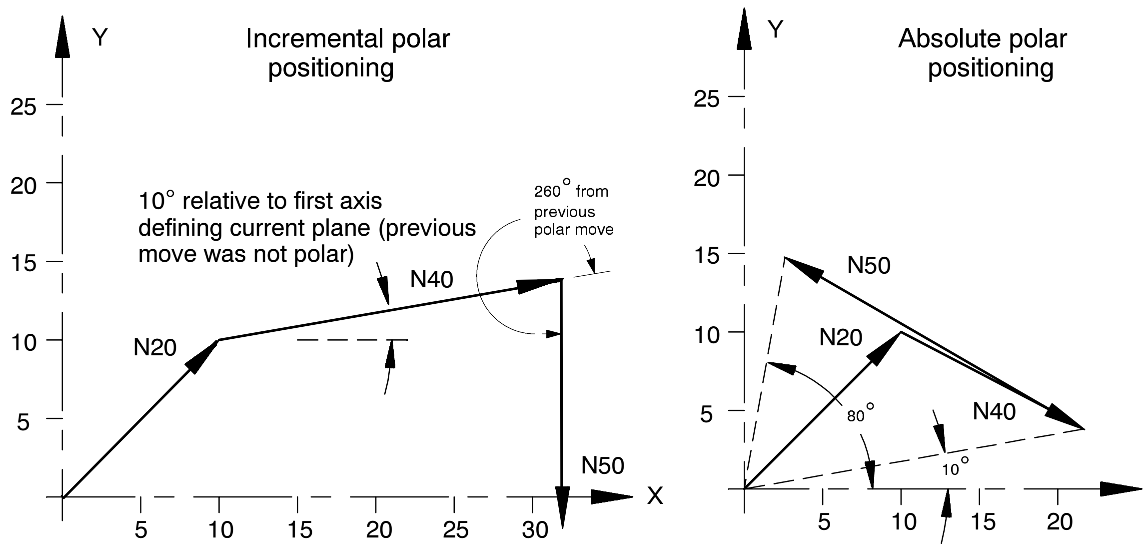
If programming in absolute mode (G90):

- The radius is measured from the zero point of the currently active work coordinate system at the specified angle and defines a vector. This vector is independent of the current tool position.
- The angle is referenced from the first axis that is used to define the currently active plane and is independent of the previous move.

Example 13.8
Incremental Versus Absolute Polar Blocks

Incremental	Absolute
N10 G00 X0 Y0 F150;	N10 G00 X0 Y0 F150;
N20 G91 G01 X10 Y10;	N20 G90;
N30 G16;	N30 G01 X10 Y10;
N40 X22 Y10;	N40 G16;
N50 X15 Y260;	N50 X22 Y10;
N60 G15;	N60 X15 Y80;
N70 M30;	N70 G15;
	N80 M30;

Figure 13.12
Results of Incremental Versus Absolute Polar Programming Example



Angles may be entered in a polar block with positive or negative values. Angles are referenced counter-clockwise if specified as positive and clockwise if negative. Clockwise and counterclockwise orientation for the X, Y, and Z axes is shown in Figure 13.3.

Angle values greater than 360 degrees are permitted. Programming 365 degrees or 725 degrees will have the same result as if 5 degrees were programmed.

Radius values may be programmed as positive or negative values. When specifying a radius value as a negative amount it is referenced in a direction 180 degrees from the currently specified angle.

The axis position displays will not show the polar coordinate values during polar programming. These displays will always show the current cartesian coordinate position as described on page 8-1.

Important: Polar programming mode has no effect on axis words that are programmed with any of the following G-codes. Any axis words in any of these blocks are executed as if polar programming was not active. Axis words specified in these blocks have no effect on the current angle or radius active in polar programming.

Coordinate system offset G52
Coordinate system offset G92
Work coordinate system rotation G68
Dwell G04
G10 blocks that modify tables
Programmable zone blocks G22
Programmable mirror image G50.1
Motion in the machine coordinate system G53

Though polar programming has no effect on programming these G-codes, many of these G-codes have a significant impact on the execution of polar programming.

For example a G68 work coordinate system rotation will have a major impact on angles specified in a polar programming block. When the coordinate system is rotated the polar angle is referenced from the rotated axis. Work coordinate system offsets will have a similar effect on the work coordinate system zero point thus moving the location of the radius vector start point.

Axis words that are programmed with any of the return to/from home G-codes (G27 - G30) to specify an intermediate point, are interpreted as polar coordinate values when in polar programming mode.

Polar Programming Special Cases

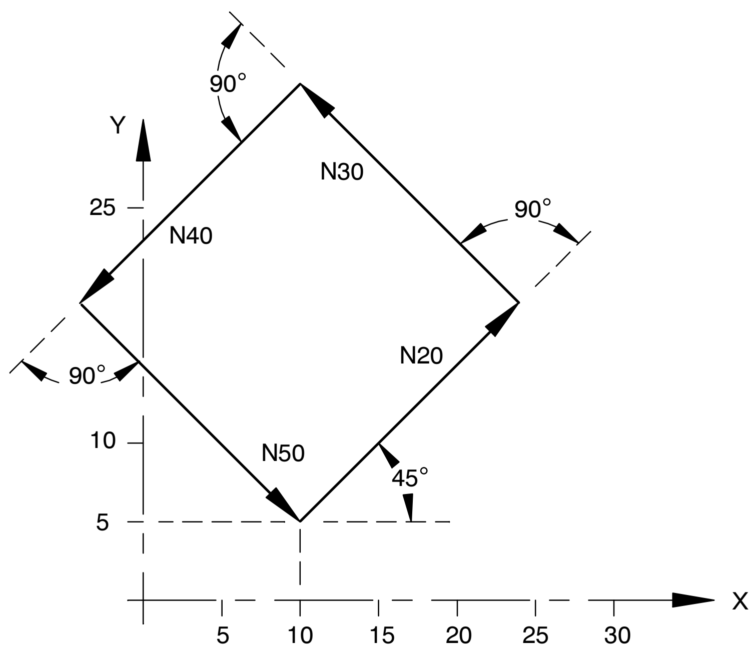
When programming using polar blocks the values programmed with the axis words are stored much as if they had been position commands. Normally, programming an incremental move of Y1.3 would position the Y axis 1.3 units from its previous position. The X axis position would not change. This also holds true for polar programming.

Programming Y20. with polar programming active specifies an angle of 20 degrees. This angle remains in effect for all subsequent blocks until a new value for Y is specified. This is also true for radius specifications. Keep in mind, however, that either a polar distance, a polar angle, or both must be programmed in the polar block.

Example 13.9
Polar Blocks with only Polar Angle Programmed

G00X10Y5;		G00X0Y0;
N10G00X10Y5;		G01G91G16F100;
N20G01G91G16F100;		N10G00X0Y0;
N30X20Y45;		N20G01G91G16F100;
N40Y90;	or	N30X20Y45;
N50Y90;		N40Y90;
N60Y90;		N50X20;
N70Y90;		N60X20;
		N70M30;

Figure 13.13
Results of Example 13.9, Polar Programming Blocks



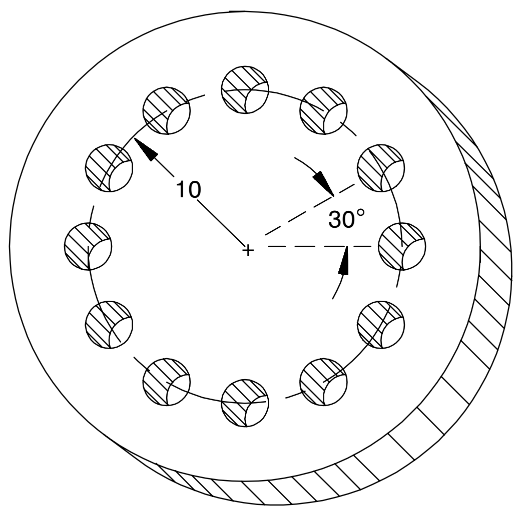
It is possible to change from incremental to absolute or absolute to incremental modes during polar programming if desired. The axis word is interpreted by the control in the mode that it was specified in. Mixed combinations such as angles designated in absolute and radii designated in incremental are possible.

Example 13.10 is used to illustrate this.

Example 13.10
Changing Between Incremental and Absolute During Polar Moves

```
N10G01X0Y0Z0F100;
N20G16;
G90X10.Y0.;
G81G91Y30.Z10.R5.L12;
G15;
M30;
```

Figure 13.14
Bolt Hole Pattern, Results of Example 13.10



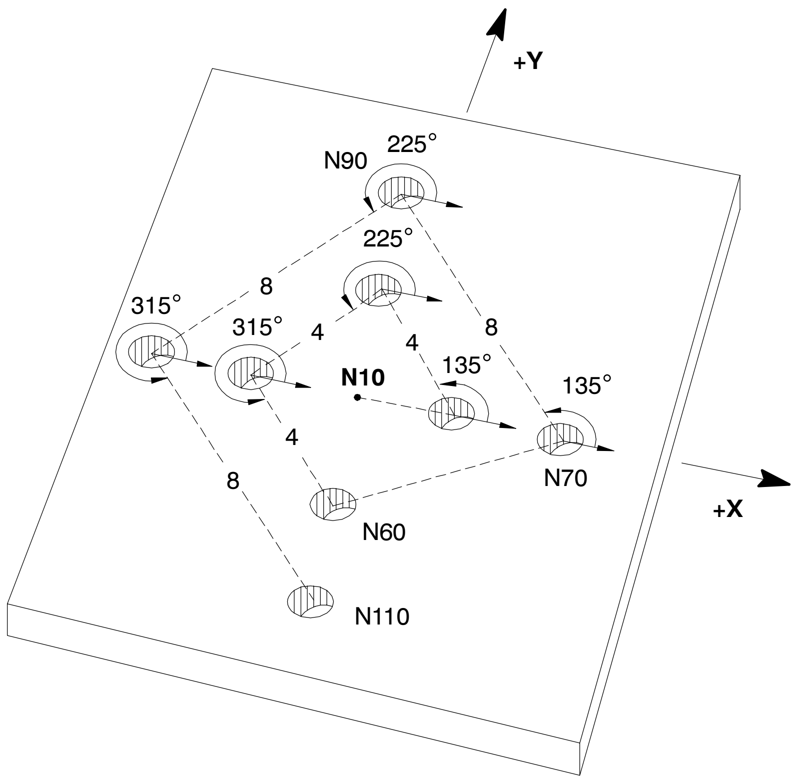
When programming an arc using polar programming the angle and radius values still define the end point of the next move. The center point of the arc must still be designated in the circular block using either the I, J, or K words or by designating a radius value (refer to page 13-5 for information about circular interpolation).

It is also possible to use polar programming when the angles are programmed in absolute mode and the radii are in incremental. See Example 13.11 and Figure 13.15.

Example 13.11
Polar Programming - Angle in Absolute, Radii in Incremental

N10 G00 X0Y0 F500;	rapid move to X0 Y0
N20 G90 G81 X3.Y0 R3. Z10.;	drilling cycle at X3 Y0
N30 G16;	polar programming
N40 G91 X4. G90 Y135.;	radius of 4 at 135 deg abs
N50 Y225.;	still radius of 4 at 225 deg abs
N60 Y315.;	still radius of 4 at 315 deg abs
N70 G15 X6. Y0	cancel polar, move to X6 Y0
N80 G16;	polar programming
N90 G91 X8. G90 Y135.;	radius of 8 at 135 deg abs.
N100 Y225.;	still radius of 8 at 225 abs.
N110 Y315.;	still radius of 8 at 315 abs.
N120 M30;	end

Figure 13.15
Results of Example 13.11

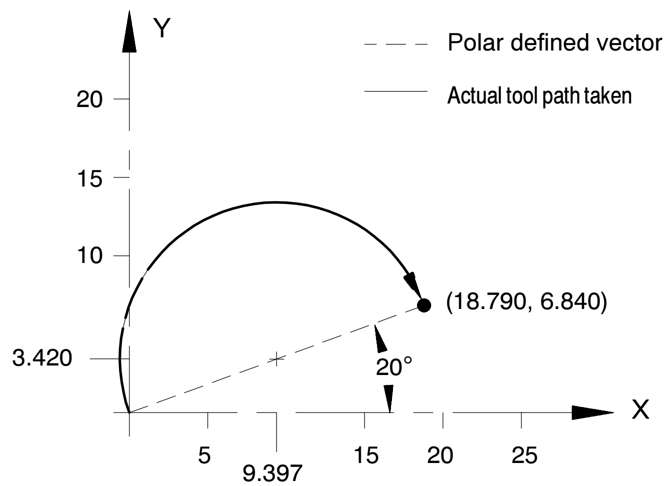


When programming an arc using I, J, or K words the control does not use these values as polar coordinates. Program the center of the arc in the same manner as normal circular programming described on page 13-5. I, J, and K are always cartesian coordinate values.

Example 13.12
Circular Polar Programming

```
G00X0.Y0.;
G91G16F100;
G02X20.Y20.I9.397J3.42;
G15;
M30;
```

Figure 13.16
Results of Circular Polar Programming Example



Automatic Motion to and from Machine Home

Machine tools have a fixed machine home position that is used to establish the coordinate systems. The 9/PC control offers two methods for homing a machine after power up.

Operation:	Description:
Manual machine home	uses switches or buttons on the MTB panel provided solely for this purpose. Manual homing is described in detail in chapter 4.
Automatic machine home	uses a programmed machine home code.

Automatic Machine Homing (G28)

You accomplish automatic homing by the using 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 4.

The coordinate values that 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 two consecutive markers on the DCM scale.

   Important: To determine an absolute position using DCMs, you must encounter at least two 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”.

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 positions 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.

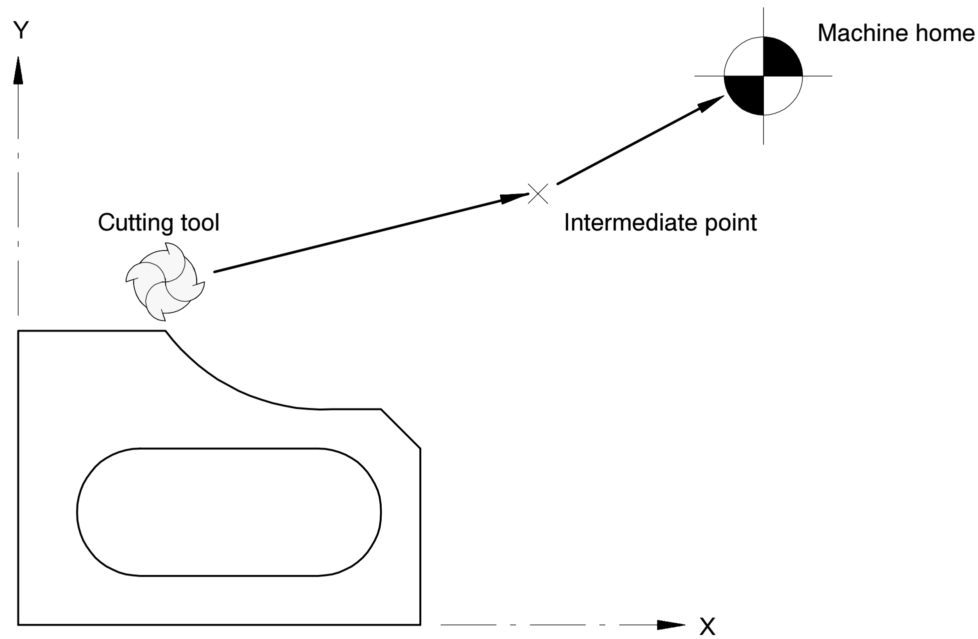
The return operation generates two axis moves both executed at the rapid feedrate. The first move is to the intermediate point, and the second is to the axis home position.

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 13.17
Automatic Return to Machine Home (G28)



Usually a G28 is followed by a G29 (automatic return from machine home) in a part program; however, the control 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 tool offsets and cutter compensation during the axis move to the intermediate point. The offsets and/or cutter compensation are automatically reactivated during the first block containing axis motion following the G28 or G30, unless that block is a G29 block. If a G29 follows, the offsets and/or cutter compensation remain deactivated on the way to the intermediate point and are reactivated when the axis moves from the intermediate point back to the point indicated in the G29 block.

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, then this has the effect of returning the axis from machine home.

For example, executing the block:

```
G29 X7.0 Y.2 Z-14.0 ;
```

in absolute mode would move the axes to (X7.0, Y.2, Z- 15.0) after passing through the intermediate point stored in control memory. In incremental mode this block would move the axes to a position that is X7, Y2, and Z14 units away from the home point.

The intermediate point is stored in control memory after a G28 return to machine home or a G30 move to alternate home is executed. A G29 block is usually executed after a G28 or G30 block, typically to return the cutting tool to the part after a tool change.

Although this command moves the axes at rapid feedrate as if in G00 mode, it is not modal. If G01, G02, or G03 modes are active, they are temporarily canceled for the return from home moves.

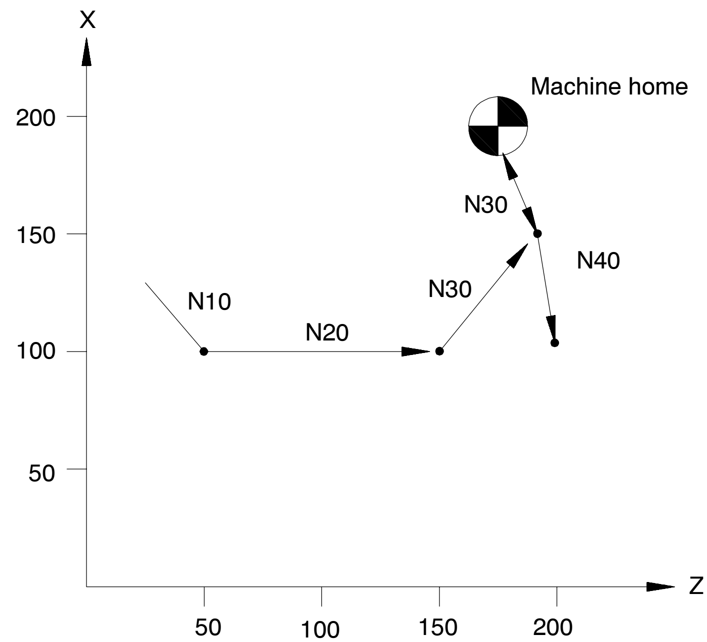
Only the axes specified in the G29 block are moved. For example:

`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 13.13
Automatic Return From Machine Home

```
N00010  X100.Z50.;
N00020  Z150.;
N00030  G28X150.Z180.;
N00040  G29X200.Z100.;
```

Figure 13.18
Automatic Return From Machine Home, Results of Example 13.13



Important: When a G29 is executed, tool offsets and/or cutter compensation are deactivated on the way to the intermediate point, and they are reactivated when the axis moves from the intermediate point back to the point indicated in the G29 block.

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__ Y__ Z__ ;
```

The value entered with the axis name in the G27 block must be the machine home coordinate for that axis. If it is not, no axis motion 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 tool offset modes have been cancelled.

If an attempt is made to execute a G27 before the axes have been homed, the control goes to cycle stop and displays this error message:

"MACHINE HOME REQUIRED OR G28"

Return to Alternate Home (G30)

The G30 command is similar to the G28 command. The main difference is the axis or axes move to an alternate home position instead of machine home. 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__ Y__ Z__ ;

     or          (second alternate home position)

G30 P2 X__ Y__ Z__ ;
```

Important: The control generates the error "P VALUE OUT OF RANGE" if the P value is illegal. For example, a P1 or P5 would be illegal and generate the error.

The axis words in the above block establish the intermediate point in the same manner as the G28 code. That is, the axes will move to the intermediate point defined in the G30 block prior to moving to the alternate home position. When intermediate values are programmed in a G28 block they replace G30 intermediate point values and vice-versa. This intermediate point is used by the G29 automatic return code.

Only those axes included in the G30 block are sent to the alternate home position. For example:

`G30 X5.6`	The control moves the X axis to second home after moving to 5.6 on the X axis. The Z and Y axes are not moved.
`G30 P3 X1.0 Z4.0`	The control moves the X and Z axes to third home after moving to 1.0 on the X axis and 4.0 on the Z axis. The Y axis is not moved.

A typical application for the G30 command would be if the automatic tool changer were located at a position other than machine home.

If an axis included in the G30 block has not been homed, block execution stops and this error message appears:

"MACHINE HOME REQUIRED OR G28"

Important: When the control executes a G28 or G30 block, it temporarily removes any tool offsets and cutter compensation during the axis move to the intermediate point. The offsets and/or cutter compensation are automatically reactivated during the first block containing axis motion following the G28 or G30, unless that block is a G29 block. If a G29 follows, the offsets and/or cutter compensation remain deactivated on the way to the intermediate point and are re-activated when the axis moves from the intermediate point back to the point indicated in the G29 block.

Dwell (G04)

The G04 command delays the execution of the next data block. Dwell length is specified in either of two types.

- Seconds
- Number of spindle revolutions

The type used is normally dependant on the feedrate mode (G93, G94 or G95) active at the time. The type can also be permanently fixed to "seconds," regardless of G93, G94 or G95 mode, by setting the proper AMP parameter.

Dwell - Seconds

In the G94 mode (feed per minute) G04 suspends execution of the commands in the next block for a programmed length of time in seconds.

```
G94G04  P__;    X__;    U__;
```

Specify the required dwell time by either a P-, X-, or U-word in units of seconds. It does not matter which of these three words you use, as long as only one appears in the same block. The allowable dwell time is 0.001 - 99999.999 seconds.

When you program a dwell in seconds you system installer has the option of writing logic to allow a portion of the dwell to be skipped. If this feature is used, when the appropriate signal is sent to logic (from a switch or other device) the control automatically skips any portion of the dwell that has not been executed and proceeds to the next block in the program. The axes positions when the skip signal is sent to logic is recorded and stored as system parameters #5071 - #5076. See specifics on the G31 skip cycles for details.

Dwell - Number of Spindle Revolutions

In the G95 mode (feed per revolution), G04 suspends execution of commands in the next block for the time it takes the controlling spindle to turn a designated number of revolutions.

```
G95G04  P__;    X__;    U__;
```

Specify the required dwell length by either a P-, X-, or U-word in units of spindle revolutions. It does not matter which of these three words you use, as long as only one appears in the same block. The allowable range is 0.001 - 99999.999 revolutions.

Mirror Image (G50.1, G51.1)

There are two types of mirroring. They are:

Mirror image:	Activate through:
programmable	programming a G50.1 and G51.1
manual	logic

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. 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__Y__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 designated point parallel with the axis. The G51.1 code is modal and remains in effect until cancelled by a G50.1 command.

Use the axis word programmed with the G51.1 command to define the mirroring location. The defined location intercepts the programmed axis at the programmed position. If only one axis is programmed, the mirroring plane is perpendicular to that axis. If more than one axis is programmed, the mirror plane passes through these points.

Important: The control mirrors only those axes that are programmed out in the G51.1 block. Axes not programmed in the G51.1 block execute normally.

A G50.1 block cancels the mirror image function.

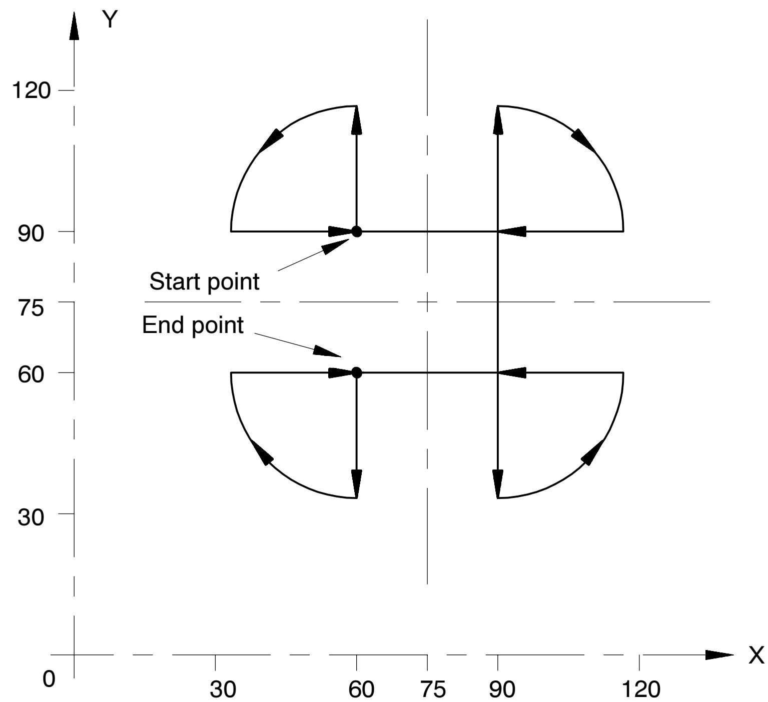
```
G50.1X__Y__Z__ ;
```

The control only cancels the mirror feature for those axes that are programmed in the G50.1 block. Axes not programmed in the G50.1 block remain mirrored. There is no significance to the values programmed with the axis words in a G50.1 block. Axis values might not be required, depending on how the way AMP was configured by your system installer. In either case, the control ignores these values.

Example 13.14
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 X
M98P8500;	call subprogram 8500
G51.1X75.;	mirror active on Z (and X)
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 13.19
Programmable Mirror Image, Results of Example 13.14



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 cutter compensation. G41 becomes tool right and G42 becomes tool 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.

The manual mirror image feature differs from the programmable mirror image feature. When you use 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. It 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 can install a switch for each of the 4 available axes. What axes are mirrored with what switches depends on the logic program in your system. You can mirror about more then one axis using more then 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. The programmable mirror image is done first, followed by the manual mirror image. The same axis can be mirrored by programmable and manual mirror image at the same time.

Axis Clamp

Use this feature to disable the axis position display and allow an axis to be clamped into position. Typically an axis clamp is performed by the execution of an M-code in a part program or by a switch of some type controlled by the operator. Your system installer determines how the axis clamp feature is enabled in logic. Refer to your system installer's documentation for details.

When an axis is clamped, the control freezes the axis position displays at their position. Any drift or movement generated by some external force does not generate any corrective response from the axis servo. This prevents the servo from trying to move an axis back into position when it has been mechanically clamped so it cannot move.

Any movement of the axis when it is clamped is added to the current value of the following error. You can view this on the screen displaying following error. Refer to the Integration manual. If the following axis error exceeds its 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.

Feed to Hard Stop (G24)

The feed to hard stop feature is used to position the axis of a transfer line station or the transfer bar of the station against a mechanical stop and hold it against the stop. This mechanical stop physically halts axis travel. The system installer determines the position of this hard stop based on mechanical consideration of the machine and the process currently being performed by the axis or transfer bar.

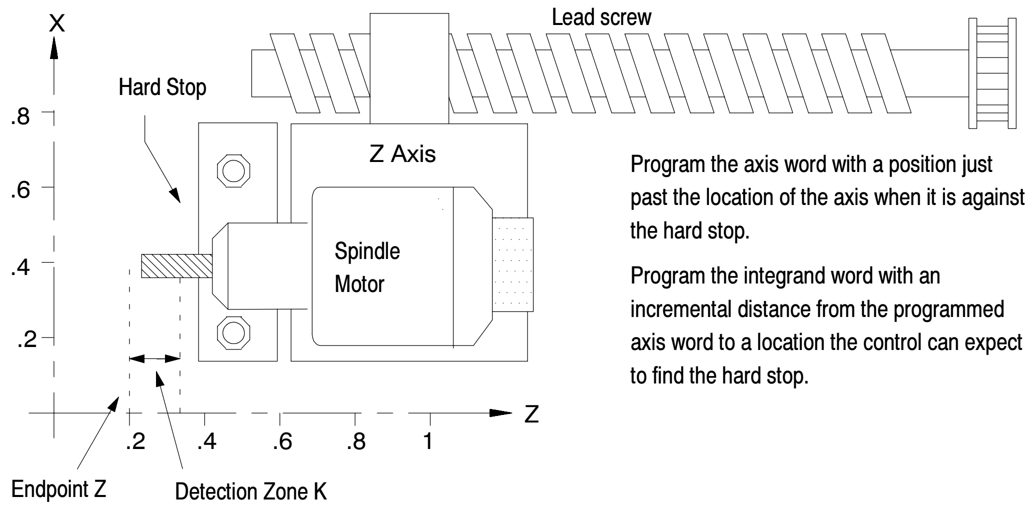
Program a feed to hard stop using a G24 code. Format for the G24 code is as follows:

$$G24 \begin{pmatrix} X\_ \\ Y\_ \\ Z\_ \end{pmatrix} \begin{pmatrix} I\_ \\ J\_ \\ K\_ \end{pmatrix} F\_\_;$$

Where:	Programs:
X, Y, or Z	Hard stop axis. Use the axis word associated with the axis to be hard stopped. Program an endpoint for the axis that is past, but very close to, the actual hard stop location. The axis word can be programmed as either an absolute or incremental value (G90 or G91 mode).
I, J, or K	Detection zone. Use the axis integrand associated with the axis to be hard stopped. Program an incremental distance from the programmed endpoint for the control to start looking for a hard stop. The axis detection zone is an unsigned value.
F	Optional in the G24 block. F defines the active feedrate for the G24 feed to hard stop. If not programmed the currently active feedrate is used.

Example 13.1
Feed to Hard Stop

G90 G01 F20;
G24 Z.2 K.15;



Program the axis word with a position just past the location of the axis when it is against the hard stop.

Program the integrand word with an incremental distance from the programmed axis word to a location the control can expect to find the hard stop.

Moving to the Hard Stop

The G24 code must be in a block that programs a position for one and only one axis. The G24 code is non-modal (G–code group 0).

The active cutting mode when the G24 code is executed must be G01 (linear interpolation). Other cutting modes and rapid traverse (modal group 01), are invalid during a G24 block.

Once the G24 code is executed the axis moves towards the programmed endpoint at the currently active feedrate. When the axis enters the detection zone set up with the integrand word, the control expects to encounter a hard stop. If a hard stop is detected before entering this detection zone, the control generates an error. An error is also generated if the control reaches the programmed endpoint without encountering a hard stop. Both of these error conditions force a cycle stop.

Detecting the Hard Stop

A hard stop is detected when the control's torque output reaches a Hard Stop Detection Torque that the system installer configured in AMP. This torque limit must be reached after entering the detection zone and before reaching the G24 blocks' programmed endpoint or an error is generated.

Once an axis is positioned against a hard stop, that axis remains in the hard stop state until it is programmed away from the hard stop. While against the hard stop, the control applies a set holding torque to the servo keeping the axis firmly pinned against the hard stop. This hard stop holding torque amount is determined by the system installer in AMP.

Only one axis can be positioned against a hard stop at any one time. Attempting to position more than one axis against a hard stop results in a programming error. Once an axis is positioned against a hard stop, it must be programmed away from the hard stop before another axis can use the hard stop feature. You can program an axis that is currently in the hard stop state to a different hard stop location with a G24 block provided the hard stop is in a direction opposite the current hard stop holding torque.

Special Considerations

Feature:	Consideration:
Control Reset	If a control reset operation is performed while the control is against a hard stop the holding torque is released and the axis is taken out of the hard stop state.
Block Reset	If a block reset is performed during a G24 block before the hard stop has been reached, the torque limits applied to that axis are removed and the G24 block is aborted.
Program Checking	Feed to hard stop blocks are executed as normal G01 moves while in QuickCheck. While in dry run the control will not execute G24 feed to hard stop blocks at rapid. Feed to hard stop blocks are executed at the programmed feedrate during dry run and all the axis torque restrictions are applied (dry run is ignored during the G24 block).
Deskew and Dual Axes	Only one servo can be fed to a hard stop. This restriction makes programming a G24 code for either a Deskew axis or Dual Axes invalid.
Probing	You can not use probing with the feed to hard stop feature because a probe G-code and a hard-stop G-code cannot be programmed in the same block. The probe must be unarmed when the G24 block is executed, and when the axis is removed from the hard stop. You can however perform probing while the axis is parked against the hard stop.
Block Retrace	You can not retrace any block that moves an axis off of a hard stop. This is because the start point of that move is determined by the hard stop which can only be safely reached by executing a G24 block. You can retrace a G24 block as long as the axis is still against the hard stop when the retrace operation begins. You cannot retrace through programmable acc/dec blocks (G47.x and G48.x). However, you can retrace through blocks where programmable acc/dec was already active.
Interrupt Programs	You can execute an interrupt program during a G24 block provided you are not performing a type 1 interrupt (type 1 interrupts are incompatible with the hard stop feature). You can not move an axis that is currently holding against a hard stop using an interrupt macro. You can not execute a G24 block within an interrupt macro.
Exact Stop Mode (G61)	The G61 (exact stop mode) does not function on G24 blocks. G61 mode is ignored when a G24 is executed.
Polar Programming (G16)	You can not program a G24 block if the axis you are programming against the hard stop is in the current plane and the control is in polar programming mode.

END OF CHAPTER

Using QuickPath Plus™

Chapter Overview

The QuickPath Plus (QPP) feature is offered as a convenient programming method to simplify programming. This method of programming can prove useful in simplifying the programming of a part directly from a part drawing. In this chapter we describe:

We discuss some QuickPath Plus features in this chapter. Major topics include:

The most significant advantage to the QuickPath Plus feature is the programmer no longer has the need to calculate the endpoint of every block or every point of intersection. QuickPath Plus determines these points from angles and lengths.

QuickPath Plus uses these addresses:

,A	Angle – This word is used to define the angle of a tool path. 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 is used to define the length of a linear tool path, the direction of which is determined by the angle (,A). It is always interpreted as an incremental value.

Programming QuickPath Plus

When programming QuickPath Plus, remember:

- 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 that the ZX plane is active (angles are measured relative to the Z axis).
- QPP always uses ",A" as the angle word. When you create new programs, always program the QPP angle with ,A. Your system installer has the ability to define in AMP an additional letter that can also be used for the QPP angle. Refer to your system installer's documentation. This additional QPP angle word is provided only for program compatibility with older systems.
- The angle word (,A) is always interpreted as an absolute angle, regardless of the current mode (G90 or G91).

- The L-word is always interpreted as an incremental distance from the current position regardless of the current mode (G90 or G91). Radius or diameter mode (G08 - G09) has no effect on the ,A- or L-word.
- If you need to 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 of the two blocks, regardless of whether the error is in the first or the second block. If programming in**`<SINGLE BLOCK>`** mode, the control stops after the execution of the first block as normal.
- If you need to program more then one block to perform the QuickPath Plus operation being used, a maximum of 4 non motion blocks can be programmed between these blocks. A non motion block is any block that does not generate axis motion on one of the two axis 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.

The G-code table in appendix C lists the G-codes and their group numbers.

- If you need to program more then 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 these blocks.
- If an angle is programmed in a circular QuickPath Plus block, an error is generated.
- If an L-word is programmed in a G13, or G13.1 block an error is generated.

Linear QuickPath Plus

One-end Coordinate

Many times part drawings give a programmer only one axis dimension for a tool path and require that the other axis dimension be calculated by the angle. This QuickPath Plus feature eliminates the need for this calculation. This must be a linear block. Refer to page 14-7 for circular block.

The format for this block is:

$$,A\_\_ \quad \begin{Bmatrix} X\_\_ \\ Y\_\_ \end{Bmatrix} \; ;$$

Where :	Is :
,A	Angle - This word is used to define the angle of a tool 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,Y	Endpoint - 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.

Only one axes word from the current plane can be programmed 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 tool path.

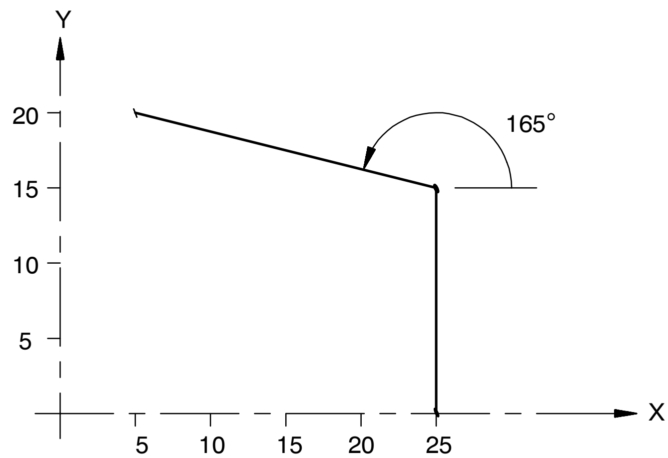
If both axis words from the current plane are entered in the block, the angle is ignored and the control moves to the coordinate position programmed with the axis words. All examples in this section assume that the ZX plane is active.

Important: If the programmed tool path is going to be parallel to an axis in the current plane, the axis word for the end point in the block should be for the axis in the current plane that is not parallel to the tool path. This means if the value of the angle (,A-word) is 0 degree or 180 degree, the second axis in the plane must be programmed in the block. If the value of the angle is 90 degree or 270 degree, the first axis in the plane must be programmed in the block.

Example 14.1
Angle Designation:

```
N10 GO0 X25 Y0 F100.;
N20 G01 Y15 ,A90;
N30 X5.,A165;
N40 M30;
```

Figure 14.1
Results of Angle Designation, Example 14.1



Important: Circular QuickPath Plus can also use an angle (,A) in a program block. This is described on page 14-7.

No End Coordinate Known (L)

This feature of QuickPath Plus allows the programmer to define a tool path using only the start point angle and length of a tool path. This must be a linear block.

The format for this block is:

```
,A__ L__;
```

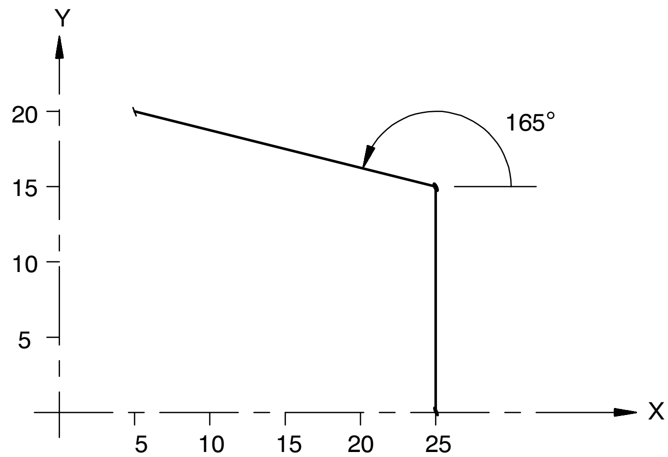
Where :	Is :
,A	Angle - This word is used to define the angle of a tool path. This manual assumes that the ,A-word is used. The angle is a positive value when measured counterclockwise from the first axis defining the currently active plane and a negative value when measured clockwise. The angle is in units of degrees.
L	Length - This word determines the length of the tool path. It is measured from the start point to the end point of the move along a linear path. No coordinate points are necessary.

Important: If any axis word from the current plane is designated in the block, the L-word is ignored and the control calculates the end point from the angle and the axis word. If an angle (,A) or a length (L) is programmed in a block that also contains both axis words in the current plane, then QuickPath Plus is not performed and the control ignores the ,A- and the L-words in the block.

Example 14.2
Angle with Length Designation:

```
N10 G00 Y0. X25. F100;
N20 G01 ,A90 L15;
N30 ,A165 L20.7;
N40 M30;
```

Figure 14.2
Results of Angle With Length Designation, Example 14.2



No Intersection Known

This feature of QuickPath Plus allows the programmer to define two intersecting, consecutive, linear tool paths without knowing the point where the actual intersection takes place. Both of these blocks must be linear blocks and programmed in absolute mode. The angle of both of these lines must be known.

This is done with a sequence of two linear blocks (in the current plane) in which QPP is used to calculated the end point of the first block. The start point of the first block is the current tool position.

Important: The second block of these two blocks must be programmed in absolute mode. Any attempt to program the second block in incremental generates an error.

The format for these blocks is:

```
,A__ ;
,A__ X__ Y__ ;
```

Where :	Is :
`,A`	Angle - This word is always displayed as by the control even if the angle is named differently in AMP. If you have a 9/240 program that uses a different address than ,A and you want to run the program on a 9/260 or 9/290 control the angles will work but the control names them ,A.
`XY`	End Point of second block - These are the actual coordinate location of the end point of the second block. They must be programmed as absolute values and must be axes in the current plane.

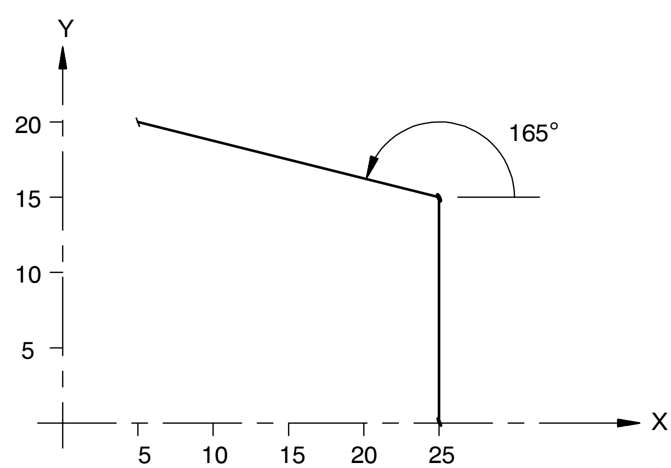
Important: There may be up to four program blocks between the two blocks in the above format. The only requirement being that these blocks may not generate axis motion in the current plane.

Both of these blocks must be programmed in the same plane. If the current plane is changed between these two blocks execution, the control generates an error.

Example 14.3
QPP When An Intersection is Unknown

```
N10 G00 X25 Y0 F100;
N20 G01 ,A90;
N30 ,A165 X5 Y20.;
N40 M30;
```

Figure 14.3
Results of Unknown Intersection, From Example 14.3



If the control cannot determine an intersection point for the two linear paths (e.g., if the paths are parallel), an error occurs.

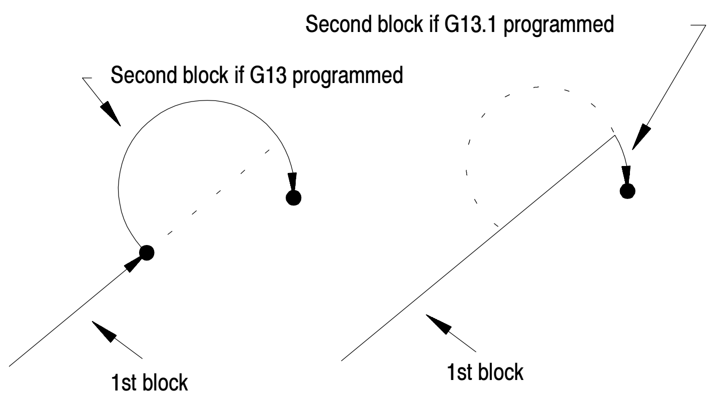
Circular QuickPath Plus (G13, G13.1)

The programmer uses the Circular QuickPath when a drawing does not call out the actual intersection of two consecutive tool paths and at least one of the tool paths is circular. This prevents the programmer from having to do any complex calculations to determine end points and start points when an arc is involved.

For most cases of circular QuickPath Plus there may be two possible intersection points for the two defined blocks. Define which intersection is desired using either G13 or G13.1 in the first of the two blocks.

Programming:	Defines:
G13	the first intersection that occurs when the tool path of the first block intersects with the second block
G13.1	the second intersection that occurs when the tool path of the first block intersects with the second block.

Figure 14.4
G13 vs G13.1 Intersections



When programming circular QuickPath Plus, remember:

- When there is only one intersection involved with the tool paths, you can program the G13 and G13.1 codes interchangeably. One of these G-codes must be programmed however.
- The G13 or G13.1 code must be programmed in the first of the two blocks defining the two tool paths.
- If the arc is programmed with an R-word, the two tool paths must be tangent. The sign (+ or -) of the R-word determines the arc center location as described on page 13-5.
- The angle word (,A) cannot be programmed in a circular block.
- Both absolute coordinate values in the current plane must be programmed for the second block. **Both must be programmed** regardless of whether the final coordinates change or not.

Linear to Circular blocks

When the coordinates of the intersection of a linear path into a circular path are unknown, use the following format. G13 or G13.1 must be programmed. These blocks must be programmed in absolute.

Format:

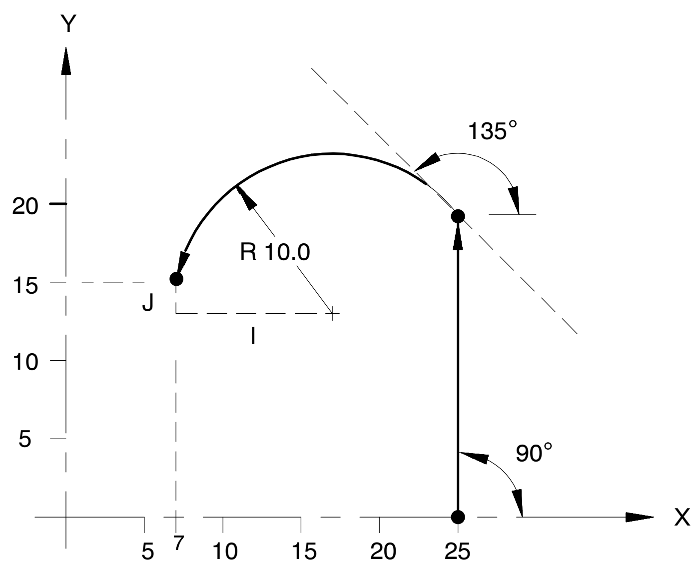
```
G13 G01 ,A__ ;               or    G13G01 ,A__ ;
G02 X__ Y__ I__ J__ ;             G02 X__ Y__ R__ ;
```

Important: If the second block is an arc and it is programmed by using I, and K integrand, 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. The integrands specify the distance from the end point of the arc to the center point.

Example 14.4
Line Into Arc Without Programming Intersection

```
G00X25.Y0.F500
G01G13.1,A90;
G03X7.Y15.I9.21J-2.;
M30;
```

Figure 14.5
Results Of Line into Arc Without Intersection, Example 14.4



Important: You cannot program R to specify the arc radius for linear-to-circular block combinations unless the two tool paths are tangent.

Circular to Linear blocks

When the coordinates of the intersection of a circular path into a linear path are unknown, use the following format. G13 or G13.1 must be programmed in the first of the two blocks. These blocks must be programmed in absolute.

Format:

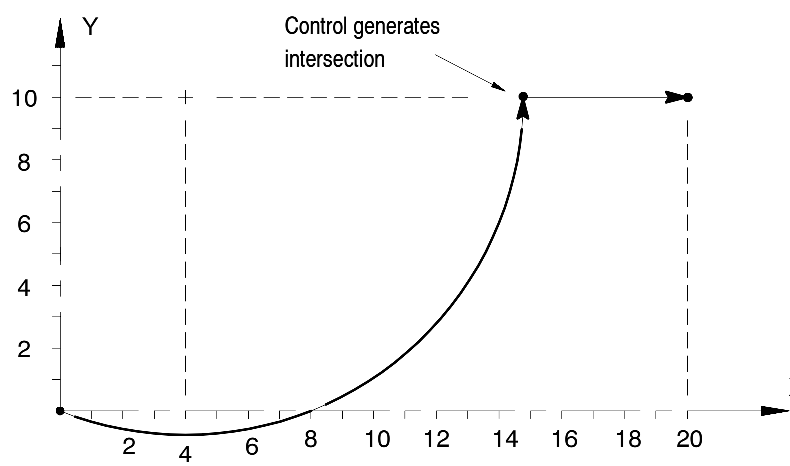
```
G13 G02 I__ J__ ;    or    G13 G02 R__ ;
G01 ,A__ X__ Y__ ;          G01 A__ X__ Y__ ;
```

Important: J, and K values are the normal integrand values when using this format (measured from start point of arc to arc center). These are discussed in chapter 16.

Example 14.5
Arc Into Line Without Programming Intersection Point

```
G00X0.Y0;
G13G03I4J10F100;
G01,A0X20Y10;
M30;
```

Figure 14.6
Results of Arc Into Line Without Intersection, Example 14.5



Important: R cannot be programmed to specify the arc radius for linear to circular block combinations unless the two tool paths are tangent.

Circular to Circular blocks

When the coordinates of the point of intersection of a circular path into a circular path are unknown, use the following format. 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:

```
G13 G02 I__ J__ ;
G02 X__ Y__ I__ J__ ;
```

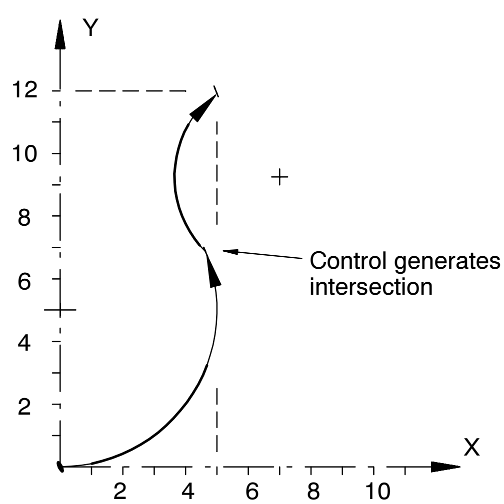
Important: The I, J, K integrand vectors are not necessarily the same values as discussed in chapter 13. The integrands of the first circular block specify the distance from the start-point to the center of the circle. The integrands of the second circular block specify the distance from the end-point to the center of the circle (this is the reverse of normal). At least one of these integrand words must be programmed in each of the two circular blocks.

Neither circular block can contain an angle word (,A) when you use this format.

Example 14.6
Arc Into Arc Without Programming Intersection

```
G0X0Y.;
G13G03J5F100.;
G02Y12X5I2J-2.75;
M30;
```

Figure 14.7
Results Arc Into Arc Without Intersection, Example 14.6



END OF CHAPTER

Chamfering and Corner Radius

Chapter Overview

During cornering, the 9/PC control has the option of performing either a chamfer (a linear transition between the blocks) or a corner radius (an arc transition between blocks).

,C	Chamfer size	This word is used to define a chamfer length that connects two intersecting tool paths. This word determines the distance that the chamfer begins and ends from the tool paths intersection.
,R	Corner radius	This word is used to define the radius of an arc that is tangent to two intersecting tool paths.

This chapter describes chamfering and corner radius in detail. Major topics include:

Both the chamfer and the corner radius features are generated between two motion blocks that 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-word is programmed in the same block, only the right-most word is used; others are ignored. 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.

ATTENTION: If you make a programming error of some type is made in the block defining the second tool path in the chamfer or radius blocks, the control is not able to cut the correct chamfer or radius. Instead, the first block is executed to its programmed endpoint. This can cause damage to the part or cutting tool.

There is a limit of four nonmotion blocks allowed between the first and second motion blocks defining the corner transition. A nonmotion block is any block that does not generate axis motion in the currently active plane. The control generates an error if more than four nonmotion blocks are programmed between the cornering plane.

Use 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.

Chamfering

Program a chamfer size following the address ,C to cut a chamfer between consecutive tool paths. The chamfer word must follow a comma (,) and is programmed in the first of two paths connected by the chamfer. The value following the ,C address is the amount of tool path cut of each programmed tool path by the chamfer. The angle that the chamfer makes with the tool paths is dependant on the size of the chamfer.

Measure the chamfer size from the intersection of the two blocks.

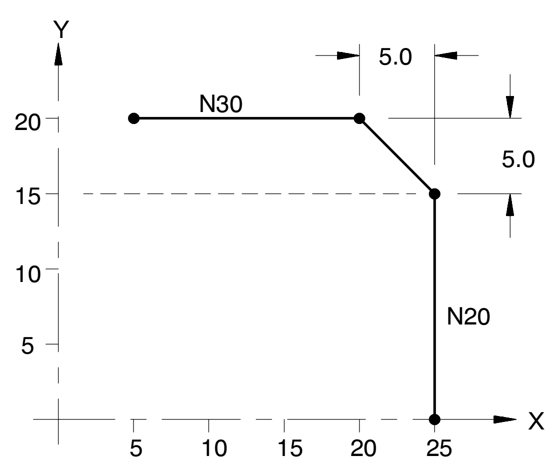
If the block:	Then:
linear	distance programmed with the ,C-word is measured from the intersection of the two tool paths along the linear path.
circular	then the chamfer distance programmed is applied as a chord length on the arc measured from the intersection between the two blocks. This applies regardless of the combination of arcs and lines to be chamfered.

The ,C-word can be programmed any where in a block as long as no space is programmed between the comma and the chamfer distance.

Example 15.1
Basic Chamfering Using ,C

```
N10 G00 Y0 X25 F100;
N20 G01 Y20., C5.0;
N30 X5.0;
N40 M30;
```

Figure 15.1
Results of Chamfering Using ,C from Example 15.1



Example 15.2
Linear to Circular Motions with Chamfer

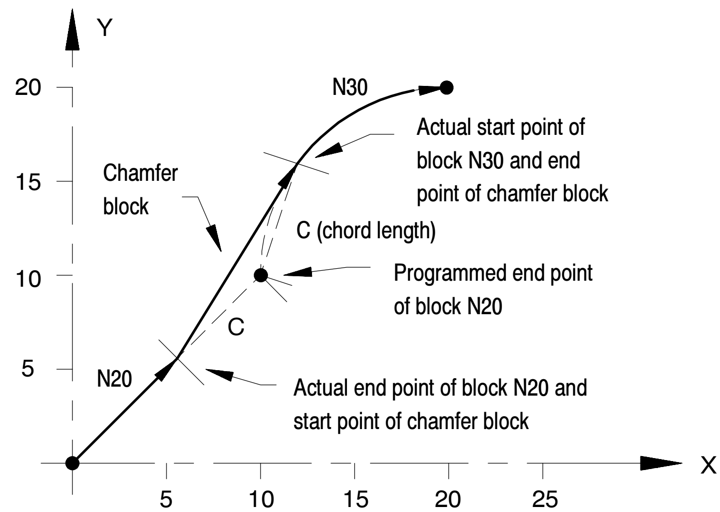
```
N10 G00 X0 Y0 F100;

N20 G01 X10. Y10.,
C3;

N30 G02 X20. Y20.
R10;

N40 M30;
```

Figure 15.2
Results of Linear to Circular Motions with Chamfer, Example 15.2



Corner Radius

Use the ,R command to program a radius between two intersecting tool paths. The R command must be programmed after a comma (,). Program the ,R followed by the radius size in the block where the first path is programmed. The control looks ahead to the block commanding the second path and automatically inserts the circular rounding bock to meet that path. This inserted circular block is always tangent to both programmed tool paths. If the control cannot generate an arc that is tangent to both paths with the programmed ,R, then the control generates an error.

Block:	Description:
The first corner radius	always terminates at the point on the block where the rounding block is tangent to the first block
The rounding	terminates at the point where the generated rounding block is tangent to the second rounding block.
The second rounding	starts from the end point of the generated circular block and continues on to the programmed end point of the second block.

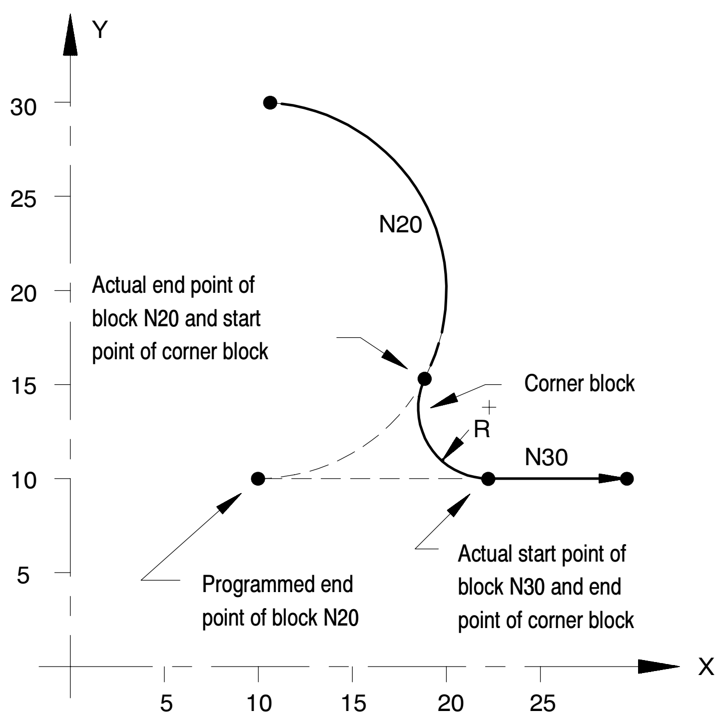
The R-word can be programmed any where in a block as long as no space is programmed between the ,R and the radius length.

Important: If the two motion blocks are tangent to each other, then any corner rounding commands are ignored.

Example 15.3
Programming a Radius for a Circular Path into a Linear path.

```
N10 G00 X10. Y30;
N20 X10. Y30 F100;
N30 G02 X10. Y10 R10,
R3;
N40 G01 X30. Y10;
N50 M30;
```

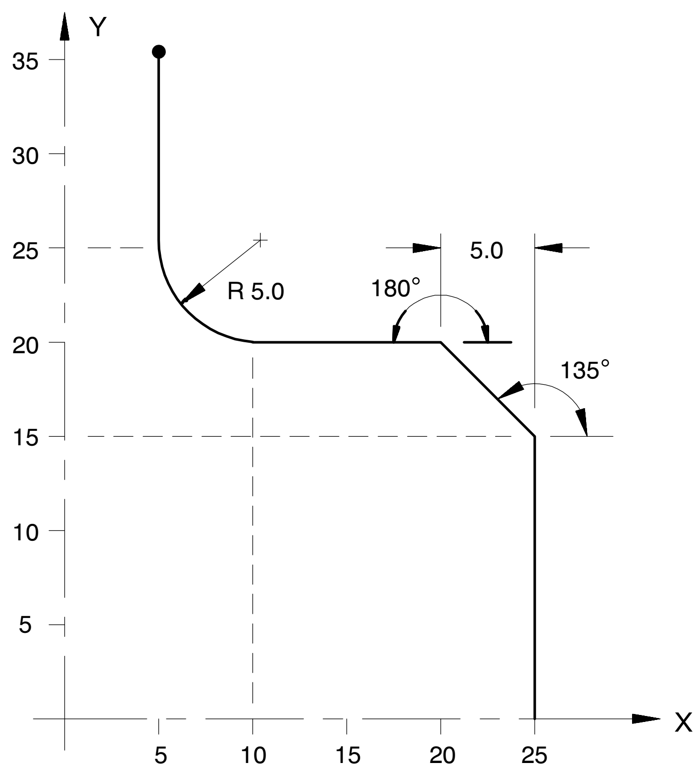
Figure 15.3
Results of Radius for a Circular Path into a Linear path, Example 15.3



Example 15.4
Radius and Chamfer with QuickPath Plus

```
N10 G00 X25. Y0. F100;
N20 G01 ,A90, C5.0;
N30 X5. Y20.
,A180.,R5.0;
N40 Y35;
N50 M30;
```

Figure 15.4
Results of Radius and Chamfer, Example 15.4



Considerations with Chamfering and Corner Radius

When using chamfering and corner radius, remember:

- 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 nonmotion 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 nonmotion blocks are executed after the control has executed the chamfer or radius.
- Any negative signs programmed with the ,C- or ,R-words are ignored. Use the absolute value of the word to cut the chamfer or radius. For example ,C-10 is used as ,C10.

- An error is generated if the length of a chamfer is larger than the programmed length of the first or second move, or for corner rounding if the programmed corner radius is so large that the tangent point on both of the two programmed blocks does not exist.
- An error is generated if you attempt to change planes between blocks that are chamfer or corner radius blocks.
- You must program ,C and ,R 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 cutting 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 documentation prepared by your system installer for details.

END OF CHAPTER

Spindles

Chapter Overview

This chapter describes how to program spindles:

Controlling Spindle (G12.1, G12.2)

The G12 code is used to program the active controlling spindle for features and modes requiring spindle operation. The G12 code is modal and only one spindle may be the controlling spindle. All other spindles are auxiliary spindles.

- G12.1 — Spindle 1 Controlling
- G12.2 — Spindle 2 Controlling

Table 16.A lists the allowed spindle capabilities.

Table 16.A
Spindle Capabilities

Control Type	Number of Spindles	Spindle Type
9/PC	2	Primary Auxiliary 2

Spindles 1 and 2 must be configured in AMP, and the associated spindle parameters must be set properly to provide for the required spindle functions.

For systems with no spindle configured, simulated spindle feedback is provided for the controlling spindle. This allows all control features that require spindle feedback, i.e., IPR feedrate, threading, CSS, to simulate the feedback from a spindle even through the AMPed system configuration contained no spindle. The default is 4000 counts-per-rev device.

Important: On the 9/PC control, if the auxiliary spindles are programmed but have not been configured as active through AMP, this error is given as decode errors on any blocks that contain the G12.2 code:

"SPINDLE 2 NOT CONFIGURED"

Spindle Speed (S-word)

Use the S-word to program the spindle speed for all configured spindles. The common S-word can be applied per spindle by associating the S-word in the same block with the spindle directional M-codes. Refer to page 16-4 for information about spindle directional M-codes. If no directional M-code is programmed in the block with the S-word, then the S-word is applied to the active controlling spindle.

ATTENTION: The displayed S-word always shows the controlling spindle's programmed spindle speed. When the noncontrolling spindles are configured, their S-word display must be handled by some other means. Refer to the system installer's documentation for display capabilities of the active spindle speed for auxiliary spindles.

The S-word units represent revolutions per minute (RPM) in most cases. Only during CSS programming are the S-word units different. While CSS mode is active, the S-word units represent surface feet per minute. Only the controlling spindle can change its S-word mode from RPM to CSS.

Spindle Orientation (M19, M19.2)

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. 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 logic. 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: A spindle orient is also sometimes automatically requested by the control when performing some of the drilling cycles described in chapter 25. This drilling cycle orient orients to either the AMP-defined position if using a closed-loop orient type or to the position defined as the open-loop orient position.

Important: In systems allowing multiple spindles, only one M19 code can be in a block. If two 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 logic dependant.

Both open- and closed-loop spindle orients can be requested either by programming the appropriate spindle orient code (M19, M19.2) in a program block, or by requesting one through logic. If closed-loop orient is requested through logic, 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), 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 replaces the current modal spindle speed used later when the M19 mode is canceled. 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 Direction (M03, M04, M05)

Use the spindle directional M-codes to program each configured spindle program controlled spindle rotation.

Table 16.B lists the spindle direction codes.

Table 16.B
Spindle Directional Codes

Spindle Type	Directional Code	This means:
Primary	M03 M04 M05	Spindle 1 clockwise Spindle 1 counterclockwise Spindle stop
Spindle 2	M03.2 M04.2 M05.2	Spindle 2 clockwise Spindle 2 counterclockwise Spindle 2 stop

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 16.1
Spindle Synchronization with 2 Spindles Configured in AMP

`N0001 M05`	Spindle 1 stop
`N0002 M05.2`	Spindle 2 stop
`N0003 M03 M04.2 S150`	Spindle 1 clockwise 150 rpm Spindle 2 counterclockwise 150 rpm
`N0004 M03.2 S10`	Spindle 2 clockwise 10 rpm

Important: On the 9/PC control, if the auxiliary spindle directional M-codes are programmed but the auxiliary spindles have not been configured as active through AMP, this error is given as decode error on any blocks that have directional M-codes of the associated spindle programmed:

"SPINDLE 2 NOT CONFIGURED"

Synchronized Spindles

Use this feature to synchronize the position and/or velocity between two spindles with feedback using your 9/PC control.

Two types of synchronization are available:

- Velocity — synchronizes only the speed between two spindles
- Velocity and Position — synchronizes the speed and angular position between two spindles

Prior to activation, you are responsible for selecting the proper gear ranges and ratios. The gear ratio between the feedback device and the spindle must be 1:1. Any other type, including nonunit ratios, will not allow repeatability of the orientation of your spindle and may cause positioning offset inaccuracies.

Spindle Configuration

Your system installer selects two spindles to make up the synchronization pair, which consists of the controlling and follower spindles. During synchronization, the controlling spindle initiates spindle motion while the follower spindle attempts to synchronize with it. Your system installer determines the configuration of these spindles. Refer to your system installer's documentation for more information about spindle configuration.

Gear ranges are set separately for each spindle. If the controlling spindle speed is outside of the current follower spindle gear range when a seek is attempted, the controlling spindle will ramp to within the follower's limits set in AMP.

Selecting the Controlling Spindle

The synchronized spindle's controlling spindle, which is determined by your system installer, must be programmed as the part program's controlling spindle in your part program prior to synchronization. Use one of the G12 codes (G12.1, and G12.2) to designate the active controlling spindle for spindle synchronization. Refer to page 16-1 for more information about the G12 codes and your system installer's documentation to identify your controlling spindle.

Important: Typically, the programmed speed of the controlling spindle dictates the speed of the follower spindle. For more information about valid gear ranges, refer to page 16-9.

Using the Spindle Synchronization Feature

Use these three G-codes to manipulate the spindle synchronization feature:

- Set spindle positional synchronization (G46)—sets the follower spindle speed/direction and relative position offset to match the controlling spindle.
- Set active spindle speed synchronization (G46.1)—sets the follower spindle speed/direction to match the controlling spindle.
- Deactive spindle synchronization (G45)—shuts off synchronization while maintaining the controlling and follower spindles' current speed and direction.

Activate Spindle Positional Synchronization (G46)

Use the "Activate Spindle Positional Synchronization" to synchronize speed and position. The position is based on a programmed S-word (degrees). If you do not program an S-word in the G46 block, it will automatically go to the relative positional offset, set by your system installer. Refer to your system installer's documentation for more information.

During a G46, the spindles attempt to match speeds. Once the speeds are matched, the spindles attempt to synchronize their relative positional offset. Once synchronization is achieved, the active spindle speed and mode (M03, M04, M05, or M19) programmed for the follower spindle is replaced by the current controlling spindle speed and mode.

Important: Changes in spindle speeds that would normally occur as a result of CSS or other programmed changes to spindle speeds, directions, and spindle speed override will not occur until synchronization is achieved.

The format for the G46 block is as follows:

```
G46S__;
```

Where:	Defines:
S	the angular offset between two spindles (degrees)

Important: No other program letters are allowed in the G46 block except auxiliary letters and system installer M-codes.

The following example assumes that the controlling and follower spindles were defined as spindle 2 and spindle 1, respectively, by your system installer.

Example 16.2
Spindle Synchronization

`M03 S200;`	Spindle 1 clockwise 200 rpm
`M04.2 S400;`	Spindle 2 counterclockwise at 400 rpm
`G12.2;`	Spindle 2 as controlling spindle
`G46 S90;`	Spindle 1 changes direction and accelerates to spindle 2's speed; spindle 1 synchronizes angular position with spindle 2 (offset 90 degrees)

Activate Spindle Speed Synchronization (G46.1)

Use the "Activate Spindle Speed Synchronization" to synchronize speed and direction only. Using G46.1 does not guarantee a consistent positional offset between the two spindles. During a G46.1, the follower spindle attempts to synchronize speeds with the controlling spindle. Once synchronization is achieved, the current spindle speed and mode (M03, M04, M05, or M19) programmed for the follower spindle is replaced by the current controlling spindle speed and mode programmed. The original follower spindle speed and direction is not retained.

Important: Changes that occur as a result of CSS or other programmed changes to spindle speeds, directions, and spindle speed override will not occur until synchronization is achieved.

The format for the G46.1 cycle is as follows:

```
G46.1;
```

Important: No other program letters are allowed in the G46.1 block except auxiliary letters and system installer M-codes.

Deactivate Spindle Synchronization (G45)

Use G45 to deactivate the synchronized spindle feature. When synchronization is deactivated, the follower spindle will remain in the same state (M03, M04, M05, or M19) and at the last programmed speed for controlling spindle until you change the program settings or if your system installer writes logic to recommand the spindle.

The format for the G45 cycle is as follows:

```
G45;
```

Important: No other program letters are allowed in the G45 block except auxiliary letters and system installer M-codes.

Special Considerations for Spindle Synchronization

When using the synchronized spindle feature, remember:

- you cannot retrace through a synchronization block (G45, G46, or G46.1). However, you can retrace through blocks where synchronization was already active.
- gear changes are not allowed during synchronization. If spindle speeds exceed the gear range of either spindle, the spindles will be limited to the more restrictive spindle's values.
- due to the servo switch from open- to closed-loop during synchronization, a one-iteration hesitation in the spindles may be seen when this switch occurs. This small deceleration may be more apparent in systems with a smaller spindle motor or if synchronization is done at higher speeds.
- Program Restart, Mid-Start, and Interrupt Macros will be allowed. If synchronization is disabled during an interrupt macro, it will resynchronize upon return, in the event that all of the condition checks listed in this section allow it to, otherwise a decode error will result. Mid-Start and restart must also pass all conditions described in this section.
- you are responsible for selecting proper gear ranges prior to activating synchronization.

The following features cannot be used while synchronization is active:

- solid-tapping
- Virtual C programming

The following features cannot be used while synchronization is ramping:

- deep-hole peck drilling
- threading

Important: Virtual C and threading are available on synchronized spindles once synchronization is achieved.

- When synchronization is active, any part program commands destined for the follower spindle (i.e., M03, M03.2, G12.1, and G12.2) will cause an error.

Important: Typically, the programmed speed of the controlling spindle dictates the speed of the follower spindle. In the event that the programmed speed exceeds the maximum or drops below the minimum allowable values for the synchronized pair, the spindle speed will be restricted to those allowable values, as shown on page 16-9.

- the example below shows what will happen when:
  - no overlap occurs between the controlling and follower spindles' gear ranges
  - the controlling spindle has a higher gear range than the follower spindle
  - the controlling spindle has a lower gear range than the follower spindle

Example 16.3
Valid Gear Ranges for Synchronized Spindles

Controlling Spindle Gear Range (RPM)	Follower Spindle Gear Range (RPM)	Requested Spindle Speed (RPM)	Valid Programmed Spindle Speeds (RPM)	Spindles will Synchronize at (RPM):
1000 to 3000	100 to 300	1500	None	N/A
1000 to 3000	800 to 1500	1800	1000 to 1500	1500
1000 to 3000	1800 to 3200	1500	1800 to 3000	1800

END OF CHAPTER

Programming Feedrates

Chapter Overview

This chapter describes 9/PC control feedrates, including special AMP assigned feedrates and automatic acceleration/deceleration.

Feedrates

Feedrates are programmed by an F-word followed by a numeric value. Feedrates can be entered in a part program block or through MDI. They become effective in the block in which they are programmed. If the block requires rapid traverse motion (G00), the programmed feedrate will be ignored for that block, but will be stored in control memory as the active feedrate.

Feedrates are modal, meaning that they remain active in control memory unless replaced with a different feedrate programmed with an F-word.

The feedrate programmed with the F-word applies to linear interpolation (G01), and circular interpolation (G02/G03) modes. For simplification this manual refers to the cutting tool moving relative to the part, even though most mills move the part across a stationary cutter.

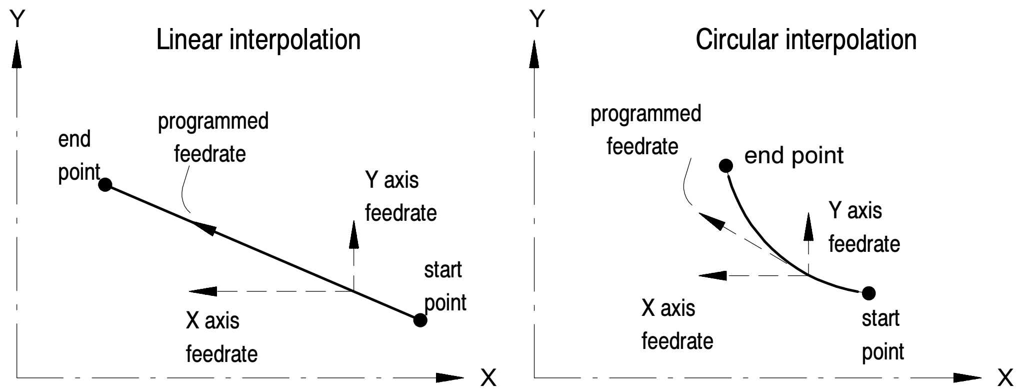
Feedrate modes are either G93 (inverse time feed), G94 (feed per minute), or G95 (feed per spindle revolution). Table 17.A shows the possible feedrate units depending on axis type.

Table 17.A
Feedrate Units

Active G Code	Linear Axis Feed	Rotary Axis Feed
G20/G21 and G93	moves/min.	moves/min.
G02/G03 and G93	radians/min.	moves/min.
G21 and G94	millimeters/min.	degrees/min.
G20 and G94	inches/min.	degrees/min.
G21 and G95	millimeters/rev.	degrees/rev
G20 and G95	inches/rev.	degrees/rev.

Feedrates for linear and circular interpolation are “vector” feedrates. All axes move simultaneously at independent feedrates so that the rate along the effective path is equal to the programmed feedrate. See Figure 17.1.

Figure 17.1
Programming a Tangential Feedrate



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 feedrate of each axis is approximately 70.7 mm per minute.

Feedrates Applied During Cutter Compensation

When the cutting tool is offset from a programmed path, (as in the case of cutter compensation) the programmed feedrate is applied to the center of the tool radius for all linear and outside arc paths. This discussion deals with the speed at which the outside surface of the tool passes across the workpiece surface disregarding tool rotation speed.

For **linear** paths, the result is not significant because the speed of the outside surface of the tool relative to the part surface remains the same as the programmed feedrate.

For **outside** arc paths, the resulting speed of the outside surface of the tool relative to the part surface is less than the programmed feedrate. This generally causes no problem and so the control does not take corrective action.

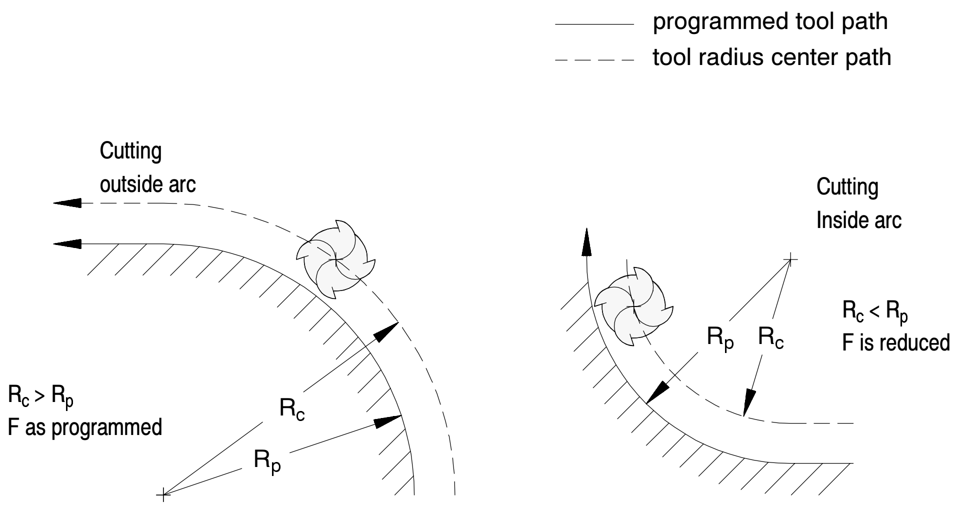
For **inside** arc paths, the resulting speed of the outside surface of the tool relative to the part surface would be greater than the programmed feedrate. Since this could cause excessive tool loading and poor cutting performance, the control automatically decreases feedrate.

For **outside** arc paths, the speed of the outside surface of the tool relative to the part surface can be determined using the following formula:

```
                                 Rp
Tool surface speed  =  F  x  ----
                                 Rc
```

Where :	Is :
F	The programmed feedrate
Rc	The radius of the arc measured to the center of the tool radius
Rp	The programmed radius of the arc

Figure 17.2
Inside and Outside Arc Feedrates with Cutter Compensation



For inside arc paths, the control will automatically maintain the programmed feedrate at the outside surface of the tool. The actual tool radius center feedrate will be reduced as needed through the arc path and then returned to the programmed feedrate after the arc is completed.

During inside arc paths, the control decreases the tool radius center feedrate by the ratio of Rc/Rp. If the Rc value is very small compared to Rp, as in the case of a small arc being cut with a large diameter tool, the value of Rc/Rp will be nearly zero, and the tool radius center feedrate will become excessively small.

To avoid this problem, the system installer must set a minimum feed reduction percentage (MFR) in AMP. This will set a minimum feedrate to be used whenever the value of Rc/Rp is very small. If Rc/Rp control will reduce the tool radius center feedrate no more than the MFR percentage.

Inverse Time Feed Mode (G93)

In G93 (inverse time feed) mode, the F-word represents the amount of programmed axis or axes motion that will be completed in a minute (moves per minute). For example, if a G93 block were programmed with a move to X20 (from X0) and a feedrate of 0.2, the X axis would move 0.2 of the move (to X4) in one minute.

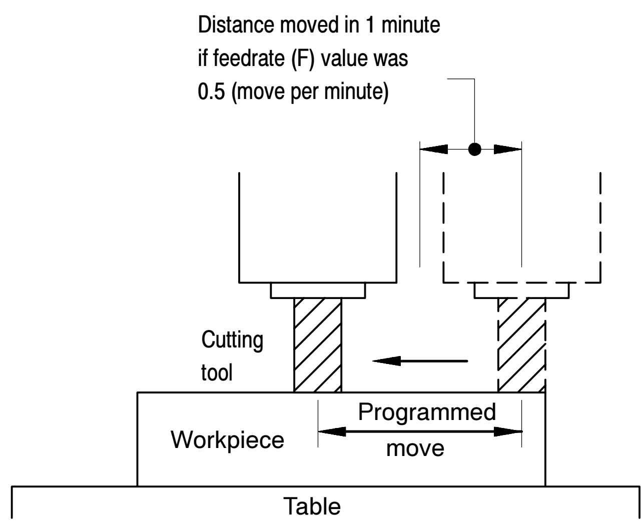
Another way of describing this would be to refer to the F-word as the inverse of the amount of time to complete a programmed move. If it were desired to complete the move described above in 2 minutes, then the F-word value would be 1/2 (program 0.5).

If the programmed move is a circular move (G02 or G03 modes), then the F-word represents the number of radians of the arc or circle to be moved per minute (radians per minute).

In G93 mode, an F-word is effective only in the block it is programmed (F is not modal in G93). Therefore, all blocks that call for axis motion when G93 is active, must contain an F-word or an error occurs.

G93 is modal. It remains active until cancelled by a G94 (feed per minute) or G95 (feed per revolution).

Figure 17.3
Inverse Time Feed (G93)



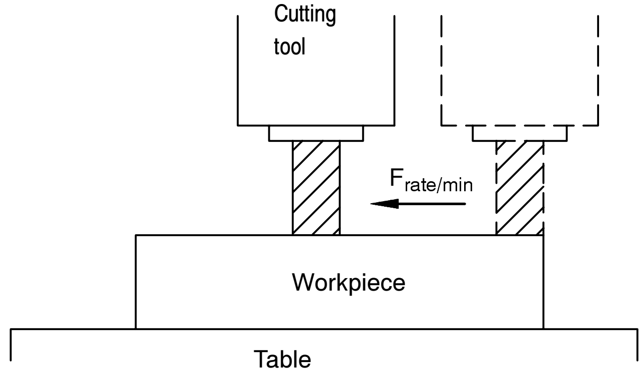
Feed Per Minute Mode (G94)

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 tool motion per minute program:

```
G94 F55.;
```

Figure 17.4
Feed Per Minute Mode (G94)



When changing from G95 to G94 modes, you must program a feedrate in the first G94 block.

Since the G94 code is modal, any F-word designated in any block after the G94 is considered a feed distance per minute until a G95 is executed.

Important: The controlling spindle determines which spindle per revolution value to use when calculating the feed per revolution.

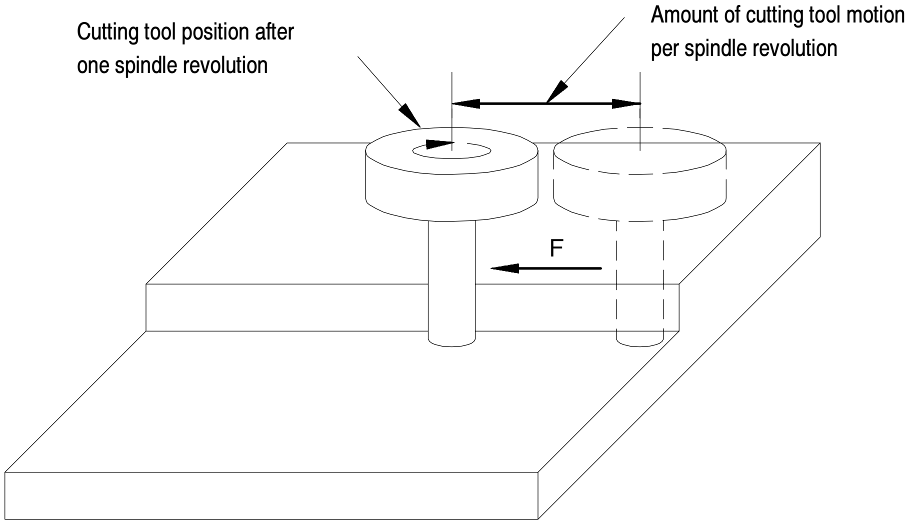
Feed Per Revolution Mode (G95)

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 spindle revolution, program:

```
G95 G21 F1.5
```

Figure 17.5
Feed Per Revolution Mode (G95)



When changing from G94 to G95 modes, you must program a feedrate in the first G95 block.

Since the G95 code is modal any F-word designated in any block after the G95 is considered a feed distance per spindle revolution until a G94 is executed.

Rapid Feedrate

Certain axis motions request from the control a rapid feedrate. For example, the G00 and some of the fixed cycles call for the rapid feedrate. The system installer specifies the rapid feedrate individually for each axis in AMP. When executing using a rapid feedrate the control drives the axes to maintain the fastest possible linear move while still remaining under (or at) each axis rapid feedrate.

When positioning mode is active, any programmed F-word executed by the control is stored as the currently active cutting feedrate. The rapid feedrate will not be affected.

Feedrate Overrides

<FEEDRATE OVERRIDE> Switch

Feedrates programmed in any of the feedrate modes (G93/94/95) can be overridden using the **<FEEDRATE OVERRIDE>** switch on the MTB panel. The **<FEEDRATE OVERRIDE>** switch has a range of 0-150 percent of the active feedrate, and can alter the active feedrate in 10 percent increments.

The feedrate override switch operates on the feedrate that is active. The active feedrate may be less than the feedrate you programmed if:

- the control has limited the feedrate to the maximum cutting feedrate that is defined in AMP

or

- the Acc/Dec feedrate clamping has limited the active feedrate to a value that the axis can decelerate from in the current block. See the section on Short Block Acc/Dec on page 17-21.

The control checks whether the feedrate resulting from the feedrate override switch setting exceeds the maximum cutting feedrate set in AMP. If it does, the feedrate is restricted to the AMP maximum.

An M49 (overrides disabled) causes the override amounts that are set by the switches or buttons on the MTB panel to be ignored by the control. With M49 active, the override switches or buttons for feedrate, rapid feedrate, and spindle speed are all set to 100 percent. They can be enabled by programming an M48 (overrides enabled).

<RAPID FEEDRATE OVERRIDE>

Use **<RAPID FEEDRATE OVERRIDE>** on the MTB panel to override the rapid feedrate for G00 mode in four increments:

- F1 -- percent value set in AMP by your system installer
- 25%
- 50%
- 100%.

Important: Normally this override is not active for any dry run motions (refer to chapter 7) unless otherwise specified in logic by your system installer.

Important: This override is also effective for jog moves that use the rapid feedrate (refer to jogging using the **<TRVRS>** button in chapter 4).

Feedrate Override Switche Disable

An M49 forces the override amounts that are set with the MTB panel to be ignored by the control. With M49 active, the overrides for feedrate, rapid feedrate, and spindle speed are all set to 100 percent. You can enable them by programming an M48 (overrides enabled). Refer to chapter 9 for details.

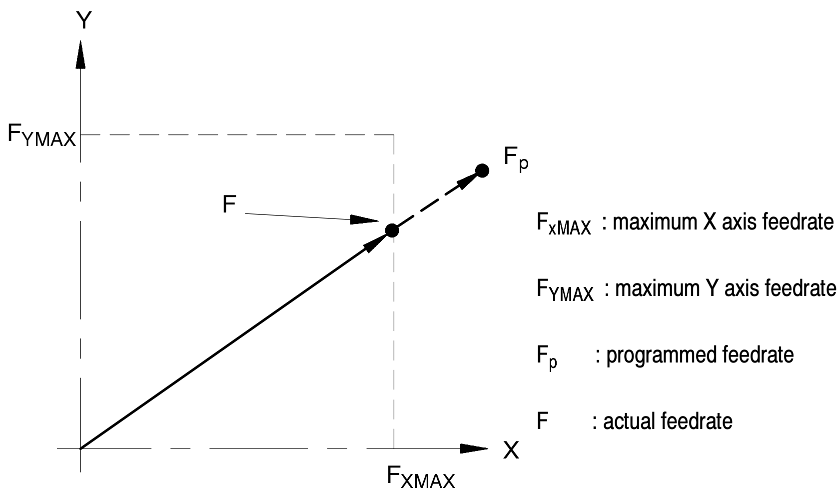
Feedhold

Your system installer can write logic to allow the activation of a feedhold state through the use of a button or switch. When activated, the control decelerates and holds 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.

Feedrate Limits (Clamp)

The maximum allowable speed for each axis is set in AMP. If any axis feedrate exceeds the maximum allowable speed for that axis the control automatically adjusts the feedrate to a value that does not cause axis speed to exceed its set limit.

Figure 17.6
Feedrate Clamp



The programmed feedrate, F_p in Figure 17.6, would cause the X axis feedrate to exceed its maximum feedrate, $F_x$max. consequently, the control uses F as the actual feedrate.

The maximum cutting feedrate limits the axis feedrate for any move controlled by a F-word. Feedrate override switch settings that cause the feedrate to exceed the maximum cutting feedrate will also be accordingly modified to keep the feedrate below or at the maximum cutting feedrate.

When the feedrate is “clamped” to a value below the programmed feedrate the control displays a flashing C next to the current axes feedrate. The displayed axis feedrate is the actual feedrate of the tool, not necessarily the programmed feedrate.

Feedrates to Control Torque Adaptive Feed (G25)

This feature will cause the servo to maintain a constant torque while cutting by varying the axis feedrate. The adaptive feed feature is helpful for machining processes where maintaining a constant feedrate is not as important as maintaining a constant cutting force. The adaptive feed feature also allows you to enter a feedrate range to prevent excessively low or high feedrates.

Your system installer must configure the adaptive feed feature for each servo using AMP.

When a move is programmed as an adaptive feed move, the control monitors the control's torque output being sent to the servo drive. If the output torque is too low, the control increases the cutting feedrate until the output torque reaches the programmed torque. If the torque output is too high, the control decreases the cutting feedrate until the servo output drops to your programmed torque.

Programming G25 Adaptive Feed

Program a G25 block as follows:

```
      X_
G25 ( Y_ ) Q__ F__ E__;
      Z_
```

Where:	Programs:
X, Y, or Z	Axis endpoint. Program the endpoint of the axis that is to be positioned using the adaptive feed feature. This endpoint can be programmed as either an absolute or incremental value (G90 or G91 mode). You can only program one axis in an adaptive feed block. You can not program axes that are positioned by more than one servo (dual or deskew axes).
Q	Desired Torque percentage. Enter an integer percentage of the selected servo's continuous rated torque as entered in AMP by your system installer. Valid ranges are from 1% to 150% of the servo's rated torque. Refer to your system installer's documentation for details on the rated torque of the servos in your system. **Important:** The torque amount applied by the servo is not the cutting force. It is the torque applied by the servo to the axis. You must calculate the equivalent cutting force based on your machine dynamics (motor rated torque, lead screw pitch, gearing, tool dimensions, etc...).
F	Maximum Feedrate. Enter the maximum feedrate that the axis is allowed to reach during the adaptive feed move. This F-word is the same as the modal feedrate and is used in following blocks. If you do not program an F-word the control will use the previously active feedrate. Program the F-word in either G93, G94, or G95 mode.
E	Minimum Feedrate. Enter the minimum feedrate that the axis is allowed to reach during the adaptive feed move. Program the E-word in either G93, G94, or G95 mode.

You must program the G25 block in G01 mode. Programming G25 in other cutting modes (such as G00, G02, or G03) will generate an error. The G25 command is not modal and must be programmed in all blocks that are to use the adaptive feed feature.

When the adaptive feed move starts, the control issues a command that moves the axis with the desired torque percentage Q. The system installer determines in AMP if ACC/DEC is used for the start and stop of an adaptive feed move. If ACC/DEC is used the control will attempt to ramp the velocity command to minimize the initial start up and decel shock.

Adaptive Feed Maximum Feedrate
When cutting under low to no load the servo may not be able to reach the programmed torque without exceeding your programmed F-word. In these cases, once the maximum servo feedrate is reached, the control allows the torque to drop below your programmed torque so as to not exceed the maximum programmed axis feedrate (F). The error “Adaptive Feed Max Limit” is displayed on the CRT.

Adaptive Feed Minimum Feedrate
When cutting under high loads the servo output torque may not be enough to keep the axis from slowing down below your programmed E-word. In these cases, once the minimum programmed feedrate is reached, the control allows the torque to go above its programmed limit to maintain the programmed minimum feedrate (E). The error “Adaptive Feed Min Limit” is displayed on the CRT.

Feedrate Override
Because of the precise feedrate control necessary to maintain a constant torque the feedrate override switch is disabled for any move that is programmed using the Adaptive Feed feature. If PAL is written to do so, you can use this switch only to request a feed hold which will halt axis motion. Cycle stop will also halt axis motion.

Block Retrace
You can perform block retrace on an adaptive feed block. When an adaptive feed block is retraced adaptive feed control is made in the opposite direction of the originally programmed block. Typically since no cutting is performed on the block retrace the cutting tool will move at the programmed maximum feed rate (F).

Polar Programming
You can not program a G25 block if the axis being programmed is in the current plane and the control is in polar programming mode (G16).

Special AMP-assigned Feedrates

You can select special feedrates that are assigned in AMP. This section covers the feedrates assigned in AMP for the external feedrate switch.

External Deceleration Feedrate Switch

Your system installer can install an optional external deceleration switch. 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 you activate this feature, any axis moves that are to take place at a cutting feedrate (G01, G02, G03, etc.) use a special feedrate assigned in AMP. 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 cutting moves cannot exceed the maximum cutting 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>** settings, and it operates as if the switches are set at 100 percent. Blocks that are programmed to move at the rapid feedrate are still executed in the rapid mode.

Use this feature to protect the machine from harsh or sudden stops. If a very high feedrate is active at the time that a hardware overtravel occurs, damage to the machine can result 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 tool if this feedrate is higher than that which the part should be cut.

ATTENTION: Your system installer can write logic to allow the operator to select the external deceleration feedrate at any time. This means that during normal automatic operation, you 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.

Automatic Acceleration/Deceleration

There are three types of axis acceleration/deceleration available. They are:

- Exponential Acc/Dec
- Uniform or Linear Acc/Dec
- S-Curve Acc/Dec

These are used to produce smooth starting and stopping of the machines axes and prevent damage to the machine resulting from harsh movements.

Your system installer determines the acc/dec parameter type (exponential or linear) for some manual motion types. To determine which motion types are configurable, refer to the following table. Refer to your system installer's documentation for more information about how your system is configured.

Refer to the table below to determine the type of acceleration/deceleration performed for manual motion and programmed moves.

Refer to the table below to determine the type of acceleration/deceleration performed for manual motion and programmed moves.

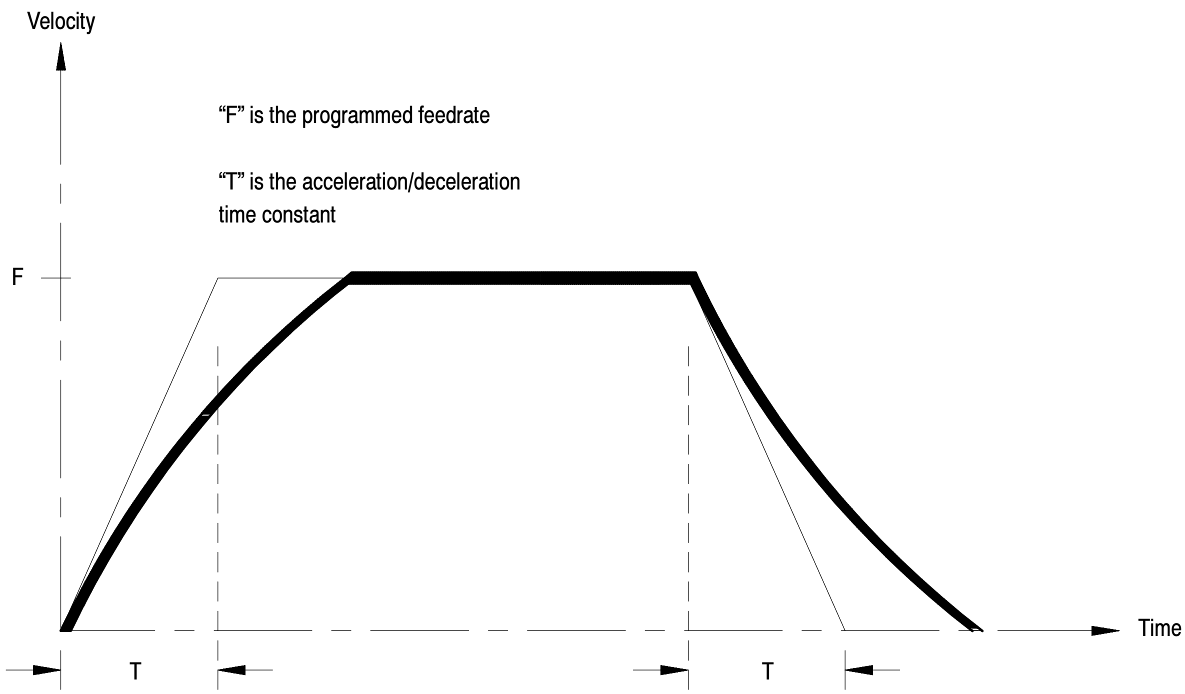
Table 17.A
Acc/Dec Type Performed with Manual Motion and Programmed Moves

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
Homing	✔			
All programmed moves except for G00 and exact stop			✔	
Manual continuous motion		✔		
Manual incremental motion		✔		
Logic axis mover		✔		
All moves programmed in G00 (positioning) mode				✔

Exponential Acc/Dec

To begin and complete a smooth axis motion, the 9/PC 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 17.7 shows axis motion using exponential Acc/Dec.

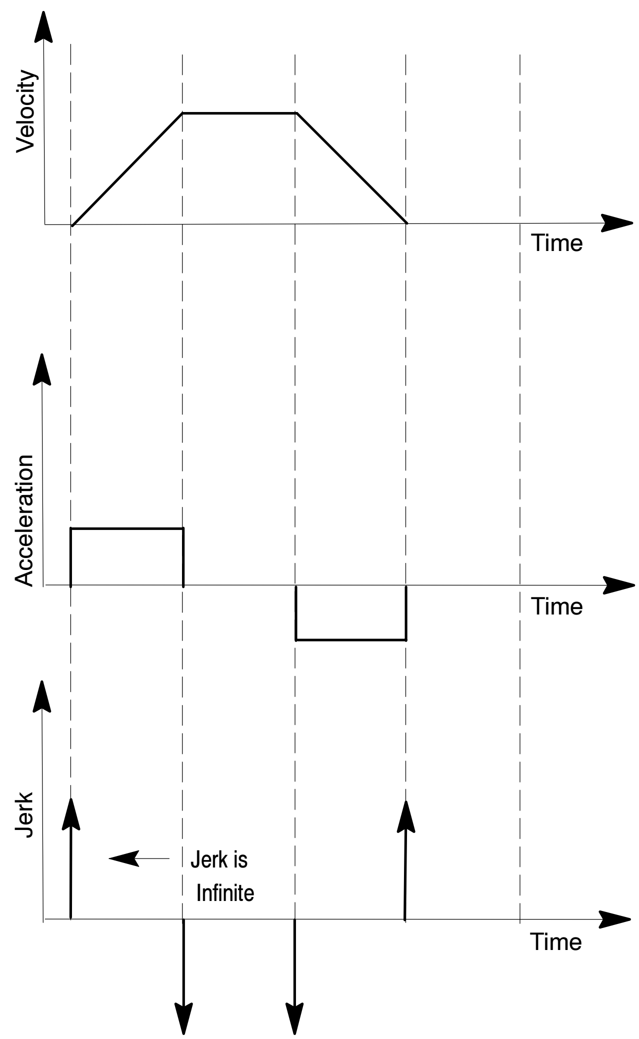
Figure 17.7
Exponential Acceleration/Deceleration



Linear Acc/Dec

Axis motion response lag can be minimized by using Linear Acc/Dec for the commanded feedrates. The system installer sets Linear Acc/Dec values for interpolation for each axis in AMP. Figure 17.8 shows axis motion using Linear Acc/Dec.

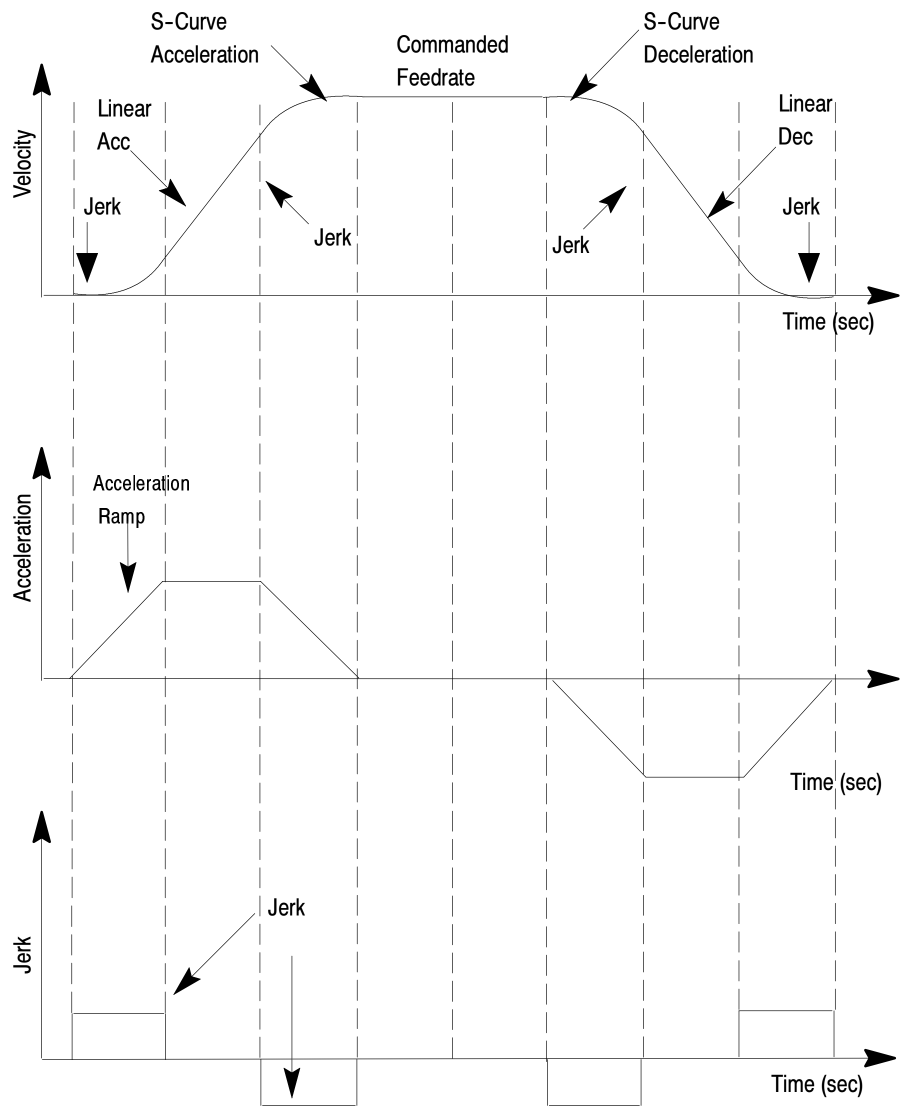
Figure 17.8
Linear Acc/Dec



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 17.9 shows axis motion using S-Curve Acc/Dec.

Figure 17.9
S-Curve Acc/Dec



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 17.A
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 :	Sets up :	Macros :
XYZ	G48.1	acceleration ramps for Linear Acc/Dec mode	#5631 to 5642
	G48.2	deceleration ramps for Linear Acc/Dec mode	#5651 to 5662
	G48.3	acceleration ramps for S-Curve Acc/Dec mode	#5671 to 5682
	G48.4	deceleration ramps for S-Curve Acc/Dec mode	#5691 to 5701
	G48.5	jerk limits	#5711 to 5722

Important: The allowable programmed range for the axis word depends on the configured format. Note that the axis word format also conforms to your current Inch/Metric settings. 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 1000,000 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 17.1
Allowable Programmed Range

```
#1 = 100000;
G48.5 X #1;
```

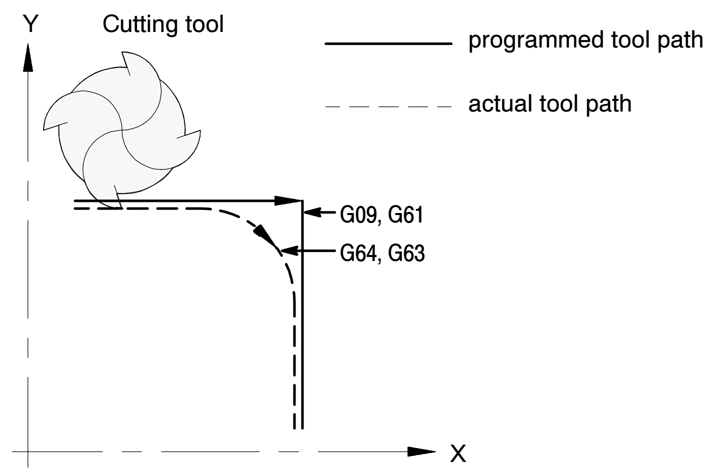
Important: The part program G48.n adjustments to Acc/Dec Ramps are not applied to jog moves. The AMPed Linear Acc/Dec mode rates are used when Manual Acc/Dec mode is linear.

Precautions on Corner Cutting

When Acc/Dec is active, the control automatically performs Acc/Dec to give a smooth acceleration/deceleration for cutting tool motion.

However, there are cases in which Acc/Dec can result in rounded corners on a part during cutting. In Figure 17.10 this problem is most obvious when the direction of cutting changes from the X axis to the Y axis. In this case, the X axis decelerates as it completes its move while the Y axis is at rest. As soon as the X axis reaches the AMP defined in-position band, the Y axis begins accelerating to make its commanded move. Since the Y axis begins motions before the X axis finishes, a slight rounding results.

Figure 17.10
Rounding of Corners



Use these G-codes to eliminate corner rounding:

Exact Stop (G09 -- nonmodal)

If a programmed motion block includes a G09, the axis moves to the commanded position, decelerates, and comes to a complete stop before the next axis motion block is executed. The G09 can be programmed in rapid (G00), feedrate (G01), or circular (G02/G03) motion blocks, but it is active only for the block in which it is programmed.

Exact Stop Mode (G61 -- modal)

G61 establishes the exact stop mode. The axes move to the commanded position, decelerate and come to a complete stop before the next motion block is executed. To cancel this mode, program G62, or G63.

Automatic Corner Override (G62 -- modal)

In cutter compensation mode (G41/G42), the load on the cutter increases while moving inside a corner. If the G62 automatic corner override mode is active, the control automatically overrides the programmed feedrate to reduce the load on the cutter. To cancel this code, program G61 or G63.

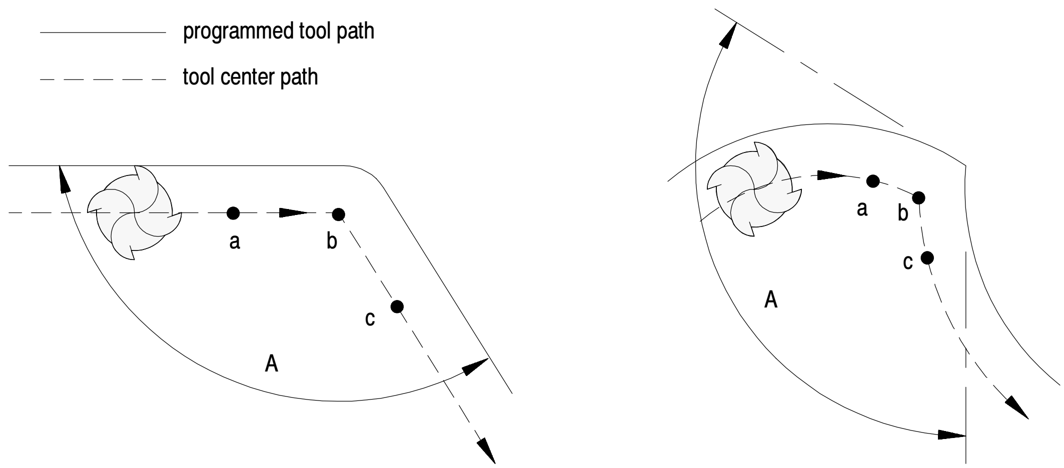
Tapping Mode (G63 -- modal)

In the G63 tapping mode, the feedrate override value is fixed at 100 percent, and a cycle stop is ignored. Axis motion commands are executed without deceleration before the end point. The program proceeds to the next block without checking in position status, similar to the operation of G64. To cancel this code, program G61 or G62.

Cutting Mode (G64 -- modal)

G64 establishes the cutting mode. This is the normal mode for axis motion and is generally selected by your system installer as the default mode active on power up. Block completes when the axes reach the interpolated endpoint. To cancel this code, program G61, G62, or G63.

Figure 17.11
Automatic Corner Override (G62)



When the corner angle, A, is smaller than angle Ap set in AMP, the programmed feedrate is overridden from point “a” to point “b”, and from point “b” to point “c”. The control compares angles A and Ap.

The system installer sets these values in AMP:

- angle Ap in AMP in 1 degree increments within a range of 1-90 degrees
- range in which the automatic corner override function is active -- essentially, the values of “a” and “c” in absolute distance measured along the tool path for “b”
- override value in 1-percent increments within a range of 1-100 percent.

To use an exact stop function while the automatic corner override mode (G62) is active, use the G09 instead of the G61. This is because G61 and G62 belong to the same G modal group and cancel each other if programmed. Be aware that G09 is nonmodal.

Spindle Acceleration (Ramp)

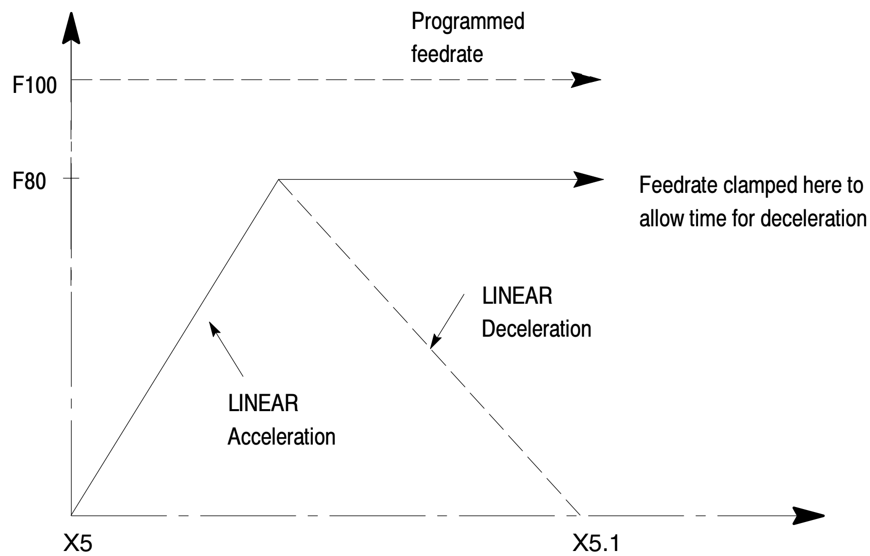
Your system installer can change the rate at which a spindle is accelerated. AMP allows the option of either a 20 ms ramp (2 ms intervals) or an immediate step in spindle speed. By writing the appropriate logic your system installer may also in effect generate a spindle “ramp” for even smoother spindle acceleration. Refer to documentation prepared by your system installer.

Short Block Acc/Dec Check (G36, G36.1)

In the control's default mode (G36), the Acc/Dec feature in some cases limits the feedrate below your programmed feedrate. This occurs when the length of the block is relatively short relative to the length of time necessary to properly decelerate the axis to a stop.

In the default mode (G36), the control limits the feedrate of any block to the maximum speed from which it can properly decelerate all the axes to a stop before that block ends. For example, consider the following velocity profile of a block moving from X5 to X5.1.

Figure 17.12
Feedrate Limited Below Programmed Feedrate to Allow Deceleration Time



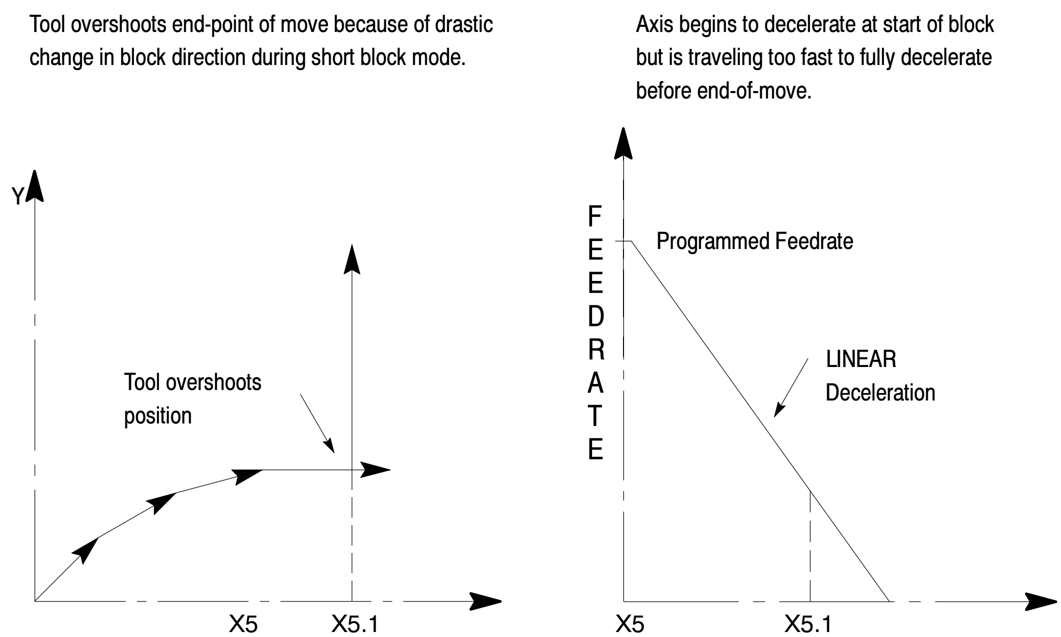
Normally this causes no problem. However, in cases where a series of very short axis moves in separate blocks exist, this limitation to the feedrate can cause finish problems as well as increased cycle time. Figure 17.12 shows the velocity profile that would result from a series of short X axis moves from 4.8 to 4.9 to 5.0 to 5.1 to 5.2.

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 these 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 17.13) . The position Z5.1 is overshot and the axis would have to reverse direction to reach proper position.

Figure 17.13
Drastic Change in Direction While in Short Block Mode



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 depends on the Acc/Dec ramp configured for your machine.

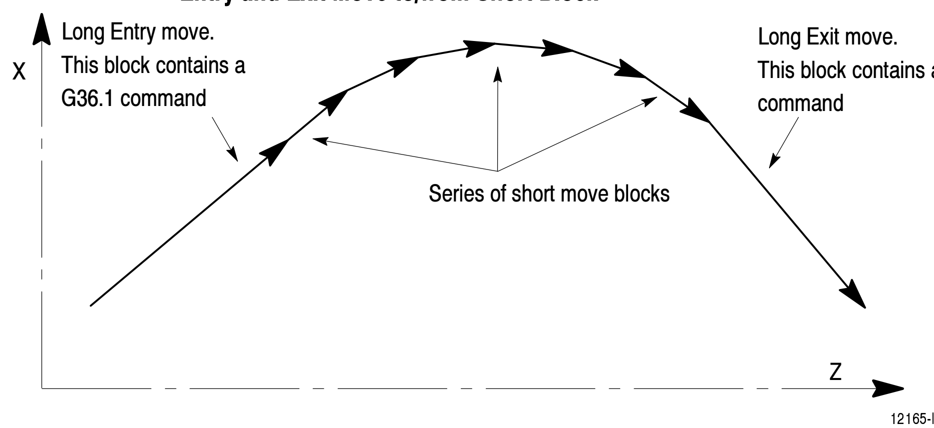
- G36 – Short Block Acc/Dec clamp Enable
- G36.1 – Short Block Acc/Dec clamp Disabled

G36 is the default mode, and it 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 17.14 shows the recommended entry and exit moves for short block Acc/Dec clamp disable mode.

Figure 17.14
Entry and Exit Move to/from Short Block



END OF CHAPTER

Dual Axis Operation

Overview

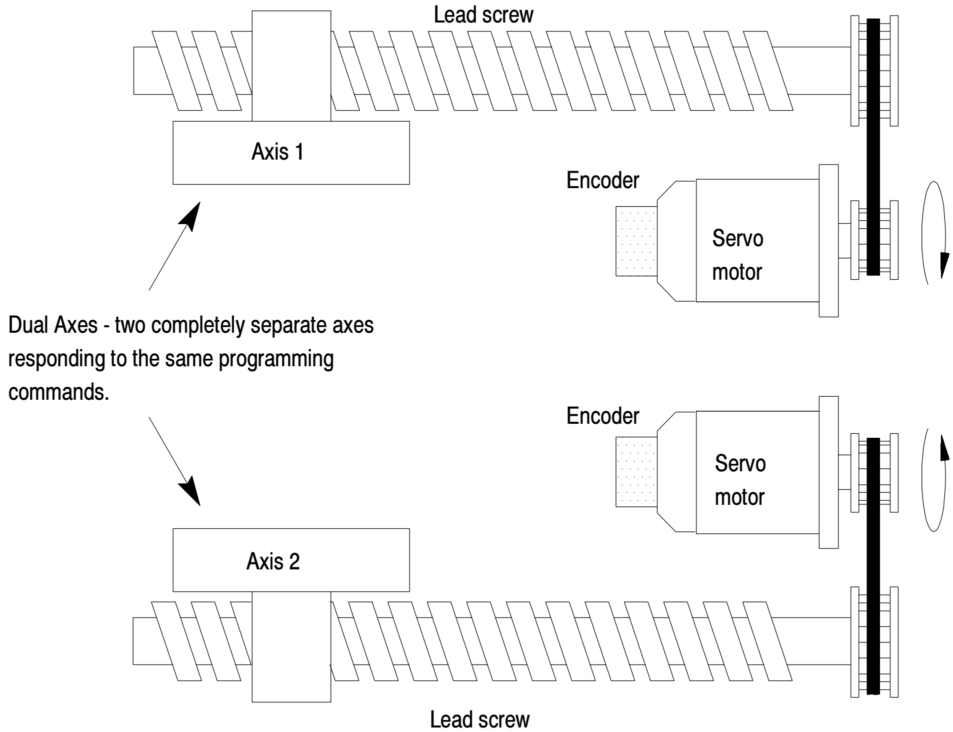
The Dual Axis feature lets the part programmer simultaneously control multiple axes while programming commands for only one. It differs from the split axis feature of the 9/PC control in that the split axis feature is used to control a **single axis** positioned by two servo motors.

This chapter reviews the following major dual axis operations:

The dual axes feature is especially useful for lathes with dual turrets and other machines running with parallel cutting tools. Figure 18.1 shows a typical configuration for dual axes.

Implementation of the dual axis feature can require significant logic modification as well as proper AMP configuration. The dual axis feature is an option. Refer to your system installer's documentation to see if the dual axis option has been purchased for your machine.

Figure 18.1
Dual Axis Configuration



The 9/PC 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.

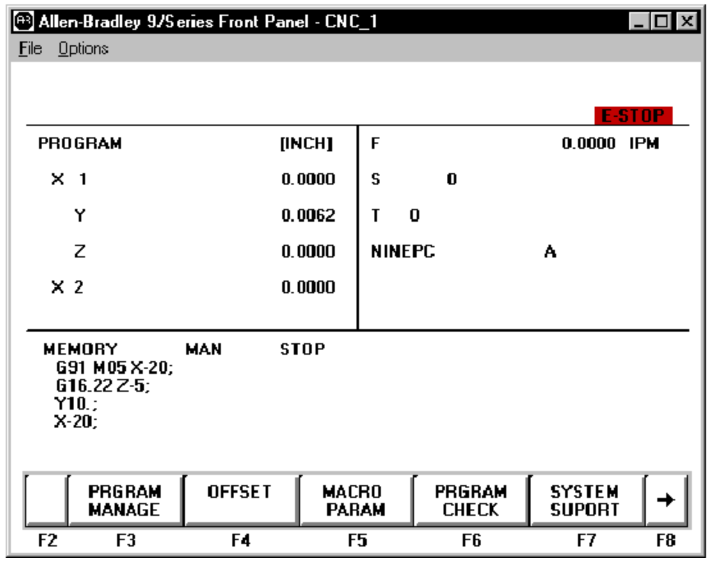
All axes that make up a dual group reach end-points at the same time. This requires that all axes that make up a dual axis group share the same feedrate parameters, acc/dec ramps, and other axes specific data for the group.

This section requires that you understand these 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 18.2 shows the position display for a system that contains a dual axis group containing two axes with a master axis name of X. Whether or not all axes of a dual group show up on the position display is determined in logic by your system installer.

Figure 18.2
Axis Position Display for Dual X Axis



Parking a Dual Axis

This feature allows you or the programmer to disable selected axes of the dual group. Any axis that is a member of a dual axis group can be parked. Axes in the dual group may be parked simultaneously. If all axes in the group are parked, no motion can take place in the dual axis group.

Once parked, no motion is allowed on the parked axis. Programmed and jog commands (including any homing requests) made to the dual axis group are ignored by the parked axes.

Axes in the dual group may only be parked or unparked when the control is in cycle stop and end-of-block state. The control cannot be in the process of completing any jog request or logic 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: Be careful when an axis is unparked. Any incremental positioning requests you make 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 you make the next command the dual axis group be an absolute command to **realign** the axes in the dual group to the same position.

Perform an axis park in a dual group through logic. Refer to your system installer's documentation for details on how axes are parked.

Important: Some systems can have special parking requirements when homing axes in a dual group. Refer to page 18-4, *Homing a Dual Axis*, for details on homing dual axes.

Homing a Dual Axis

There are two methods to home axes in a dual axis group. Your system installer determines through logic which method is available. The two methods are:

- home each axis in the dual group individually
- home all axes in the dual group simultaneously

Both of these homing methods can be available for automatic (G28) as well as manual homing operations.

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 your system installer's documentation for details on these speeds and locations.

Homing Axes Individually

This method requires that each axis be homed individually. When a manual home operation is performed, a home request must be made to each axis in the dual group on an individual method. Refer to chapter 4 for details on how to request a manual home operation.

When you use automatic homing (G28), the axes must be homed one at a time. This is accomplished by parking all other axes in the dual axis group except the axis that is to be homed and requesting that the AMP-assigned master axis name be homed in the G28 block. Once homed, that axis should be parked, the next axis to be homed should be unparked, and the homing procedure repeated. Refer to chapter 13 for details on how to request an automatic homing operation (G28).

Homing Axes Simultaneously

This method allows a request for all axes in the dual group to be homed at the same time. This does not mean that all axes reach home at the same time. Keep in mind that your system installer can define different feedrates and different home positions for each axis in the dual group.

With proper logic 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 you use this homing method, all unparked axes home together. Refer to chapter 4 for details on how to request a manual home operation and chapter 13 on how to request an automatic home operation (G28).

Programming a Dual Axis

You can position axes in a dual axis group using any of the normal programming or manual motion operations. Only the master axis name can be requested to position a dual axis. 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 cutting, external decel, and rapid feedrate limitations for each axis in a dual axis group. The control uses the slowest feedrate for each of these features from any axis in the dual axis group.

Special consideration must be given when programming 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 logic. Refer to the system installer's documentation and chapter 13 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 12 for details.

Important: You can use the Logic Axis Mover feature if it is necessary to position dual axis group members separately without requiring any parking. Refer to the *9/PC Logic Reference Manual* and the system installer's documentation for details.

Invalid Operations on a Dual Axis

Table 18.A lists the features that are not compatible with dual axes. If you must execute one of these features on a dual axis, only the AMP master axis can be used. **All other axes in the dual group must be parked.** Refer to your system installer's documentation to determine which axis has been assigned in AMP as the master axis.

Table 18.A
Features Not Compatible With Dual Axes

G-code	Feature
G16	Polar Programming
G16.1	Cylindrical Interpolation
G31-G31.4	External Skip Functions
G37-G37.4	Automatic Tool Gauging Skip Functions
G38, G38.1	Probing Cycles
G74.1, G84.1	Solid Tapping
G76	Boring Cycle (with spindle shift)
G87	Back Boring Cycle
G88.1-G88.6	Pocket Milling Cycles
G89-G89.1	Irregular Pocket Milling

Offset Management for a Dual Axis

Give consideration 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 dual axis operation 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. See chapter 3 for implementation details about the offset you are using.

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 you specify a G52 offset value 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 takes 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.

Different G92 offset values can be created for each axis if necessary. This is accomplished by performing a jog offset or by using the logic 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 logic, you must specify which axis in the dual group to set zero. When the set zero operation is performed on an axis, the current axis location becomes the new zero point of the coordinate system.

Cutter Compensation

Only one tool diameter can be active at any one time. Any offset created by cutter compensation affects all axes in the dual group.

Tool Length Offsets

The system installer must select one of the dual axis group members as the tool length axis in AMP if separate tool length offsets are to be used for a dual group. Once one axis of the group has been assigned as the tool length axis, all axes in the group may have independent tool length values assigned to them.

Assuming tool length offsets are valid on your dual axis, their activation is the same as the tool length offset on nondual axes. Refer to chapter 19 for details.

The offset function is selected with G43, G44, and G49. The offset value to activate is selected with an H word. This H word is the offset number and used for all axes in the dual group. The tool length offset values called by this H word, however, may be assigned individually. These values may be assigned either :

- manually through the tool wear and geometry tables as discussed in chapter 4.
- automatically through programming the correct G10 codes as discussed on page 10-7.

Assigning Tool Length Offsets Manually
For dual axes, extra tool length offset tables have been provided, one for each member of the dual axis group. By pressing the **{NEXT SELECT}** or **{PREV SELECT}** softkey, you can select which axis you are assigning length offset values in the dual axis group. Each member of the dual axis group is represented by the master axis name followed by a number indicating which axis in the group is active. Note that you can not activate the tool length offset using a softkey on a dual axis member.

```
(softkey level 3)
```

↑	SEARCH NUMBER	REPLCE VALUE	ADD TO VALUE	CHANGE OFFSET	MORE OFFSET	→
F2	F3	F4	F5	F6	F7	F8

↑	MEAS- URE	INCH/ METRIC	PREV AXIS	NEXT AXIS	COPY OFFSET	→
F2	F3	F4	F5	F6	F7	F8

If the same offset is desired for different dual axes, use the **{COPY OFFSET}** softkey to copy the offset values from one axis to another (refer to chapter 4 for details on using this feature).

Assigning Tool Length Offsets Through Programming G10

For dual axes, additional programming for the G10L10 and G10L11 codes is available when a dual axis is the tool length axis. In place of the normal G10 block which assigns tool length data using an R-word, a dual axis used as a tool length axis must program length offset data using the following format:

```
G10L10P__A__B__C__;
G10L11P__A__B__C__;
```

where A, B, and C are the names the system installer assigned each axis in your dual group in AMP. A, B, and C are used to assign tool length data in place of the R word. Refer to the system installer's documentation for details.

END OF CHAPTER

Tool Control Functions

Chapter Overview

This chapter describes these tool control functions:

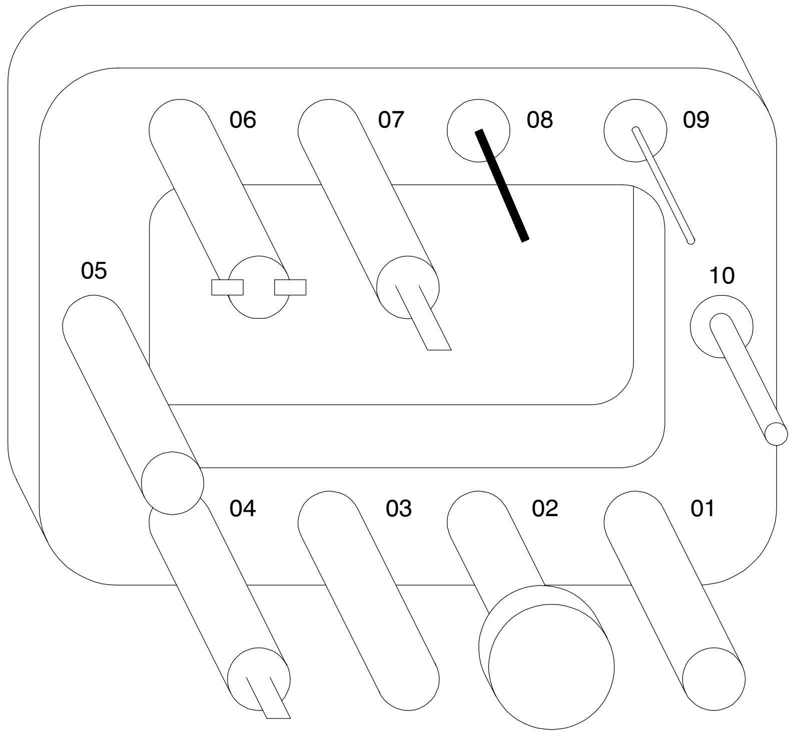
- **Programming a T-word** -- Different formats available for selecting a tool number and tool offsets
- **Tool length offsets** -- Compensate for the difference between the tool length assumed while programming, and the actual length of the tool used for cutting. This feature can offset up to 4 axes.
- **Programming alterations of the offset tables** -- Modify or enter data within the offset tables by programming the correct G10 code.

Enter tool length offsets and cutter compensation data in to the control's offset table (refer to chapter 3). Once entered, call for a particular set of data by programming an H- (tool length data) or a D- (tool diameter data) word that corresponds to a offset number from the offset table.

Programming a T-word

A workpiece usually requires different kinds of cutting processes, and usually there are cutting tools that correspond to each process. The cutting tools are typically stored in a tool magazine and are assigned tool numbers (see Figure 19.1).

Figure 19.1
Typical Mill Tool Magazine



A T-address followed by a numeric value programs a tool selection (or tool group number - refer to page 19-18 on tool life management). The system installer determines in AMP how a tool change operation is programmed. There are four different options available. They are:

Return tool in M06 - When this method the T-word to activate is programmed in a block that does not contain an M06. The T-word is stored until some later block that contains an M06. When the M06 is executed the currently active tool is replaced with the last tool number programmed with a T-word. It is required with this method that the tool number (or group number) that is being replaced as the active tool is programmed in the block that contains the M06 command. If the M06 block does not contain the previously active tool number/group number, or if the wrong tool number/group number is programmed with the M06, the control will generate an error.

Next tool in T-word - This method is identical to the "Return tool in M06" method with the exception that the block containing an M06 can not contain a T-word. It is not necessary to program the previously active tool number/group number in the M06 block.

M06 Required - This method defines that a tool is only activated in an M06 block. A T-word that is programmed by itself becomes the next tool activated at an M06 block. Programming an M06 by itself activates the next tool. If a T-word is programmed in an M06 block that T-word is used as the active tool and any other unactivated T-word is discarded.

Activate Tool in T-word - For this method no M06 needs to be programmed to change tools. A tool change occurs immediately when the T-word is executed.

When the correct M06 block or T-word block that will execute a tool change is programmed the control outputs a tool selection signal to a tool changer. The tool changer should perform a sequence of operations to deliver the proper tool in response to the tool selection signal. For example, to select a cutting tool that is assigned tool number "03", write "T03" in the part program.

Since tool changers vary in style, size and function,the system installer is responsible for specific implementations through PAL. Refer to the PAL programmers manual and the manual supplied by the system installer for more details.

Important: When changing cutting tools it is usually necessary to change the tool offset at the same time. This is done with an H- or a D-word. For details refer to chapter 9.

Important: When the MISCELLANEOUS FUNCTION LOCK feature is activated, the control displays M-, B-, S-, and T-words in the part program with the exception of M00, M01, M02, M30, M98, and M99. This feature is activated through an optional switch installed by the system installer.

Tool Length Offset Function (G43, G44, G49)

To cut a workpiece using the bottom face of the cutting tool, it is more convenient to write the part program assuming that the gauge line of the tool holder equals the bottom face of the tool.

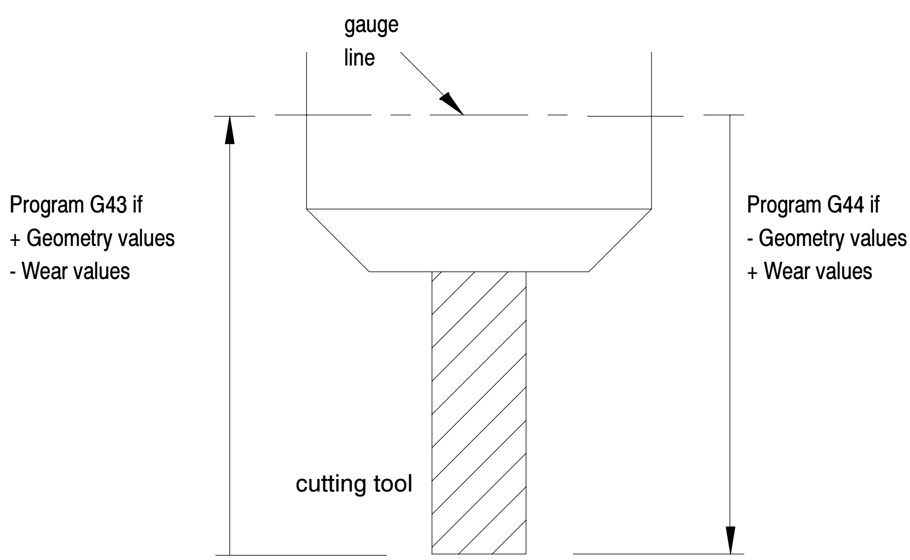
The term "gauge point" defines the precise point on the chuck or tool holder from which all programmed tool paths originate. Offsets refer to the distance from this gauge point to the edge of the tool that contacts the part being cut.

But when a cutting tool is set in the spindle, its bottom face is not at the gauge line. To cut the workpiece to the proper dimensions, offset the tool path by an amount that equals the difference between the gauge line and the bottom face of the cutting tool.

The control offers a function called tool length offset for offsetting tool paths. The tool length offset is usually equal to the difference between the bottom face of the tool and the gauge line. Put the tool length offset into memory in advance. This function lets the control use the same program to produce the same workpiece regardless of the length of the cutting tool.

Figure 19.2 illustrates the reference points used for deriving a tool length offset.

Figure 19.2
Tool Length Offset



There are three G codes, G43, G44 and G49, that are used when programming tool length offsets. To know when to use them, see below:

G43

If the sum of the tool geometry and the tool wear is a **positive** offset value, program G43.

For example:

If the values for tool offset no. 1 are:

```
Tool Geometry              +3.0000
Tool Wear                  -0.1000
The tool offset is:        +2.9000
```

G44

If the sum of the tool geometry and the tool wear is a **negative** offset value, program G44.

For example:

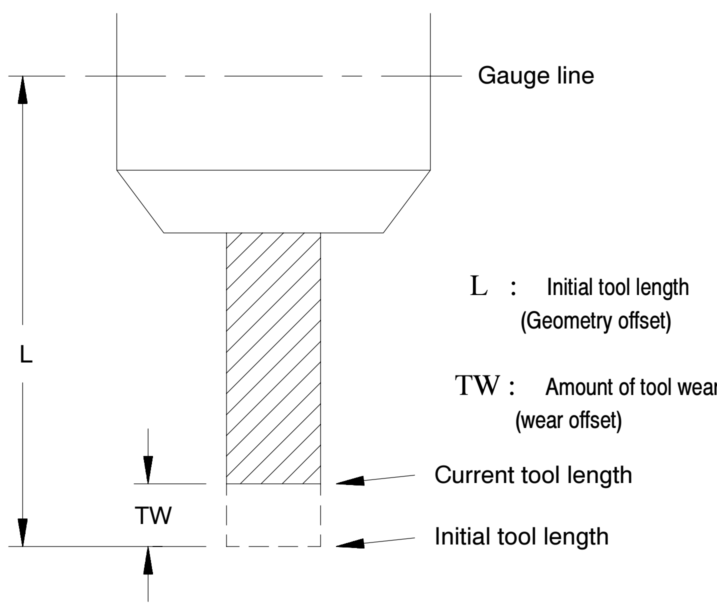
If the values for tool offset no. 1 are:

```
Tool Geometry                -3.0000
Tool Wear                    +0.1000
The tool offset is:          -2.9000
```

G49

To cancel the tool length offset function, program G49.

Figure 19.3
Tool Offset Data



These G codes are modal, which means they are active from the program block that they have been entered to either the end of the program or until canceled. (They also belong to the same modal group.) However, the system installer must select one of these G codes in AMP. This G code would then be active during the entire program or until canceled. This manual assumes that G49 has been selected in AMP.

Use these formats for programming G43 or G44:

```
G43H__;
G44H__;
("H" is the tool offset number.)
```

G43 or G44 does not have to be programmed with an H-word in the same block, or vice versa, in order for a tool offset to be made active. But the tool offset will only be activated at the time both a G-word and H-word are active.

Important: If using the tool life management feature, programming a H-word may not be necessary. (refer to page 19-18 for details on tool life management).

Depending on how the system installer has configured AMP, tool offsets may remain active after "end of program commands" (M02 or M30) are executed, a "control reset" is performed or E-Stop is reset.

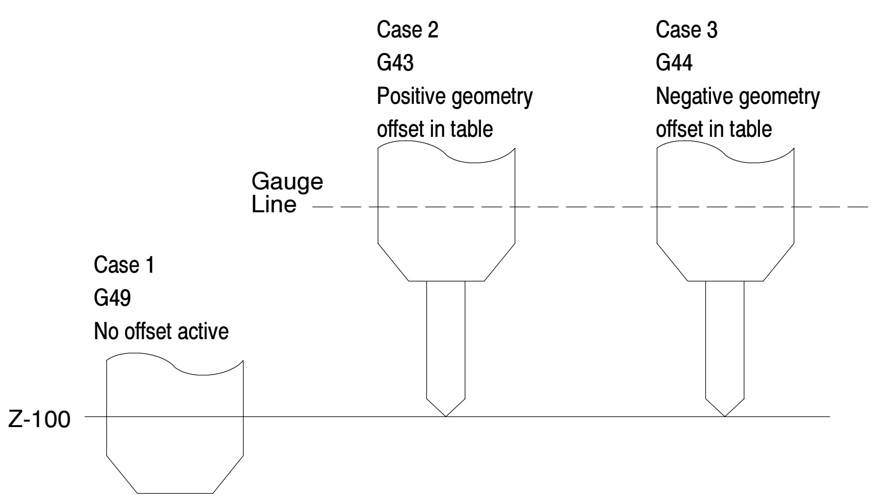
Example 19.1 and Figure 19.4 illustrate offset program blocks and how they affect tool position.

Example 19.1
Programming G43 or G44

```
Assume H01 offset data to be 15
Assume H02 offset data to be -15
```

G91G00Z-100.G43H00;	(Case 1)
G91G00Z-100.G43H01;	(Case 2)
G91G00Z-100.G44H02;	(Case 3)

Figure 19.4
Results of Example 19.1



Offset “H00” in the offset table is always equal to a value of zero, but does not cancel the tool offset mode like G49. HOO cancels H-words. Programming a G49 will not change the current H-word to H00. Example 19.2 illustrates this.

Example 19.2
Modal G43, G44 and Modal H-words

Program Block	Comment
N1G00G90;	
N2G43 H01;	G43 mode, H01 offset
N3 ;	G43 mode, H01 offset
N4H02;	G43 mode, H02 offset
N5;	G43 mode, H02 offset
N6G44	G44 mode, H02 offset
N7;	G44 mode, H02 offset
N8G49;	Offset mode cancelled, H02 offset
N9G43;	G43 mode, H02 offset
N10H0;	G43 mode, No offset value
N11G44;	G44 mode, No offset value

Important: Whenever a new tool length offset is initiated or cancelled, the block that changes the offset must be a linear block (G00 or G01). In the above example, blocks N2, N4, N6, and N8 must be linear blocks.

Activating Tool Length Offsets

The system installer has the option in AMP to determine exactly when the geometry and wear offsets will take effect and when the tool position will change to the new position. This manual makes the assumption that the system is configured to immediately shift the coordinate system by the geometry and wear amounts, and delay the move that will reposition the tool to the same location in the current work coordinate system. Refer to documentation prepared by the system installer to determine the application in a specific system.

Provided the system is configured as described above, the control activates a tool offset as described in the following steps.

1. The control reads a block that activates or deactivates a tool length offset. This may be a G43, G44, or G49 block or simply a block that contains an H-word (see Example 19.2).
2. The control immediately shifts the work coordinate system the amount of the tool geometry and tool wear amounts called by the H-word. The tool position display will change reflecting this shift. The absolute position display does not change. The offset is interpolated into the next move that generates axis motion on the offset axis, unless you are in incremental mode. If you are in incremental mode the offset is not interpolated into the next move of the axis. Example 19.3 shows how the move is generated in incremental and absolute modes when the tool offset programmed as 3.

Example 19.3
Immediate Shift/Delay Move in Incremental and Absolute Modes

Absolute Mode		Incremental Mode	
G00Z0	Rapid mode	G00Z0	Rapid mode
G90	Absolute Mode	G91	Incremental mode
T01	Activate tool 1. Program display changes Z position to -3.	T01	Activate tool 1. Program display changes Z position to -3.
Z1	Axis moves to +1	Z1	Axis moves to -2

If "immediate" is chosen for the move, the control generates a linear move that will reposition the cutting tool to its old coordinate position in the work coordinate system. This block is executed in the same block that calls for the offset. If axis words are present in the block that activates or deactivates a tool length offset, the control will add this generated move to the programmed move.

Important: Any block that activates or deactivates a tool length offset must be programmed in linear mode (G00 or G01) when executed. If a tool change is made in the circular mode, no axis motion may take place in the block changing the tool offset. The offset must be activated in a block with no axis words.

Tool Length Offset (TLO) Axis Selection (G43.1, G44.1)

When you program one of these TLO axis-select G-codes, the axis programmed in the block becomes the axis to which the tool length offset is applied; the forced axis name replaces the AMP-defined TLO axis. Otherwise, these G-codes have the same effect as the G43/G44 codes.

Important: The G43.1 and G44.1 blocks are motion blocks; therefore, the axis will move to the position (and offset) commanded in the block.

You must program a G49 before you can switch to a new TLO axis with G43.1 or G44.1. You must also remove the offset from any previously active TLO axis.

The axis that the control is currently using for tool length offset calculations is shown in reverse video on the tool offset screen. An axis shown in reverse video does not mean the offset is active, only that the axis is the current tool length axis. An asterisk indicates the currently active offset value.

The G10L10 and G10L11 offset table-modifier blocks now recognize axis names to let you modify TLO data for axes other than the currently active offset axis.

These conditions cause the AMP-defined TLO axis to become the active axis, replacing the axis that was selected with a G43.1 or G44.1:

- you perform a control reset
- the AMP parameter *Cancel Tool Offsets on M02/M30* has been set to *yes* by the system installer, and an M02 or M30 block is executed

Switching planes with a G17/G18/G19 does not change the active TLO axis; the active TLO axis can be switched only by programming a G43.1/G44.1 with an axis name.

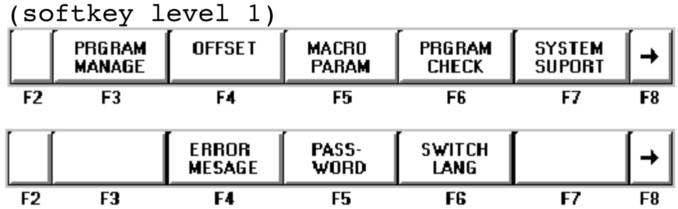
Copying Tool Length Offset Tables

Each selectable tool length axis has its own tool offset table and its own set of offset values. For example, if you select the X-axis as the tool length offset axis, you must enter an offset value for the X-axis for those tool offsets to be used on X.

To copy the offset values from one axis to another, follow these steps:

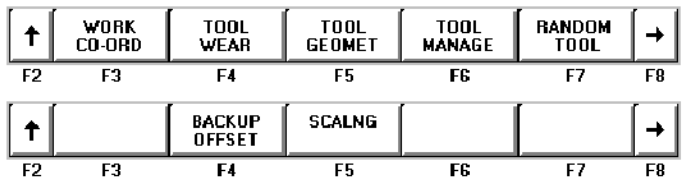
1. Press the **{OFFSET}** softkey.

(softkey level 1)



2. Press the **{TOOL WEAR}** or **{TOOL GEOMET}** softkey, choosing the table from which you want to copy.

(softkey level 2)

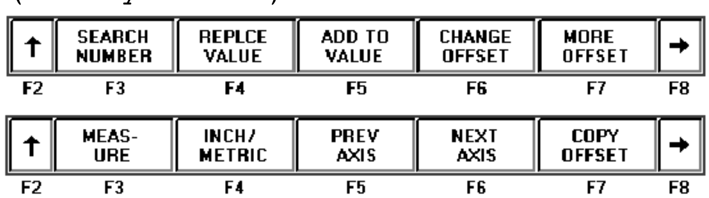


3. Choose the axis whose values you want to copy.

If you want to move:	Press this softkey:
forward	**{NEXT AXIS}**
backward	**{PREV AXIS}**

4. Press the **{COPY OFFSET}** softkey.

(softkey level 3)



"COPY (SOURCE, DESTINATION):" appears.

5. Enter the axis letter *from* which the data is coming, then a comma, and then the axis letter *to* which the data is going. For example,

```
COPY (SOURCE, DESTINATION): X,Z
```

copies the offset data from the X axis to the Z axis for all offset numbers.

Random Tool

Use the random tool feature to speed up production by saving cycle time when a tool is returned to the tool changing device. This is done by allowing the tool changer to randomly return the cutting tool to the most convenient pocket in the tool changing device. The control remembers what pocket the tool is returned to, and it is able to call the same tool from the new pocket at any time.

Important: This feature can be used with normal tool selection or the tool life management feature.

This feature has no effect on tool length offsets or cutter compensation. These features must still be activated correctly as described in their individual sections.

The random tool feature automatically decides the pocket that contains the requested tool based on the information in the pocket assignment table. If the requested tool has not been assigned to a pocket, the control generates an error.

Based on the current pocket number, which is maintained by logic, the control tells logic which pocket to move to, and how far, and in which direction to move. The control also tells logic where the tool currently in use can fit in the tool turret.

Important: This feature is very logic dependant. Before using this feature make sure your system installer has written the logic program to allow the use of Random Tool.

The control automatically updates the tool pocket assignment table when you make tool changes. The control indicates to logic the best location to return the tool to. Logic then decides where the tool gets placed in the tool holder. The pocket that is vacated by the new tool is marked as empty.

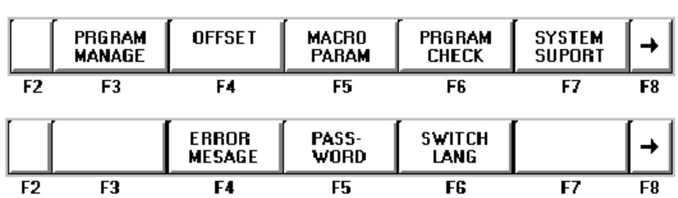
Manually Entering Random Tool Data

Data can be entered into the random tool table either manually, as described here, by programming, or by running a backup program of the tool data. These other methods are described later in this section.

To manually enter the random tool data, follow these steps:

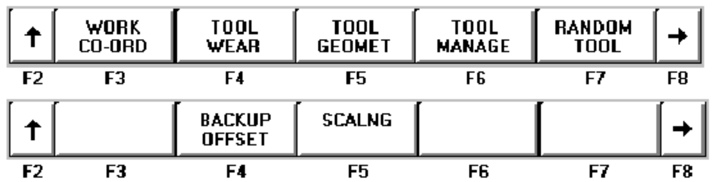
1. Press the **{OFFSET}** softkey.

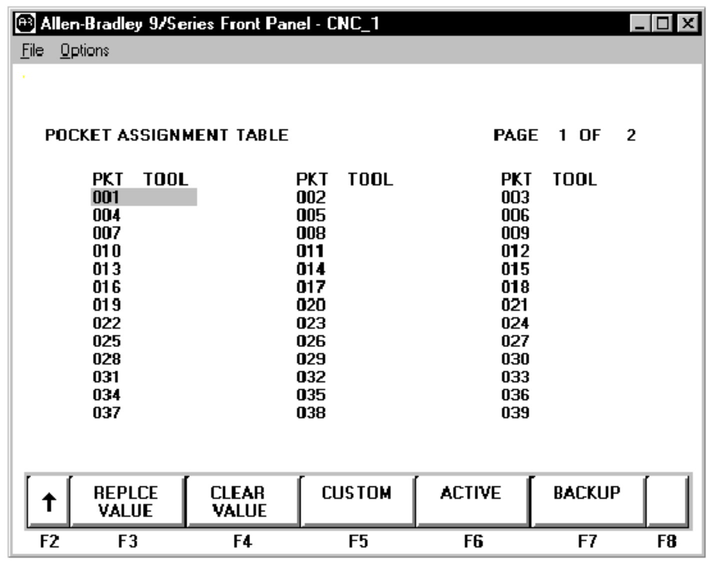
```
(softkey level 1)
```



2. Press the **{RANDOM TOOL}** softkey. The pocket assignment table screen is displayed as shown below. This screen shows the current tool to pocket assignments. Your system installer designates the number of tool pockets available on a system in AMP.

(softkey level 2)





The columns labeled PKT give the tool changer pocket numbers. The columns labeled TOOL give the tool number of the tool in the corresponding pocket. Pockets with no tools assigned to them show no information next to the pocket number. Pockets with tools shown as XXXX indicate that a custom tool (tool that requires more than one pocket) has been assigned to use that pocket.

3. Move the cursor to the pocket number with the assignment or change is to be made. Press the up, down, right, or left cursor keys on the operator panel. Move the cursor full pages by holding down the **[SHIFT]** key while pressing the up or down cursor keys. The selected pocket appears in reverse video.

Important: If random tool is not to be used for your system, make sure that none of the tool pockets have tool numbers assigned to them.

4. To modify tool data there are three choices:

- To remove a tool assigned to a pocket, press the **{CLEAR VALUE}** softkey. The selected tool is deleted from the table.
- To enter a tool number for the pocket, press the **{REPLCE VALUE}** softkey, key in the new tool number, and press the **[ENTER]** key. The old tool value is replaced with the new value just keyed in.
- To enter a custom tool (a tool that requires more than one tool pocket) enter the tool number of the custom tool in the pocket that is to be used as the “shaft pocket”. The shaft pocket is where the tool changer is positioned when the particular custom tool is to be used. Enter the number of pockets needed (to a max of 9), a comma, followed by the position of the shaft pocket in this group of pockets. Press the **[ENTER]** key enters the data into the table.

The screen shows XXXX for the tool number of any pockets that have been configured as part of a custom tool, and show the tool number in the pocket where logic is told to go in order to find the tool.

For example, in the pocket assignment screen, pocket number 19 is a shaft pocket for custom tool number 6. This custom tool requires 3 pockets, pockets 18, 19, and 20. When the **{CUSTOM}** softkey was pressed for pocket number 19, a value of 3,2 was entered.

Programming Random Tool Data

This feature is available so that it is not necessary to always manually enter the data into the pocket assignment table. By programming the correct G10.1 blocks all information may be entered into the tool pocket table. Note the control may automatically generate a G10.1 program by using the backup softkey as described later in this section.

Important: G10 blocks cannot be programmed when TTRC is active.

Programming of random tool data can only be done on a tool pocket if data has not already been configured for that pocket. If you need to make changes to a tool pocket that already has a tool assigned to it, you must either clear and re-load the entire random tool table as discussed below (you can not use a G10.1 to clear individual pocket data), or use the softkeys to manually access the random tool table and change the data using the keyboard.

Clearing the Random Tool Table

This block clears all information in the random tool table:

```
G10.1 L20 P0 Q0 O0 R0;
```

Format for Programming Random Tool Table

Use this block to set data for the random tool pocket assignment table:

```
G10.1 L20 P__ Q__ O__ R__;
```

Where :	Is :
`G10.1 L20`	This tells the control that the block will be setting data for the random tool pocket table. The G10.1 L20 is not modal, it must be programmed in every block that sets data for the random tool pocket assignment table.
`P__`	The value following the P-word determines the pocket number that is being set.
`Q__`	The value following the Q-word determines the tool number of the tool that is in the pocket determined with the P-word.
`O__`	The value following the O-word enters the number of pockets that are needed for the tool. Normally a value of one is entered here however, for custom tools that require more than one pocket, program the number of pockets that are required.
`R__`	The value following the R-word enters the pocket number of the shaft pocket for the tool. Normally a value of one is entered here. However, for custom tools that require more than one pocket, program the location relative to the other pockets for that tool that the tool changer goes to to access that tool.

For example, this block

```
G10.1L20P1Q20O1R1;
```

tells the control that tool number 20 is in pocket number 1;

```
G10.1L20P3Q23O4R2;
```

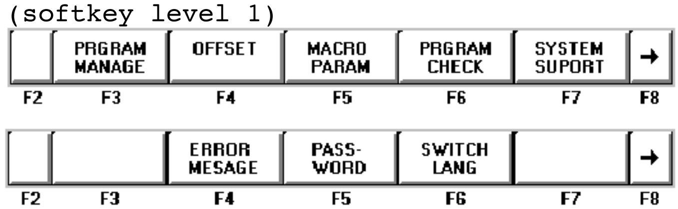
tells the control that tool number 23 has its shaft pocket as pocket number 3, four pockets are required for the custom tool and the second of these four pockets is the shaft pocket. This means that pockets 2, 3, 4, and 5 are used for the custom tool number 23.

Backup Random Tool Table

The control has a feature that allows you to back up (save) the information in the random tool table. The control generates a G10.1 program from the information already in the table. To do this follow these steps:

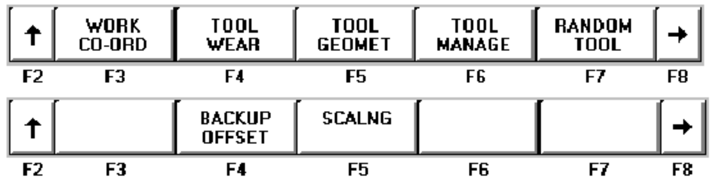
1. Press the **{OFFSET}** softkey.

(softkey level 1)



2. Press the **{RANDOM TOOL}** softkey.

```
(softkey level 2)
```



3. Press the **{BACKUP}** softkey. The control prompts you for a program name. Key in the program name that is to contain the information from the random tool table and press the **[ENTER]** key. This program name cannot already exist in control memory.

This allows the control to generate a program that automatically loads the necessary data into the random tool table. This program can be edited as changes to tool table are needed.

The control automatically places this G10.1 program in the control's memory.

Starting a Program with a Tool Already Active

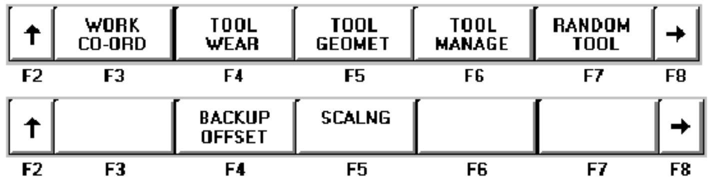
You can begin a part program with a tool already active in the chuck. In order for random tool to be able to properly handle that tool, it must enter information about that tool in the random tool table.

Important: If you use random tool when the tool was loaded into the chuck, it do not need to enter any data since random tool remembers what tool is loaded even after power is turned off. This procedure is only necessary if a tool is loaded manually or if random tool was not used when the tool was loaded.

The control needs the following information to properly handle a tool that is already active in the chuck. Tool number, number of pockets the tool uses, and position of the shaft pocket relative to these other pockets (refer to the section on manual entry of data for details on shaft pocket and custom tool data). Do this in the following way:

1. Press the **{ACTIVE}** softkey. The control prompts you for the tool number, the number of pockets, and the position of the shaft pocket relative to the these other pockets all separated by commas.

(softkey level 2)



(softkey level 3)

2. The control displays the configuration of the tool that it thinks is currently loaded into the chuck. If these values are incorrect, replace them using the correct tool information. Enter a value for tool number, number of pockets, and position of the shaft pocket all separated by commas on the input line. Data can be edited on the input line as described in chapter 2.

3. When the data for the tool that is currently in the chuck is correct, press the **[ENTER]** key. The control is now able to return the tool that is in the chuck to the best location in the tool changer at the proper time.

Important: You can also use the **{ACTIVE}** softkey to clear the currently active tool and specify no tool is currently in the spindle. To specify no tool is in the spindle press the **{ACTIVE}** softkey and delete any information that appears at the prompt. When the **[ENTER]** key is pressed, the active tool is cleared and the random tool assumes no tool is in the spindle.

Programming Alterations of the Offset Tables (G10L10 - G10L13)

It is possible to alter or generate values in the tool offset tables (refer to chapter 3) by using the programming feature discussed in the following section.

It is possible to enter data in the offset tables by programming the correct G10 command. The following section describes the use of the G10 commands.

Important: Note that G10 blocks may not be programmed when cutter compensation is active.

A G10 code used to modify a tool offset table value will only change the value in the table. If that offset value is currently being used by the control, the active offset value will not be updated until the offset value is called again from the table using a D or an H-word.

Any values entered in an offset table using the G10 command while the control is in incremental mode (G91), are added to the currently existing offset data. Any values entered in an offset table using the G10 command while the control is in absolute mode (G90), are used to replace the currently existing offset data.

The following is a representation of the basic format for modifying the offset tables.

```
G10 L(10-13) P__ R__;
```

Where :	Is :
L(10-13)	Designates the definitions of the other parameter data being used in the G10 block. The L-word defines the other parameters as described in Table 19.A.
P	refer to Table 19.A
R*	refer to Table 19.A

* When setting tool length axis values, R can be replaced with the name of the axis (typically X, Y, or Z). Axis names are always used by the control when offset tables are backed up by the control. If you use an R to set a tool length value, it is applied to the axis currently selected as the tool length axis. Refer to page 19-18 for details on selecting the active tool length axis.

Table 19.A
Parameters for Modifying the Tool Offset Tables

Value for the L Parameter	Parameter Definition P	R, X, Y, Z
L10 Geometry table	Offset Number	tool length geometry value
L11 Wear table	Offset Number	tool length Wear table
L12 Geometry table	Offset Number	Tool radius geometry value
L13 Wear table	Offset Number	Tool radius wear value

Example 19.4
Replacing the Tool Offset Tables Through Programming (G90)

Assume a Z axis geometry value (tool length) of 2 for offset number 4.

```
N00001 G90;

N00002 G10 L10 P4 Z3;     Offset number 4 has a new value of 3 for
                          tool length.

N00003 G10 L10 P4 Z1;     Offset number 4 has a new value of 1 for
                          tool length.
```

Example 19.5
Modifying the Tool Offset Tables Through Programming (G91)

Assume a Z axis geometry value (tool length) of 2 for offset number 5.

N00001 G91;	
N00002 G10 L11 P5 Z1;	Offset number 5 has a new value of 3 for tool length wear.
N00003 G10 L11 P5 Z3;	Offset number 5 has a new value of 6 for tool length wear.

Automatic Tool Life Management

Use the automatic tool management feature to monitor the life of a tool, determine when the tool should be replaced, and provide a replacement tool when that tool is requested in a program.

Tools are assigned to selected groups. Instead of calling a specific tool in a program, the programmer calls a tool group. The control then selects the first tool assigned to that group. If that tool has exceeded its entered tool life, then a replacement tool is selected from the next tool number assigned to that group. If that tool has exceeded its expected tool life, then the next tool in the group is selected. This continues until no more tools are available in that tool group. When a group is called that no longer has any available tools, an error is generated.

The correct tool length and tool radius offsets are assigned independently for each tool in the group.

Tool Directory Data

This section describes how to set up the tool groups and the information that must be entered for each tool group. This section described the manual method of entering this information.
Page 19-29 describes a method of entering all information into the tables by programming.

Assigning Tool Numbers to Groups

Normally tools that are assigned to the same group have similar characteristics (such as a boring tool or a drilling tool). If one tool in the group is worn, the control should be allowed to select any tool in the same group and still be able to cut the same part using the same program.

Your system installer determines in AMP the usable range of tool group numbers by determining a boundary. Any tool number that is programmed above this boundary is used as a tool group number (the value of the boundary is subtracted from the tool number programmed). Any tool number that is programmed below this boundary is used as a normal tool number. A maximum of 200 group numbers are available.

Enter different tool length offset numbers, and radius offset numbers into the tool management table with the tool numbers in each group. When you select a tool from a group by the control, the tool length and radius offset numbers are activated with them getting the data for the tools radius, length's for each axis, and orientation from the tool offset tables. See chapter 3 on entering tool data for details.

Tool Life Measurement Type

The control can measure the life of a tool using one of three possible methods:

Tool Life Type	Method Selected	Meaning
0	time	This is selected by choosing 0 as the type of tool life measurement. Time measures tool life as the length of time that a cutting tool is operated at a cutting feedrate. The value for the expected tool life is entered in units of minutes.
1	number of times used	This is selected by choosing 1 as the type of tool life measurement. Number of times used measures tool life as the number of times that the tool is selected as the active tool. The value for the expected tool life is entered as the number of times the tool may be used to cut parts; this number is per program. Regardless of the number of times that a tool is selected as active in a specific program, it only counts as one use each time the program is executed.
2	distance	This is selected by choosing 2 as the type of tool life measurement. Distance measures tool life as the distance that the tool has been moved using a cutting feedrate. The value for the expected tool life is entered in units of inches or millimeters depending on the mode that the control is operating in at the time. For multi-axis moves, the vectorial distance traveled by the tool is the distance used for tool life measurement.

Select the tool life type (selected as either 0, 1, or 2) on a per-group basis. Different groups may use different tool life types, however, each tool in the group uses the same tool life type.

Tool Life Threshold Percentage

A threshold level may also be assigned to a tool group. The threshold level is assigned as a percentage of the total expected life of the tool. When a tool reaches this threshold level, it is classified as old for that tool group. A tool is classified as old only to allow the operator to see that a tool is close to expiration. If the tool is being used when it reaches the threshold level, it continues to be used as normal until the tool reaches the “expired state” (100% of the expected tool life).

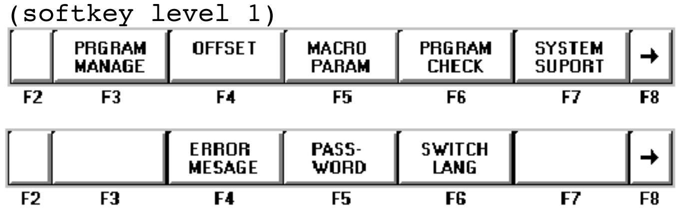
The tool life threshold percentage is selected on a per-group basis. Different groups can use different threshold percentage, however each tool in the group uses the same threshold percentage.

Entering Tool Group Data

To enter tool group data, you must create the tool groups. This is done automatically when the group is selected to edit. To enter tools into groups and enter other tool group data follow these steps:

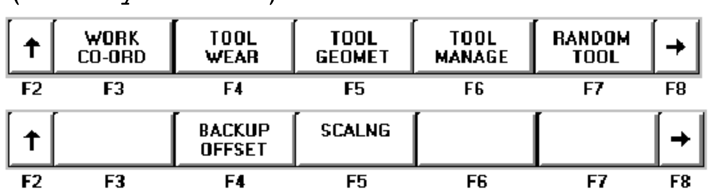
1. Press the **{OFFSET}** softkey.

(softkey level 1)



2. Press the **{TOOL MANAGE}** softkey.

(softkey level 2)



3. Press the **{TOOL DIR}** softkey. The control displays the current tool directory screen showing all of the current tools and the groups that they have been assigned to (see the following figure). The control displays the prompt “EDIT GROUP:”.

(softkey level 3)

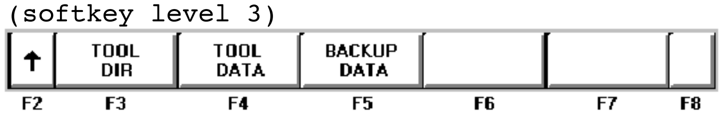
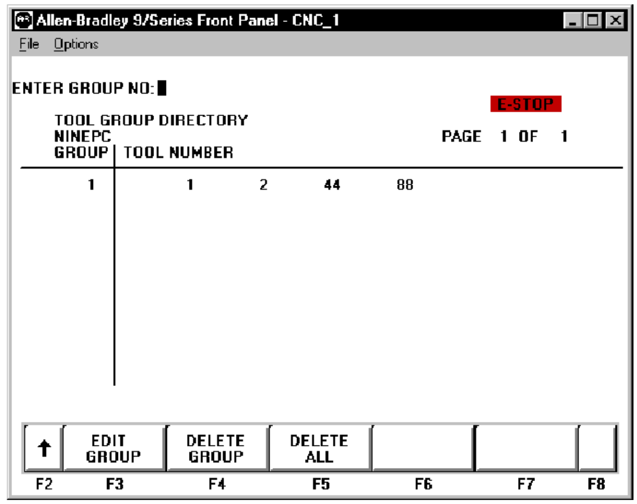


Figure 19.5
Typical Tool Group Directory Screen

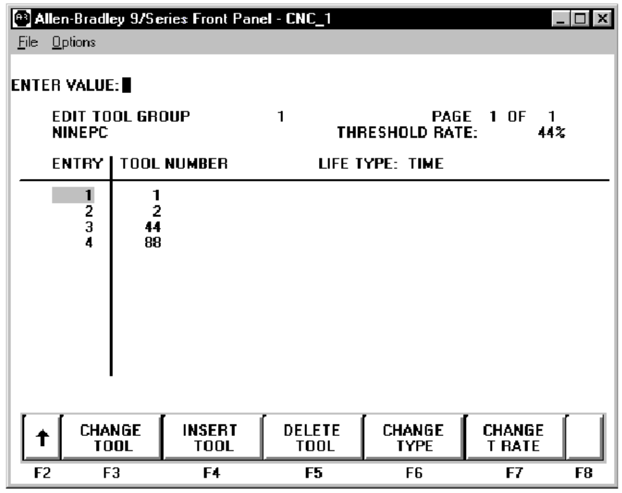


At this point, you can delete any or all tool groups that already exist for some reason follow these steps:

To delete:	Press:
select tool group	the **{DELETE GROUP}** softkey. Key in the desired group number to delete and press the **[ENTER]** key. This deletes all information in the tool group including the tool offset numbers, threshold rate, tool numbers, etc.
all of the tool groups	the **{DELETE ALL}** softkey. The control prompts "DELETE ALL TOOL MANAGEMENT DATA? (Y/N):". Entering "Y" deletes all tool management data that has been entered into the management tables (this does not delete any G10 programs that are backups or used to set the tool management tables). Entering "N" aborts the delete operation.

4. Key in the group number that is to be edited. When you select the correct group, press the **[EDIT GROUP]** key. Figure 19.6 shows all of the information for that tool group that is displayed.

Figure 19.6
Typical Tool Group Data Screen



5. From this screen, you can:

Operation:	Description:
Change tools	Alter one of the tool numbers that has already been entered in the group. Move the cursor to the tool number to be changed by pressing the up or down cursor keys (move the cursor full pages by holding down the shift key while pressing a cursor key). Press the **{CHANGE TOOL}** softkey. Key in the new tool number and press the **[ENTER]** key.
Insert tools	Insert a new tool number for that group. Move the cursor to the location to insert a new tool number at by pressing the up or down cursor keys (move the cursor full pages by holding down the shift key while pressing a cursor key). Press the **{INSERT TOOL}** softkey. Key in the new tool number and press the **[ENTER]** key. The actual range of allowable tool numbers is 1 to 9999.
Delete tool	Remove a tool number from that group. Move the cursor to the tool number to be removed by pressing the up or down cursor keys (move the cursor full pages by holding down the shift key while pressing a cursor key). Press the **{DELETE TOOL}** softkey. Respond yes or no and press **[ENTER]**.
Change life type	Alter how the control records and measures tool life for that group. Press the **{CHANGE TYPE}** softkey. The prompt "LIFE TYPE..." is displayed on line 2 of the CRT. The same life type is assigned to all tools in any one tool group. Key in the **number** of the desired tool life measurement type for that group and press the **[ENTER]** key. 0 for "time," 1 for "number of tool uses," and 2 for "distance."
Change life threshold rate	Alter the percentage of a tool's expected life so that a tool is labeled as old. This percentage applies to all tools in the selected group. To alter the threshold percentage (the percentage of total tool life that has been used before a tool will be classified as old) press the **{CHANGE T RATE}** softkey. Key in the percentage of the total tool's life so that the tool is classified as old and press the **[ENTER]** key. Tool life threshold rate is explained earlier in this section.

The application of these operations was described in detail earlier in this section. All of this information can be entered into the tool groups using the programming method described on page 19-26.

Assigning Detailed Tool Data

This section assumes that tools have already been assigned to their specific groups. This section describes specific information that is to be entered into the tool life management tables for the individual tools. This information may also be entered into the tool management tables using the programming method described on page 19-26. This information includes:

- Tool length offset number
- Tool diameter/radius offset number
- Expected life of a tool
- Renew a tool life

Tool Length and Diameter/Radius Offset Number

Use this feature of tool life management so the programmer does not need to know what tool has been called by tool life management and still have the correct tool offsets and cutter compensation activated.

Important: The control only automatically enters the tool length and cutter compensation offset numbers. This may or may not activate the tool length offset or cutter compensation features. These features must still be activated as normal.

Expected Tool Life

Use this feature of tool life management to set the expected life of a tool. The type of tool measurement used is assigned to the tool group as described on page 19-18. This tool measurement type determines the units that are used for the expected tool life.

As a tool is used the amount of usage is recorded and displayed as the accumulated tool life (the amount of the expected tool life that has been used). This is displayed individually for each tool on the tool data display screen. The accumulated tool life can be reset to zero by pressing the **{RENEW TOOL}** softkey.

The following is a description of the units that should be entered for the different tool life measurement types:

- If tool life is measured in units of time (0 is selected as tool life type), then the units for the expected tool life is minutes. Enter the minutes of operation that the tool is expected to operate and still be within the tolerance required for the part being cut. The accumulated life of a tool is only measured when that tool is the active tool, and it is performing a cutting operation. Moves that are rapid, or blocks that do not produce axis motion are not added to the accumulated tool life.

- If tool life is measured by the number of uses (1 is selected as tool life type), then the units for the expected tool is the number of programs that the tool may be selected as an active tool in. The accumulated life of a tool is increased by one if that tool is selected in a program as the active tool. Remember that the same tool may be active more than once in a program, however its accumulated life only increments by one. Enter the total number of program executions that can use the tool before the tool no longer meets the required tolerance for the part being cut.
- If tool life is measured in units of distance (2 is selected as tool life type), then the units for the expected tool life is either inches or millimeters (depending on the current operating mode of the tool). Enter the distance of travel that the tool is expected to cut and still be within the tolerance required for the part being cut. The accumulated life of a tool is only measured when that tool is the active tool, and it is performing a cutting operation. Moves that are are rapid, or blocks that do not produce axis motion are not added to the accumulated tool life. For multi-axis moves (including arcs and helices) the distance added to the accumulated life is the vectorial distance, not necessarily the distance traveled on each axes.

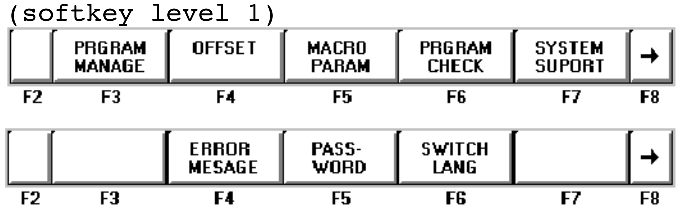
Entering Specific Tool Data

The following steps describe the method of entering specific tool data for tool management. This includes tool offset numbers, and expected tool life:

Important: This section assumes that the steps required to assign tools to specific groups has been performed as described in *Assigning Tool Numbers to Groups*.

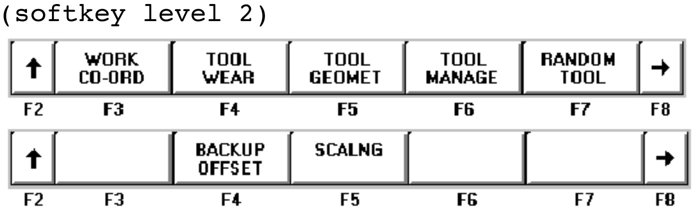
1. Press the **{OFFSET}** softkey.

(softkey level 1)



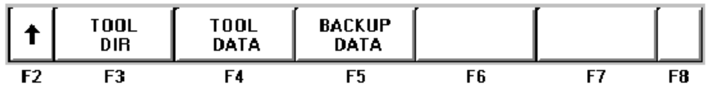
2. Press the **{TOOL MANAGE}** softkey.

(softkey level 2)



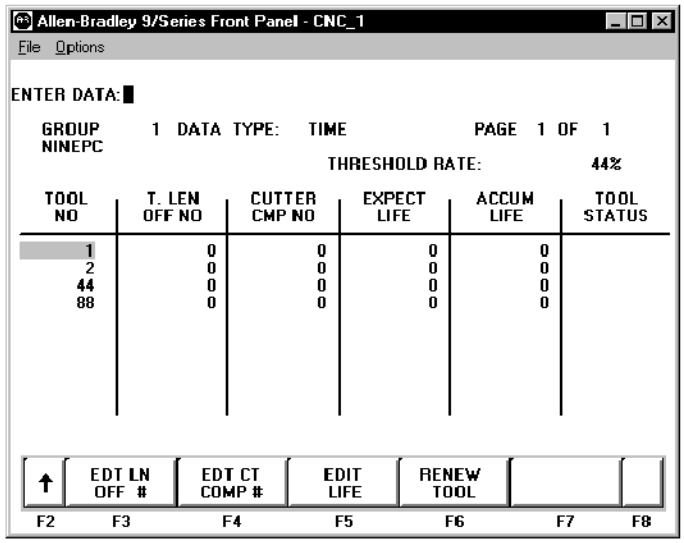
3. Press the **{TOOL DATA}** softkey. The control displays the prompt "EDIT GROUP:".

(softkey level 3)



4. Key in the group number to edit using your keyboard and press the **[ENTER]** key. Figure 19.7 shows all of the information for that tool group that is displayed.

Figure 19.7
Typical Tool Data Screen



5. From this screen it is possible to perform the following operations. The application of these operations was described in detail earlier in this section.

Operation:	Description:
Enter or alter the tool length offset number	To enter or alter a value for the tool length offset number, move the cursor to the tool number of the tool to alter and press the **{EDIT LN OFF}** softkey. Key in the new offset number that calls the correct data from the offset tables for that tool for its tool length offset number and press the **[ENTER]** key. The old value for tool length (if any) is discarded and the new value replaces it.
Enter or alter the tool radius offset number	To enter or alter a value for the tool radius offset number, move the cursor to the tool number of the tool to alter and press the **{EDIT CT CMP}** softkey. Key in the new offset number that calls the correct data from the offset tables for that tool for it's radius and press the **[ENTER]** key. The old value for radius offset numbers (if any) is discarded and the new value replaces it.

Enter or alter the expected life of a tool	To enter or alter a value for the expected life of a tool, move the cursor to the tool number of the tool to alter and press the **{EDIT LIFE}** softkey. Key in the new expected life of the tool (in units as determined by the tool life type) and press the **[ENTER]** key. The old value for expected life (if any) is discarded and the new value replaces it.
Reset the accumulated tool life to zero	To reset the accumulated tool life to zero, move the cursor to the tool number of the tool to alter and press the **{RENEW TOOL}** softkey. The old accumulated life of the tool is discarded and a value of zero is entered as the new accumulated tool life. This is normally performed after an old or expired tool has been replaced with a new tool. This updates the status of the tool and remove any “OLD,” or “EXPIRED” status.

Programming Data and Backing Up Tool Management Tables (G10L3, G11)

This feature allows the rapid loading of information into the tool management tables. This is done by executing a program that automatically loads the tool management tables. This program can also be generated automatically when the tool management tables are backed up as described later in this section.

Data is sent to the tool management tables when the control executes this G10 block:

```
G10L3;
```

This block indicates to the control that any information following this block is to be used to set the tool management tables.

ATTENTION: Any time that a G10L3; block is executed the control automatically clears all information that is in the management tables for all tools and tool groups.

Any time after the G10L3 command, parameters may be programmed to enter what tool group is being entered, the type of tool life measurement that is being used, and the tool life threshold percentage. The format for this block is:

```
P__I__Q__;
```

Where :	Is :
P__	The value entered with the P-word is used to program what tool group number is being edited. The following blocks assign tools to that tool group.
I__	The value entered with the I-word is used to program the type of tool life measurement that is to be used for all the tools in that group. I0 sets a type of time, I1 sets a type of number of uses, and I2 sets a type of distance. Refer to page 19-18 for details. If more than one I-word is programmed for a tool group the control uses the last programmed I-word for that group. If no I-word is programmed for a group the control uses I1 as a default value.
Q__	The value entered with the Q-word is used to program the threshold percentage for that tool group. Enter the percentage of the total expected tool life that causes the tools in the group to be classified as old. Refer to page 19-18 for details on threshold percentage. If the Q-word is not programmed in a block the control uses a default value of 80%. If more than one Q-word is programmed for a tool group the control uses the last programmed Q-word for that group. If no Q-word is programmed for a tool group the control uses Q80 as the default value.

The following program blocks assign tools to groups, length and cutter compensation offset numbers, and expected tool life to specific tools. This information is assigned to the last group number programmed in a block using the P-word. The format for these blocks is:

```
T__ H__ D__ L__;
```

Where :	Is :
T__	The value entered with the T-word is the tool number of the tool to be assigned to that group.
H__	The value entered with the H-word is the tool length offset number from the tool geometry and wear tables that is to be assigned to this tool. The H-word is only valid if programmed in the same block as a T-word.
D__	The value entered with the D-word is the tool radius number from the tool geometry and wear tables that is to be assigned to this tool. The D-word is only valid if programmed in the same block as a D-word.
L__	The value entered with the L-word is used to program the value of the expected tool life for that tool. The controls interpretation of this is dependant on the value set with the I-word in this program block. The value programmed with the L-word remains active for all following tools in that group until replaced with a different L-word, or a new tool group is programmed with a P-word.

All of the tools should then be programmed for that group in individual blocks. When all of the tools for that group have been entered, change groups by programming a different P-word in a block.

When all of the tools for all of the different groups have been entered, end the execution of editing the tool life management table by programming either a M02 or M30 end of program blocks or by entering this block:

```
G11;
```

This cancels the G10 data setting mode for tool management.

Important: Any information that was contained for a specific tool group that was written to using a G10L3 command (as previously described) is overwritten by the information programmed with the G10 blocks. All previous data for tool management for any of the groups is lost.

Example 19.6
Programming Tool Life Management Data

Program Block	Description
`G10L3;`	Starts loading tables.
`P1I1Q60;`	Begins loading data for tool group 1. Type 1 (number of uses) measurement. Threshold 60%.
`T1H5D7L25;`	Places tool 1 in group 1 with length offset number of 5, cutter radius offset number 7, and expected life of 25 uses.
`T2H2;`	Places tool 2 in group 1 with length offset number of 2, no cutter radius offset number and expected life of 25 uses.
`T15H7;`	Places tool 15 in group 1 with length offset number of 7, no cutter radius offset number and expected life of 25 uses.
`P2;`	Begins loading data for tool group 2. Type 0 measurement (default). Threshold at 80% (default).
`T12H3D6L40;`	Places tool 12 in group 2 with length offset number of 3, cutter radius offset number of 6, and expected life of 40 minutes.
`T13;`	Places tool 13 in group 2 with length and radius offset numbers of 0 and expected life of 40 minutes.
`P4I0Q90;`	Begins loading data for tool group 4. Type 0 (time) measurement. Threshold at 90%.
`T20H3D6;`	Places tool 20 in group 4 with length offset number of 3, cutter radius offset number of 6, and expected life of 0 minutes.
`Q50;`	Resets the threshold at 50% for group 4.
`G11;`	Ends the loading operation.
`M02;`	

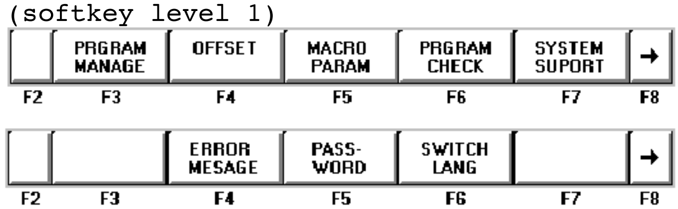
Backing Up Tool Management Tables

This feature causes the control to automatically generate a G10L3 program that stores all of the information that it finds in the current tool management table. Any time that this G10 program is executed, it clears any information that is currently in the management tables and replaces it with the information that is in the G10 program.

To generate the G10L3 backup program of the tool management tables, follow these steps:

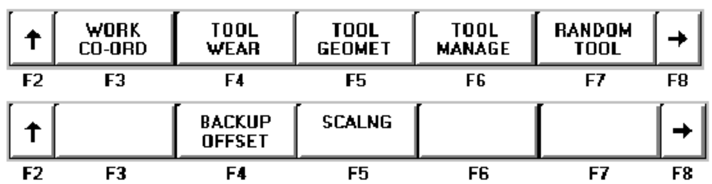
1. Press the **{OFFSET}** softkey.

(softkey level 1)



2. Press the **{TOOL MANAGE}** softkey.

(softkey level 2)



3. Press the **{BACKUP DATA}** softkey. The prompt “BACKUP FILENAME:” is displayed on the input line.

(softkey level 3)

4. Key in any legal program name and press the **[ENTER]** key. This program name is used as the program that stores all of the backed up tool management data. The control generates the tool management G10 program.

Programming a T-word Using Tool Management

This section describes how to activate a tool using tool life management. When using tool life management, remember:

- Your system installer sets up a boundary for T-words used with tool life management in AMP. Any T-word that is programmed less than or equal to this number will be used as a normal tool number. Any T-word that is programmed greater than this number is used as a tool group number for tool management.

- When a T-word is programmed using tool life management, the group that is called with the T-word is equal to the programmed T-word, minus the value of the boundary set in AMP by the system installer. For example, if your system installer sets the boundary as 100, programming T101 would call a tool from group 1. T102 would call a tool from group 2, programming a T-word of 100 or less would call the specific tool number that is programmed. This manual assumes that a boundary of 100 is set in AMP. Refer to the system installer's documentation for details on the boundary value for a specific system.
- Your system installer determines in AMP when the control activates a tool after a T-word is programmed. This manual assumes that an M06 block is required to activate the tool change process, and any T-word that is programmed in the M06 block must be the tool number or group number of the tool that is being replaced. This may not be the case in a specific system (refer to page 19-1 for the four different types). Refer to the system installer's documentation for details on activating a T-word.
- Any time after a tool from a tool group has been activated and a D- or an H-word is programmed, the newly programmed D or H value will take priority. The values in the tool management table for the tool length offset number and tool radius offset number are ignored until a different tool is selected or that same tool is reselected.

Example 19.7
Programming Tool Changes Using Tool Life Management

The following example assumes that the system installer has configured in AMP, both, the boundary for tool life management at 100, and an M06 to perform a tool change. It also is assumed that the tool changer is located at the secondary machine home point called by a G30, this is not necessarily true for different machine applications.

Program Block	Description
`G49G30X10Z10F100;`	Return to secondary home position.
`T101;`	Next tool change will be a tool from group 1.
`M06;`	Change to a group 1 tool.
`G43;`	Activate tool length offset using the offset number for the tool as assigned in the tool management table.
`G29;`	Return from secondary home position
`G42;`	Activate cutter compensation right using the offset number for the tool as assigned in the tool management table.
`T102;`	Next tool change will be a tool from group 2.
`G01X13Y1F200;`	Cutting with a group 1 tool.
`G30;`	Return to secondary machine home.
`M06T101;`	Replaces the group 1 tool with a group 2 tool. Note the T-word is optional in this block.
`G29;`	Return from secondary home position. New tool length offset values and new tool radius offset values take effect.
`G01X2Y2F100;`	Cutting with a group 2 tool.
`G41D2;`	Changes the current tool radius number that was activated with this tool and replaces it with the new D2 offset values. Note that the tool management table does not get changed. Also changes to cutter compensation left.
`M30;`	

END OF CHAPTER

Cutter Diameter Compensation (G40, G41, G42)

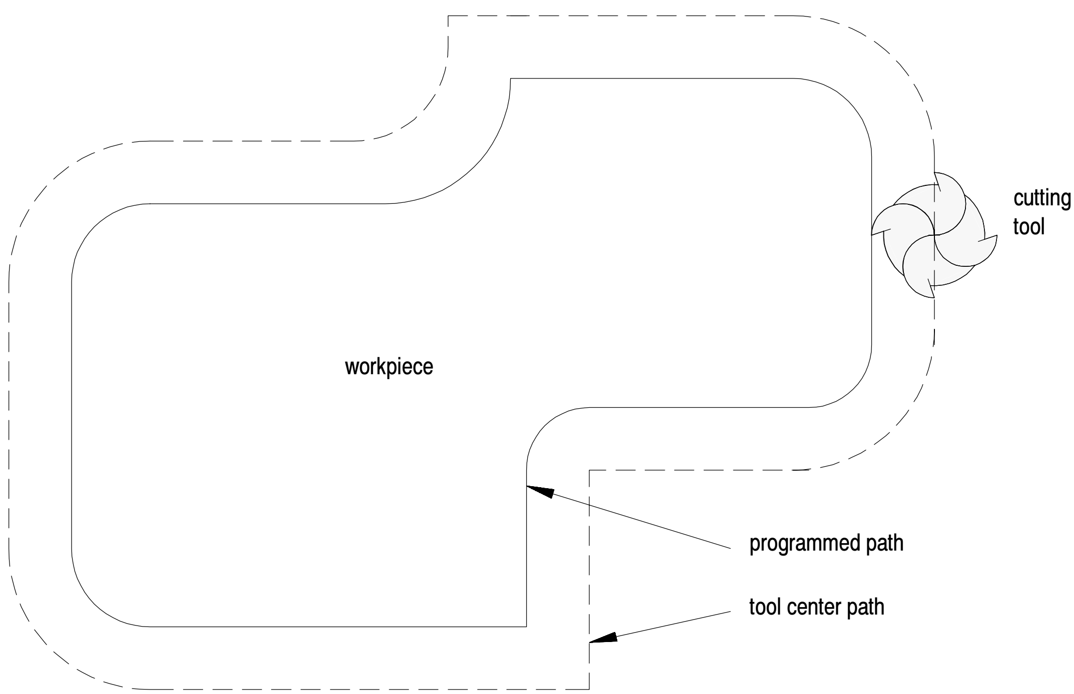
Chapter Overview

To cut a workpiece using the side face of the cutting tool, it is more convenient to write the part program so that the center of the tool moves along the shape of the workpiece.

Since all cutting tools have a diameter, a program written for moving the center of the tool will not cut the workpiece to the proper size. To produce a workpiece that has the correct size, offset the path of the tool center from the shape of the workpiece by an amount that equals the radius of the tool.

The control offers a function, called cutter compensation, for offsetting tool paths by the amount of a tool radius. Put the radius of the tool into the offset tables in advance (refer to chapter 3 or page 19-16). This function lets the control use the same program to produce the same workpiece regardless of the radius of the tool that does the cutting.

Figure 20.1
Tool Radius Diameter Compensation



We use these terms in this section:

- **inside** -- An angle between two intersecting programmed tool paths is referred to as inside if, in the direction of travel, the angle measured clockwise from the second tool path into the first is **less than or equal to 180 degrees**. See Figure 20.2. If one or both of the moves are circular, the angle is measured from a line tangent to the tool path at their point of intersection.
- **outside** -- An angle between two intersecting programmed tool paths is referred to as outside if, in the direction of travel, the angle measured clockwise from the second tool path into the first is **greater than 180 degrees**. See Figure 20.2. If one or both of the moves are circular, the angle is measured from a line tangent to the tool path at their point of intersection.
- **r** -- cutter radius
- **CR** -- cross-point between two programmed paths after the cutter compensation is activated

Two types of cutter compensation are available on the control:

- type A (as described on page 20-9)
- type B (as described on page 20-19)

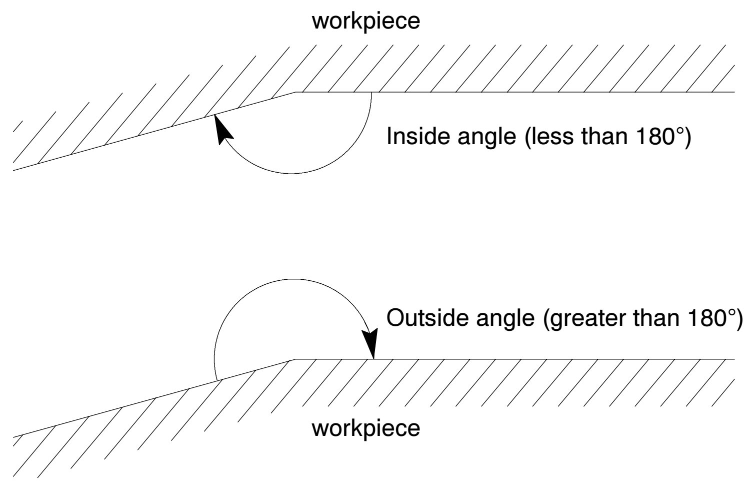
This table highlights the differences between the two types:

Type of Move	Type A	Type B
Entry Move Into Cutter Compensation	The tool takes the shortest possible path to its offset position.	• The tool stays at least one radius away from the start point of the next block at all times. • Extra motion blocks can be generated to attempt to prevent gouging of the part as may occur in Type A.
Tool Path	Same	
Exit Move From Cutter Compensation	The tool takes the shortest path to the end-point of the exit move for both inside and outside corners.	• The tool takes the shortest path to the end-point of exit move for **inside corners only.** • For **outside corners**, the tool stays at least one radius away from the end-point.

The system installer determines whether type A or type B is used by a control in AMP.

ATTENTION: If you use a 2-turret lathe, be aware that the X tool offset and the tool orientation values will be opposite of the A turret values for the second mirrored (B).

Figure 20.2
Definition of Inside and Outside



Active Cutter Compensation

Use these G-codes for cutter compensation:

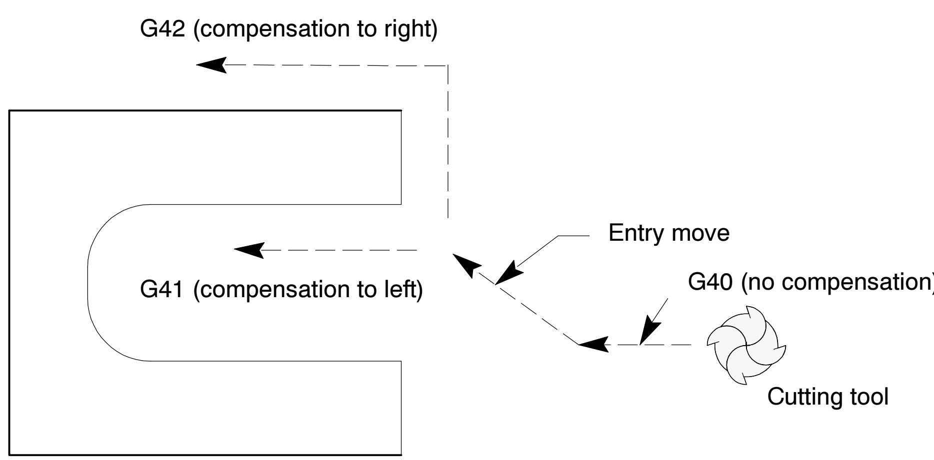
- G41 -- cutter compensation, left
- G42 -- cutter compensation, right
- G40 -- cutter compensation, cancel

Left or right is defined as offsetting the tool to the left or right of the programmed cutting path when facing the direction of cutter motion.

Important: If a negative value is set in the offset tables as the tool radius, compensation direction (tool left or right) is reversed for G41 and G42. G41 and G42 are also reversed during the mirroring operation (refer to chapter 13).

All of these G-codes are modal and belong to the same modal group.

Figure 20.3
Cutter Compensation Direction



Program the cutter compensation function with the following format:

```
G41(or G42)X ___ Y ___ Z ___ D ___ ;
```

Where :	Is :
G41(or G42)	cutter compensation direction, G41=left, G42=right
X, Y, Z	End-point of entry move into cutter compensation. Program an entry move on axes only in the currently active plane. Axis motion must take place in order for cutter compensation to be active on an axis.
D	Designates the offset numbers and pulls data: 1) from the wear and geometry tables for the tool radius, and 2) from the geometry table for tool orientation (refer to chapter 10 for information on D-words). The D-word is optional in the G41 or G42 blocks. The D-word may also be designated in any previous or following program block.

Cutter compensation can be programmed in various ways. Following are examples of 1-, 2-, and 3-block programs activating cutter compensation with entry moves.

Example 20.1
Initializing Cutter Compensation

Assume: G17 (XY Plane Selection)

Program Block	Comment
One Block	
G42 D1 X1 Y1;	Sets compensation right, selects tool radius offset number, and activates move to X1 Y1
Two Blocks	
D1;	Selects tool radius offset number
G42 X1 Y1;	Sets compensation right and activates move to X1 Y1
Three Blocks	
D1;	Selects tool radius offset number
G42;	Sets compensation right
X1 Y1;	Activates move to X1 Y1

Important: Any entry move (refer to page **21.3.1 or 21.4.1**) into cutter compensation must be a linear move. Initial activation of cutter compensation by programming of either the G41 or G42 commands in a circular cutting mode (G02 or G03) is not allowed. However, if cutter compensation is already active, the G41 or G42 commands may be programmed in a circular block to change cutter compensation direction either left (G41) or right (G42).

Example 20.2
Cutter Compensation Sample Paths

All of the following blocks result in the same tool path. Assume the selected plane is the XY plane.

```
N1D1X0Y0;
N2G41X1Y1;
N3X2;
M30;
```

or

```
N1X0Y0F500;
N2G41X1Y1D1;
N3X2;
M30;
```

or

```
N1X0Y0F500;
N2G41;
N3X1Y1D1;
N4X2
M30;
```

Important: The cutter compensation feature is not available for any motion blocks that are programmed in MDI mode (refer to page 20-46). The cutter compensation mode may be altered by programming either G41, G42, or G40; or the tool radius may be changed in an MDI program. However, none of the tool paths executed in MDI is compensated. Any changes made to cutter compensation are applied until the next block executed in automatic mode.

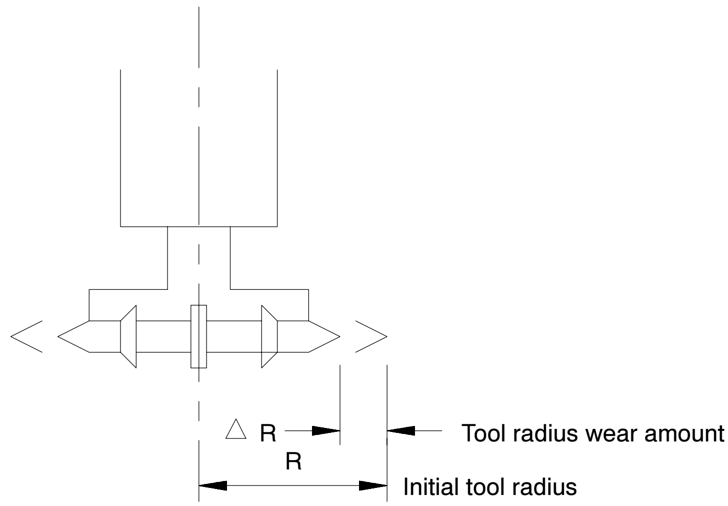
The D-word calls the following data from the offset tables:

- initial cutter radius data
- cutter radius wear offset data

The sum of these two types of offset data is used by the control as the data for the cutter compensation function.

Unless **Cutter Compensation** is active, when a program recover is performed, the control automatically returns the program to the beginning of the block that was interrupted. In the case of power failure, the control will even reselect the program that was active prior to the interruption.

Figure 20.4
Tool Radius Wear



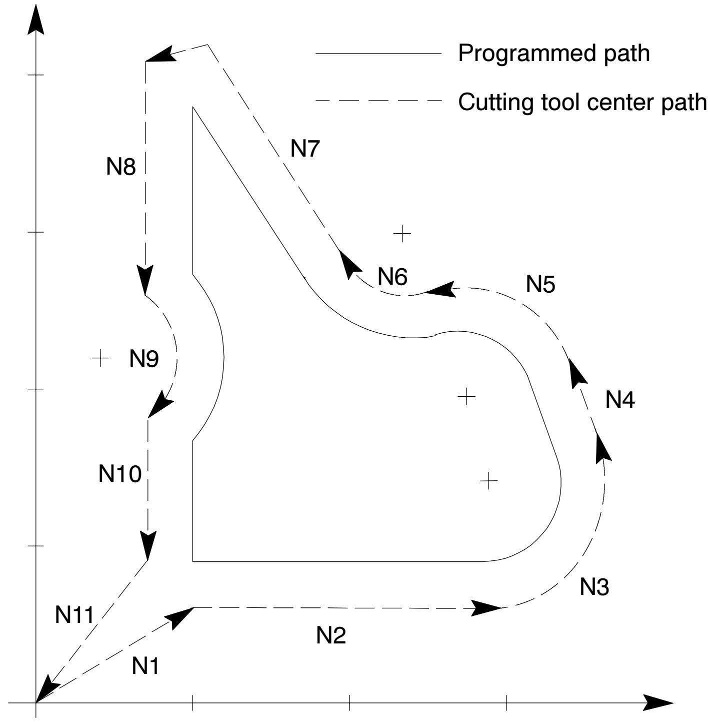
G40 (cutter compensation cancel) is active when power is turned on, E-Stop is reset, when the control is reset, or when an M02 or M30 end-of-program block is executed.

Example 20.3
Cutter Compensation Sample Path

Assume: D01 = 5mm

Program Block	Comment
`G00X0Y0;`	Establish current position as X0Y0
`N1G00G42X20.Y20.D01;`	Rapid to start and set compensated right
`N2G01X70.F1000.;`	Feed move 1
`N3G03X82.99Y42.5R15.;`	Move 2
`N4G01X72.99Y62.5;`	Move 3
`N5G03X59.33Y66.16R15;`	Move 4
`N6G02X38.521Y69.797R16;`	Move 5
`N7G01X20.Y95.;`	Move 6
`N8Y71.18;`	Move 7, creates a generated block
`N9G02Y48.82R15.;`	Move 8
`N10G01 Y20.;`	Move 9, cutter moves away from path at end
`N11G00G40X0Y0D00;`	Rapid to start point and cancel compensation
`N12M30;`	End of Program

Figure 20.5
Results of Cutter Compensation Program Example



Cutter Compensation-generated Blocks (G39, G39.1)

In certain instances, cutter compensation creates a non-programmed move called a generated block. These blocks are improve cycle time and corner-cutting quality.

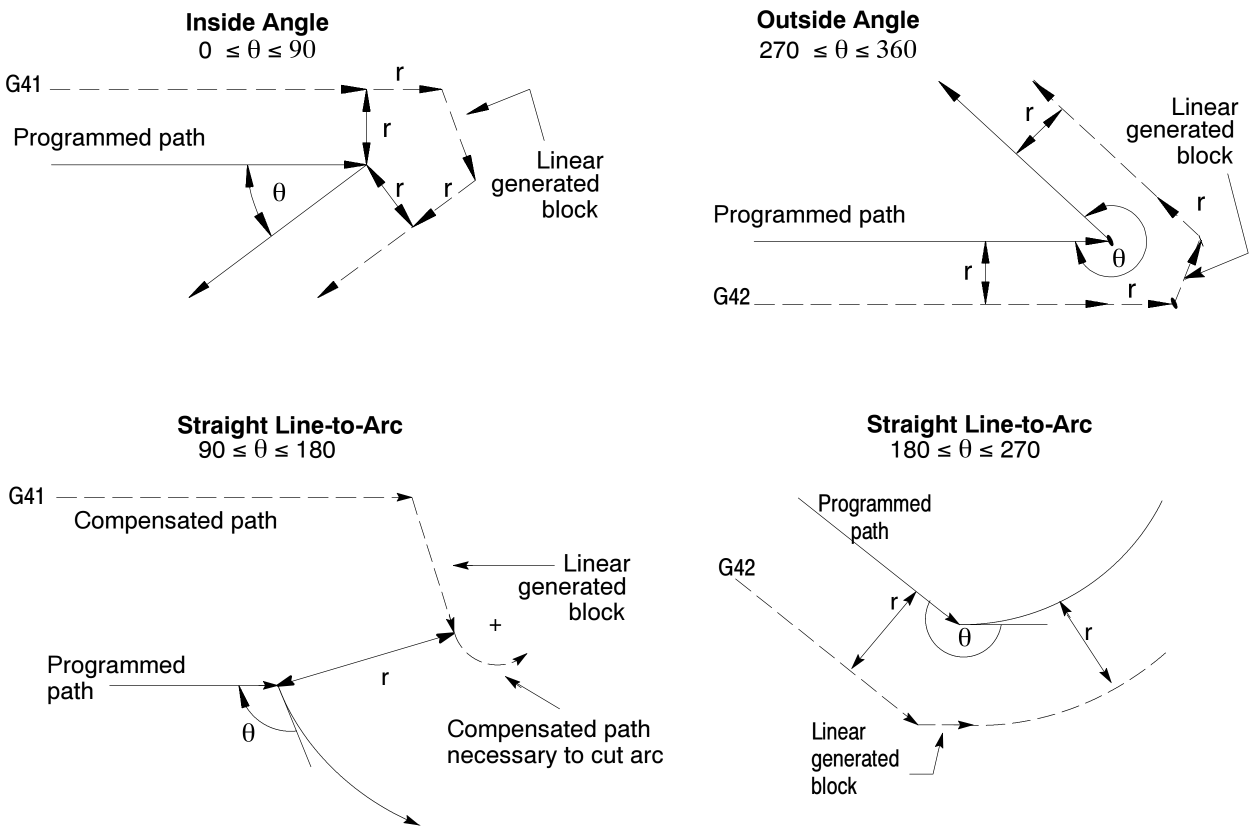
Cutter Compensation generates blocks for type A or B moves as follows:

Type of Move	Type A	Type B
Entry Move	No block is generated	Block is generated
Tool Path	Block is generated	Block is generated
Exit Move	No block is generated	Block is generated

Important: Cutter compensation generated blocks, as shown in Figure 20.6, are created only under these conditions:

When	is active and is cutting:	which is:
G41	an inside angle	less than 90 degrees
G42	an outside angle	greater than 270
G41	straight line to arc (or arc to straight line)	greater than 90 degrees but less than 180 degrees
G42	straight line to arc (or arc to straight line)	greater than 180 degrees but less than 270 degrees

Figure 20.6
Cutter Compensation Examples

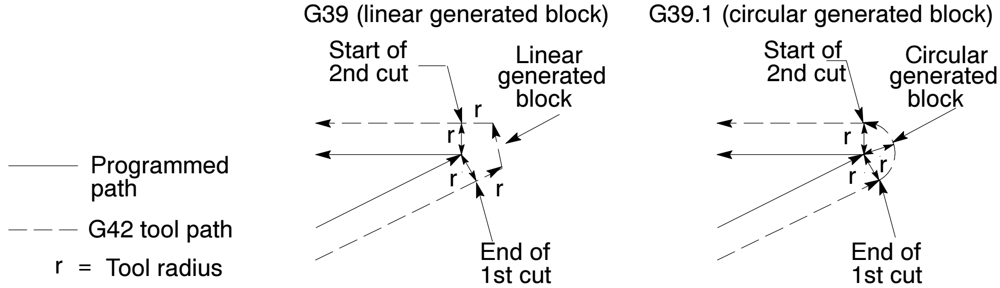


Besides choosing between types A and B (selected in AMP), cutter compensation generated blocks can also be controlled by programming a G39 or G39.1. These G-codes determine whether the generated block will be linear (G39) or circular (G39.1) as shown in Figure 20.7.

```
G39(or G39.1);
```

Where :	Causes:
G39	linear generated blocks. If neither G39 nor G39.1 is programmed, G39 is the default. This command is modal.
G39.1	circular generated blocks. When cutting straight line-to-arc or arc-to-straight line moves, the generated block will always be linear and the G39.1 will be ignored. This command is modal.

Figure 20.7
Cutter Compensation Generation Blocks (G39 vs G39.1)



G39 or G39.1 can be programmed in any block. However, they must be programmed in or before the block that causes a cutter compensation generated block.

Important: For linear generated blocks, the system installer can define a minimum block length in AMP. If the generated move length is less than the system-defined minimum block length, no generated block is created. The tool path proceeds to the intersection of the two compensated paths. If the generated move length is equal to or greater than the system-defined minimum block length, a generated block is created.

Throughout this chapter, we show drawings where a generated block is created. Both G39 and G39.1 are shown in these drawings where applicable.

Cutter Compensation (Type A)

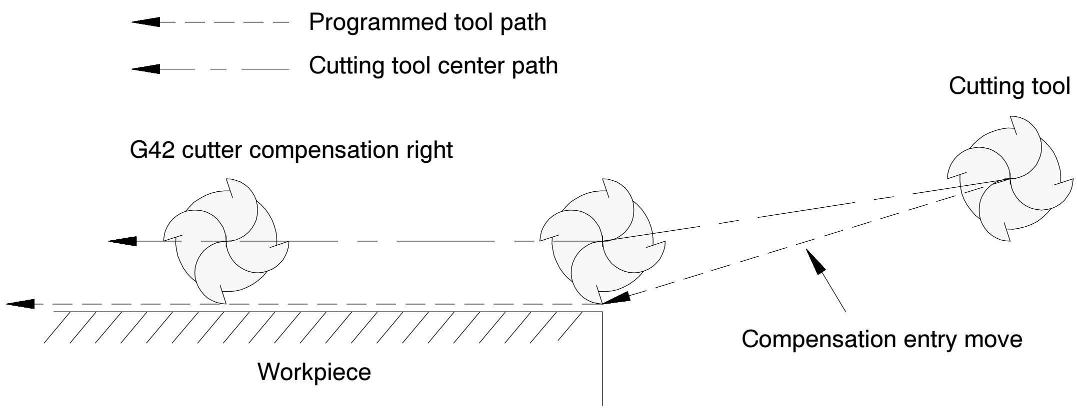
The easiest way to demonstrate the actual tool paths taken by the cutting tool when using cutter compensation type A is by pictorial representation. The following subsections give a brief description of the cutter path, along with a figure to clarify the description.

Cutter Compensation Type A Entry Moves

An entry move is defined as the path that the cutting tool takes when the cutter compensation function first becomes activated in a program. Figure 20.8 gives an example of a typical entry move.

Important: Any entry move into cutter compensation must be a linear move. Initial activation of cutter compensation by designation of either the G41 or G42 commands in a circular cutting mode (G02 or G03) is not allowed. The G41 or G42 commands may be designated in a circular block to change cutter compensation direction, as long as cutter compensation is already active.

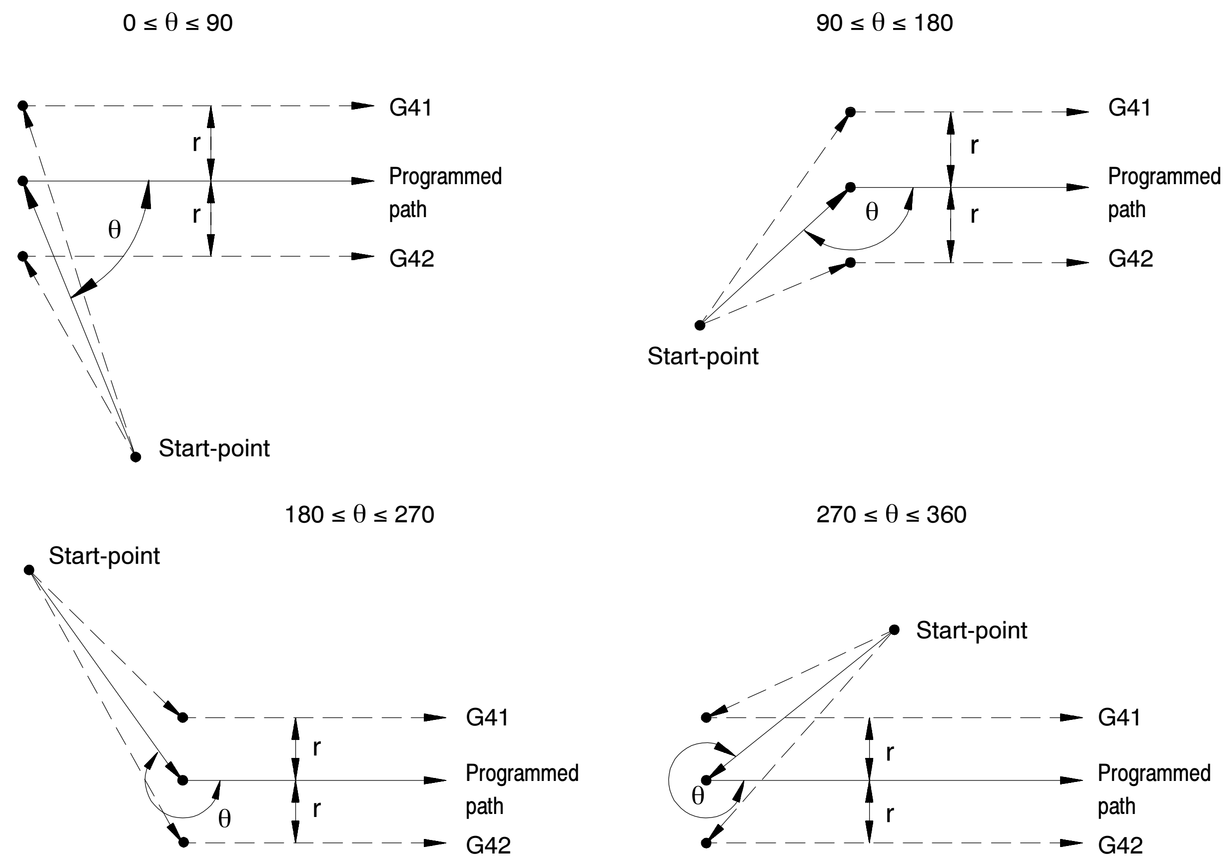
Figure 20.8
Cutter Compensation Entry Move



The entry move of the cutting tool for type A cutter compensation takes the shortest possible path to its offset position. This position is at right angles to and on the left or right side of the next programmed move in the currently defined plane.

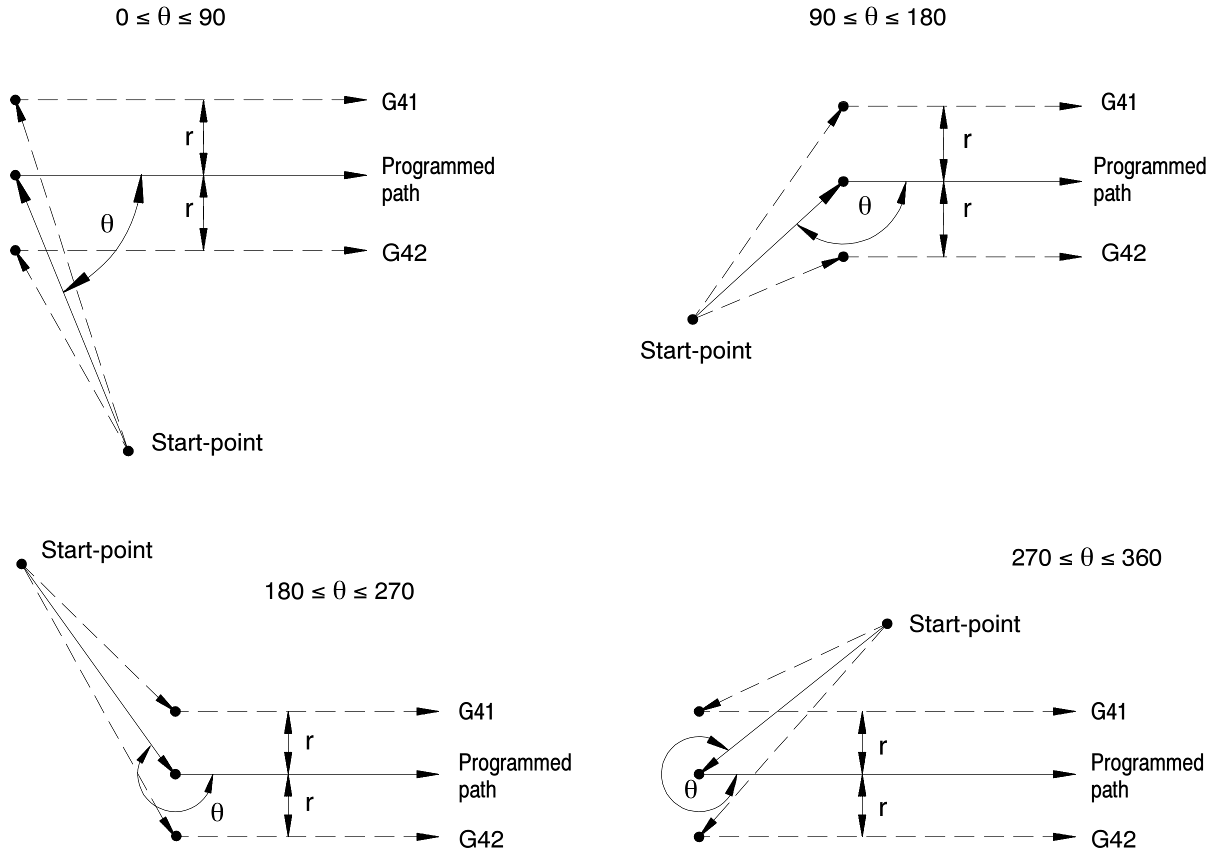
Figures 20.9 through 20.11 show examples of typical entry moves using type A cutter compensation:

Figure 20.9
Tool Path for Entry Move Straight Line-to-Straight Line



If the next programmed move is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the start-point of that circular move.

Figure 20.10
Tool Path for Entry Move Straight Line-to-Arc



There is no limit to the number of blocks that may follow the programming of G41 or G42 before an entry move takes place. The entry move is always the same regardless of the number of blocks that do not program motion in the current plane for compensation.

Example 20.4
Sample Entry Move After Nonmotion Blocks

Assume current compensation plane is the XY plane.

N1X0Y0F500;	
N2G41D1;	This block commands compensation left.
N3Z1;	This is not the entry block since no axis motion takes place in the current plane.
N4...;	No axis motion in current plane.
N5...;	No axis motion in current plane.
N6...;	No axis motion in current plane.
”	”
”	”
”	”
N999X1Y1;	This is the entry move for the previously programmed G41.
N1000M30;	

The system installer selects in AMP the maximum number of nonmotion blocks to be allowed during cutter compensation before the entry move must be reinitialized.

Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

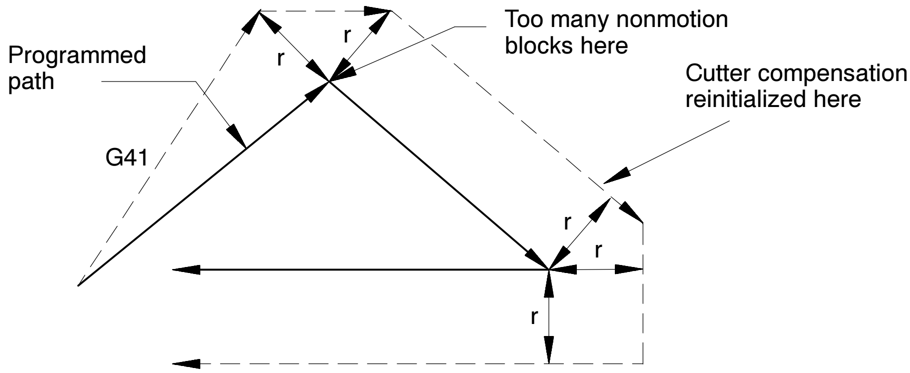
For example, assume that the system installer has designated that only 2 nonmotion blocks may be performed. If more than two blocks following the entry move do not contain axis motion in the current plane, then the entry move is re-performed at the next block containing axis motion in the current plane.

Example 20.5
Too Many Nonmotion Blocks After Entry Block

Assume current plane to be the XY plane and the system installer has designated that only 2 nonmotion blocks may be performed before cutter compensation is reinitialized.

N1X0Y0F500;	
N2G41D1X1Y1;	Entry move cutter compensation left.
N3Z2;	No axis motion in current plane.
N4...;	No axis motion in current plane.
N5X4Y-4;	New entry move cutter compensation left.
N6M30;	

Figure 20.11
Entry Move Followed by Too Many Nonmotion Blocks



Cutter Compensation Type A Exit Moves

The cutter compensation feature is cancelled by programming G40. The path that is taken when the tool leaves cutter compensation is refereed to as the exit move. The path that the tool follows during an exit move is dependant on:

- The direction of compensation (G41 or G42)
- The angle between the last motion made in cutter compensation (in the current compensation plane) and the motion of the of the exit move

Designating a tool offset number D00 in a program does not cancel cutter compensation and does not generate an exit move. Cutter compensation simply continues on as if a tool radius had been changed to a radius of zero (refer to page 20-43 on changing cutter radius). The exit move, if D00 is the active tool radius, is then equal to the programmed tool path.

Important: An exit move cannot be a circular move (G02 or G03). Any exit move must be programmed on a linear path. Any attempt to generate an exit move using a circular path generates a block-format error.

Example 20.6 gives some sample exit move program blocks.

Example 20.6
Type A Sample Exit Moves

Assume the current plane to be the XY plane and cutter compensation is already active before the execution of block N100 in the following program segments.

N100X1Y1;	
N110X3Y3G40;	Exit move.
N100X1Y1;	
N110G40;	Exit move.
N120X3Y3;	
N100X1Y1;	
N110G40;	
N120Z1;	No axis motion in the current plane.
N130...;	No axis motion in the current plane.
N140...;	No axis motion in the current plane.
”	”
”	”
N200X3Y3;	Exit move.
N100X1Y1;	
N110Z1;	No axis motion in the current plane.
N120...;	No axis motion in the current plane.
N130...;	No axis motion in the current plane.
”	”
”	”
N200G40X3Y3;	Exit move.

All of the program blocks in Example 20.6 produce the same exit move provided that the number of nonmotion blocks in the compensation mode has not exceeded a value selected by the system installer in AMP.

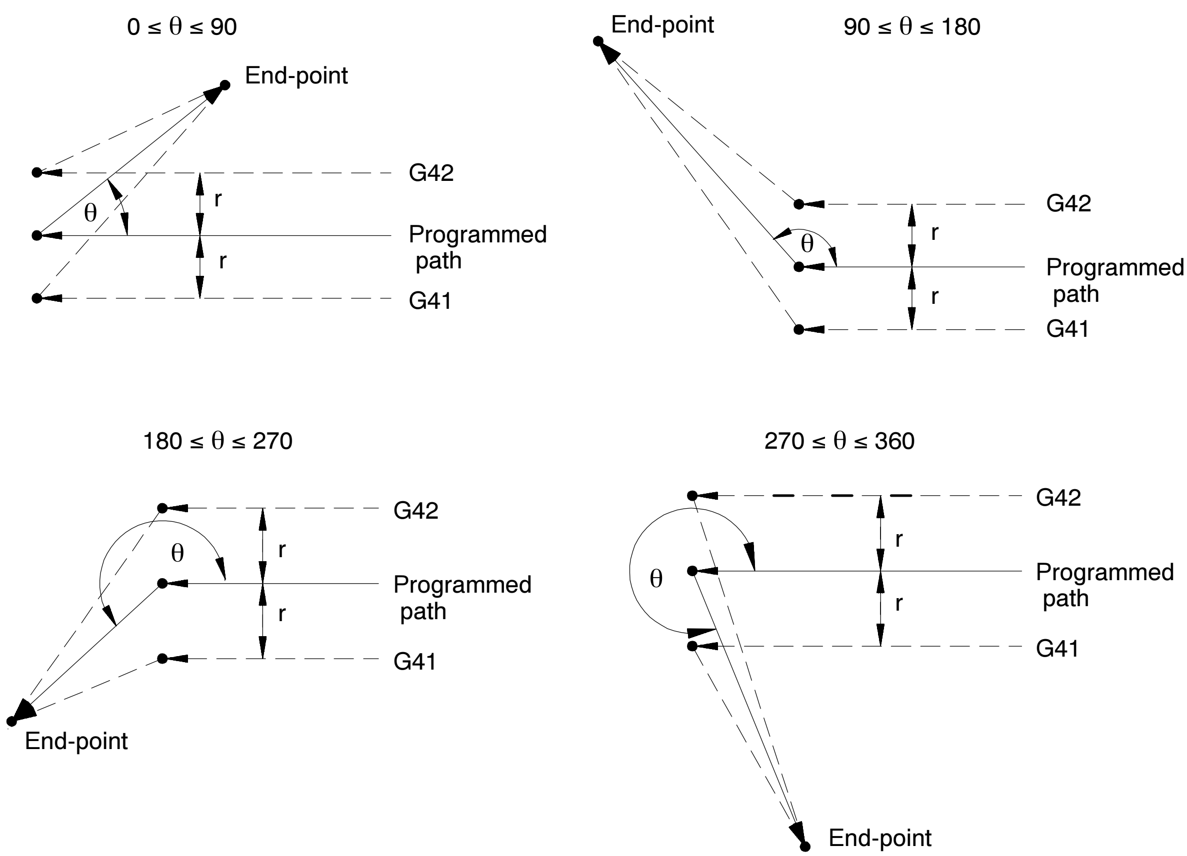
Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

The exit of the cutting tool for type A cutter compensation takes the shortest possible path to the end-point of the exit move. This path starts at right angles to the left or right of the end-point (depending on G41 or G42) of the last move in the currently defined plane (it is possible to redefine this start-point using an I, J, and/or K word as described later in this section). The end-point of the exit move is no longer offset to the left or right.

Figures 20.12 through 20.16 show examples of typical exit moves using type A cutter compensation. All examples assume that the number of nonmotion blocks before the designation of the G40 command have not exceeded the number allowed as determined by the system installer in AMP.

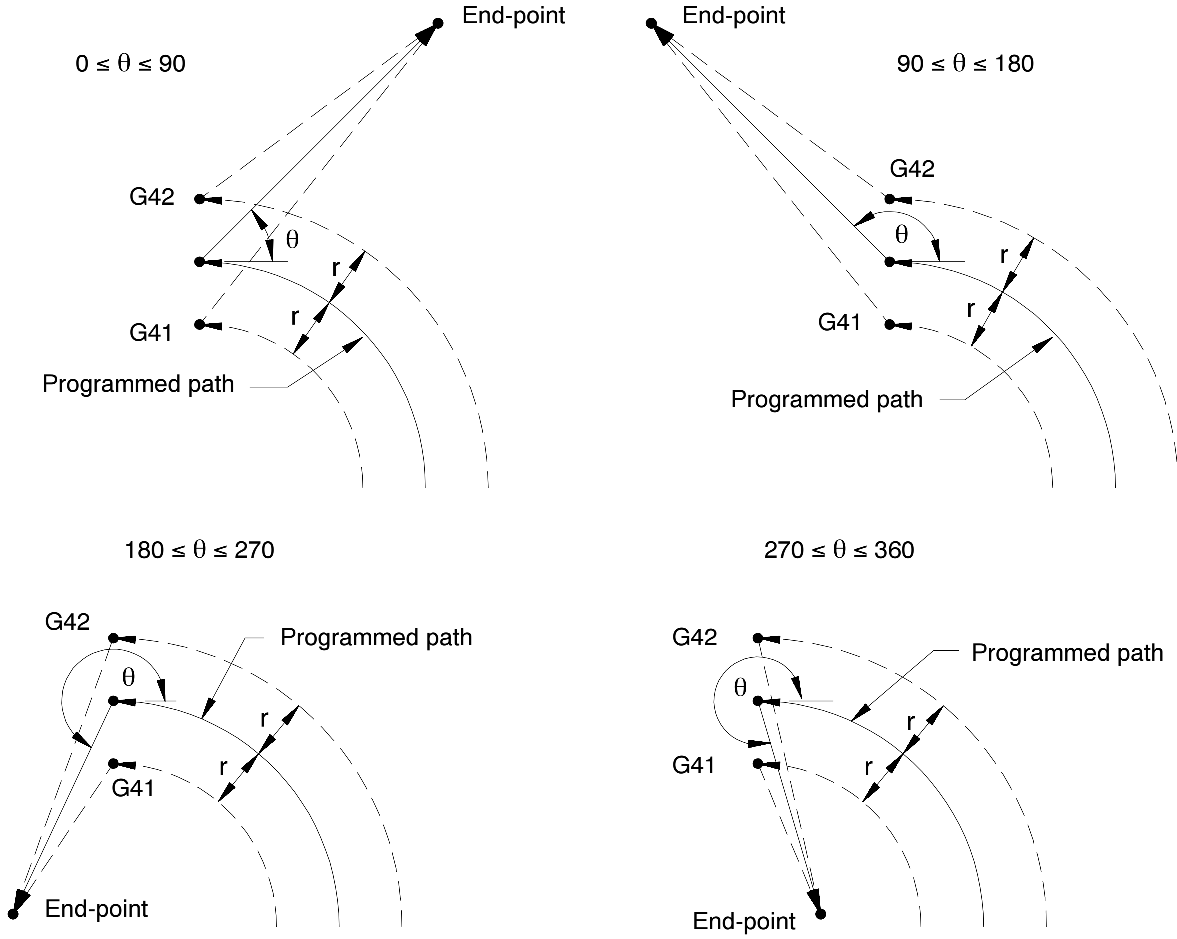
Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

Figure 20.12
Tool Path for Exit Move Straight Line-to-Straight Line



If the last programmed move prior to the exit move (which must be linear) is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the end-point of that circular move.

Figure 20.13
Tool Path for Exit Move Arc-to-Straight Line



The above examples in Figures 20.12 and 20.13 assume that the number of blocks that do not contain axis motion in the current plane, following the programming of G40 before an exit move takes place, does not exceed an amount selected in AMP by the system installer. If the number of nonmotion blocks following G40 exceeds the limit, the control generates its own exit move. This may often cause over-cutting of the part.

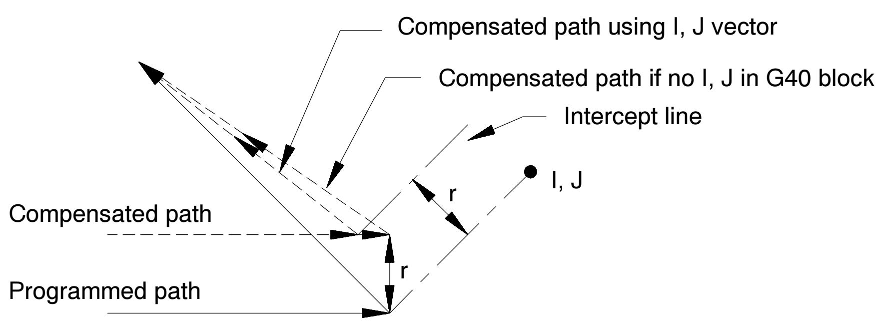
Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

It is possible to modify the path that the tool takes for an exit move by including an I, J, and/or K word in the exit move. Only the I, J, or K words that represent values in the current plane are programmed in the block containing the exit move. I, J, and K correspond to the X, Y, and Z axes respectively.

The I, J, and K words in the exit move block define a vector that is used by the control to redefine the end-point of the previous compensated move. I, J, and K words are always programmed as incremental values regardless of the current mode (G90 or G91).

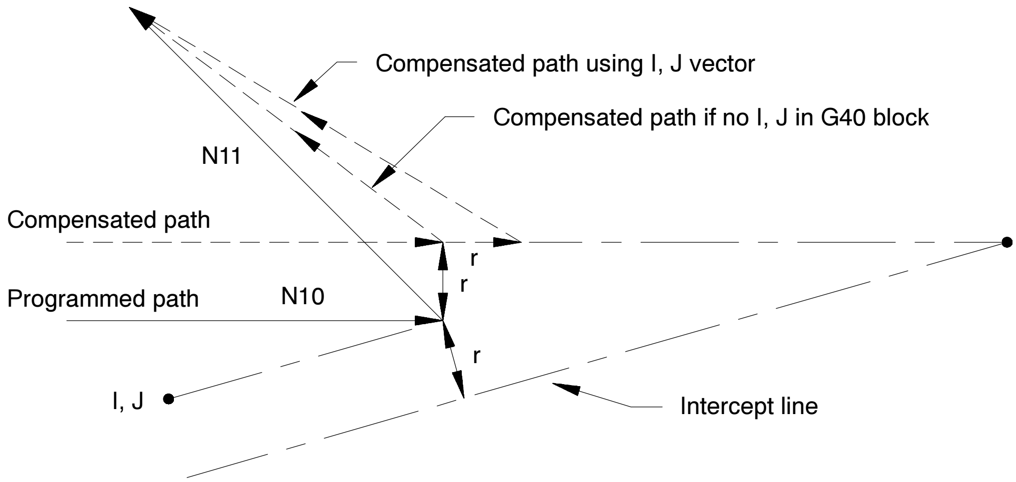
The vector defined by the I, J, and/or K words is along a line drawn from the end-point of the programmed path to a point referenced from the end-point of the programmed path a distance along the axes in the current plane an amount as designated with the I, J, and/or K words. A new vector is then defined parallel to the vector defined by the I, J, and/or K word and offset from this vector in the direction and amount of the currently active offset (G41 or G42). The intersection of this new vector with the current compensated tool path defines a point which is the new end-point of the last programmed compensation move.

Figure 20.14
Exit Move Defined By An I, J, K Vector



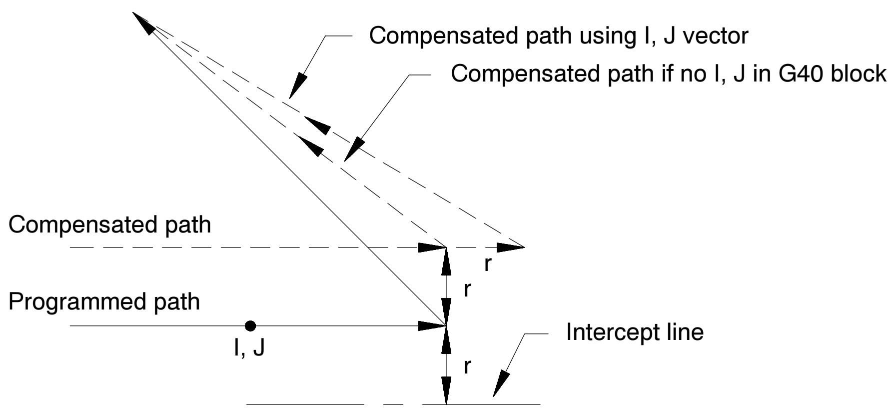
Exception is taken to the above figure when the change in length of the compensated path is more than one radius of the tool. In this special case, this offset is limited to one radius of the tool. The direction of the offset is towards the point of intersection of the I, J, K vector with the current compensated tool path.

Figure 20.15
Exit Move Defined By An I, J, K Vector But Limited To Radius



If the vector defined by I, J, and/or K is parallel to the programmed tool path, the resulting exit move are offset in the opposite direction of the I, J, K vector by one radius of the tool.

Figure 20.16
Exit Move When I, J, K Vector Parallel to Programmed Tool Path



Important: If one I, J, and/or K value is programmed without the second one, the value of the second I, J, and/or K word defaults to 0.

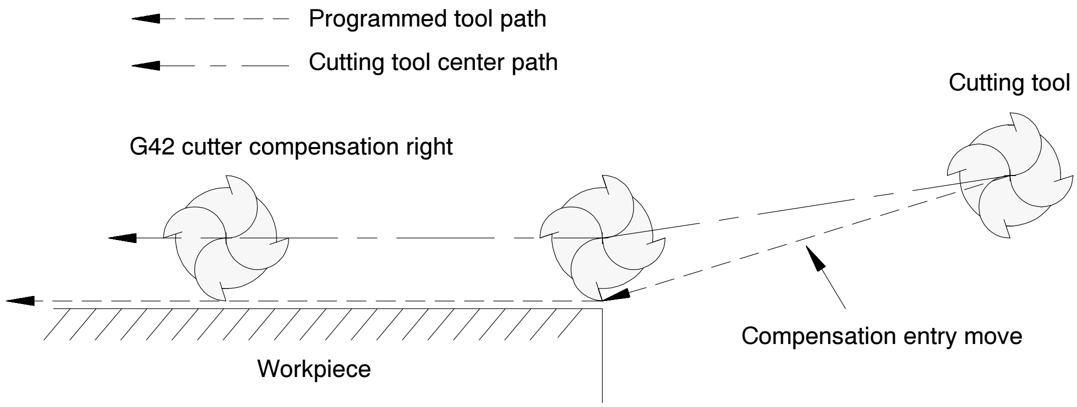
Cutter Compensation (Type B)

The easiest way to demonstrate the actual tool paths taken by the cutting tool when using cutter compensation type B is by pictorial representation. The following subsections give a brief description of the cutter path along with a figure to clarify the description.

Cutter Compensation Type B Entry Moves

An entry move is defined as the path that the cutting tool takes when the cutter compensation function first becomes activated in a program. The following figure gives an example of a typical entry move:

Figure 20.17
Cutter Compensation Entry Move

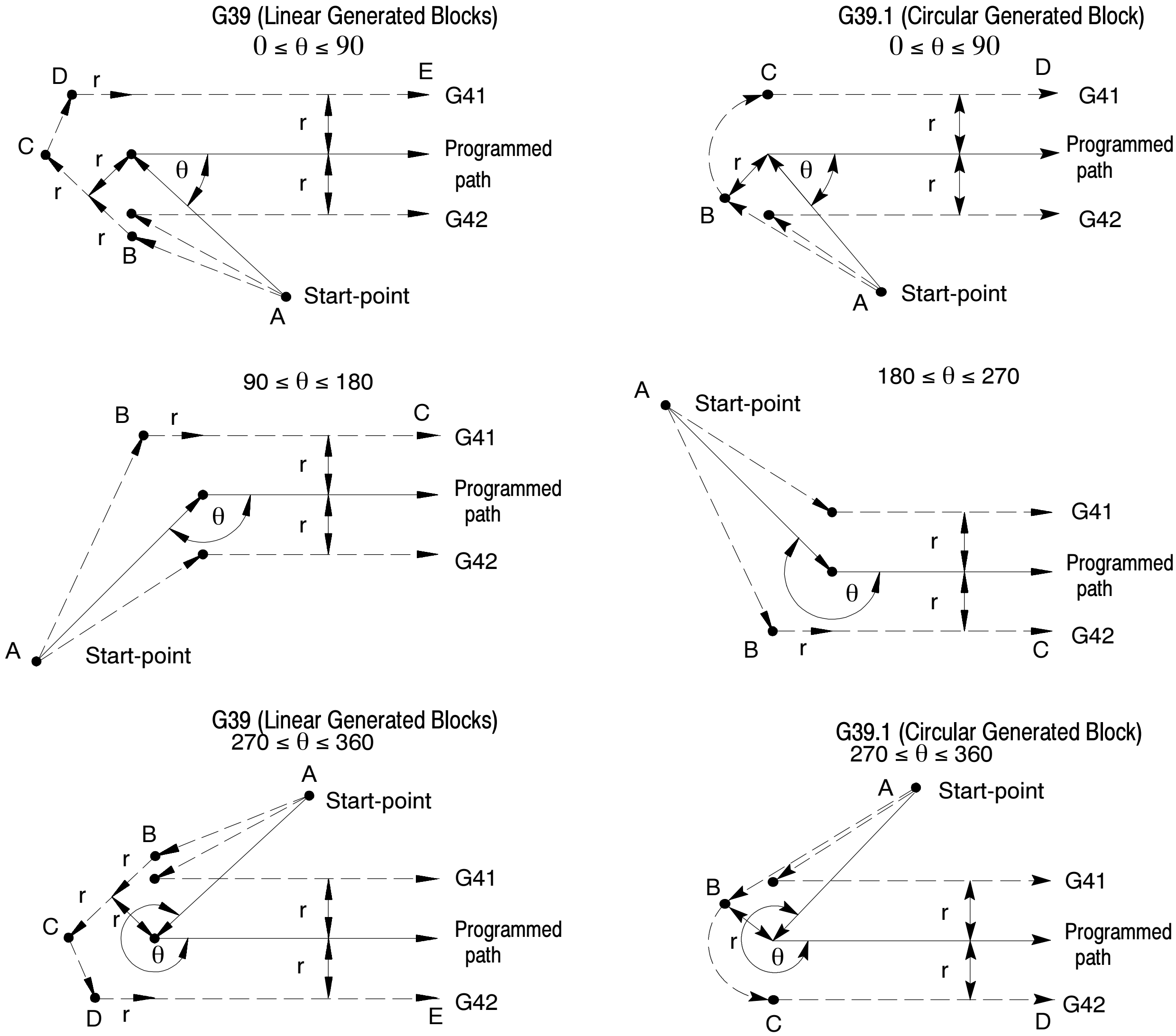


Important: Any entry move into cutter compensation must be a linear move. Initial activation of cutter compensation by designation of either the G41 or G42 commands in a circular cutting mode (G02 or G03) is not allowed. The G41 or G42 commands may be designated in a circular block to change cutter compensation direction, as long as cutter compensation is already active.

The entry move of the cutting tool for type B cutter compensation can generate extra motion blocks to attempt to prevent gouging of the part as may sometimes occur using compensation type A. Type B cutter compensation keeps the cutting tool at least one radius away from the start-point of the next block at all times during an entry move. The final end-point of the entry move is a position at right angles to and on the left or right side of the next programmed move in the currently defined plane.

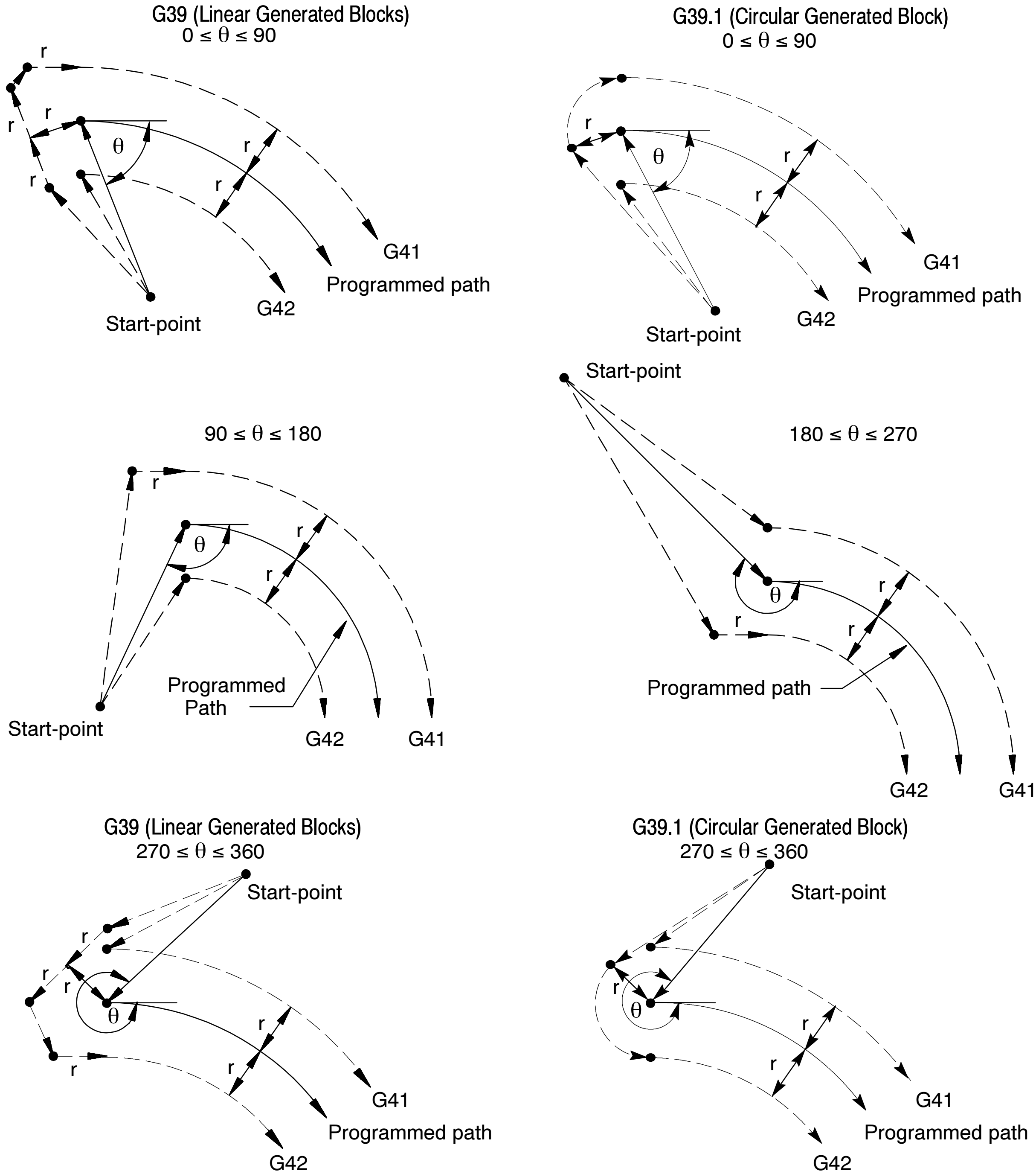
Figures 20.18 and 20.19 show examples of typical entry moves using type B cutter compensation:

Figure 20.18
Tool Path for Entry Move Straight Line-to-Straight Line



If the next programmed move is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the start-point of that circular move.

Figure 20.19
Tool Path For Entry Move Straight Line-to-Arc



There is no limit to the number of blocks that may follow the programming of G41 or G42 before an entry move takes place. The entry move are always the same regardless of the number of blocks that do not program motion in the current plane for compensation.

Example 20.7
Sample Entry Move After Nonmotion Blocks

Assume current compensation plane is the XY plane.

```
N01X0Y0F500;
N2G41D1;        This block commands compensation left.
N3Z1;           This is not the entry block since no axis
                motion takes place in the current plane.
N4...;          No axis motion in current plane.
N5...;          No axis motion in current plane.
N6...;          No axis motion in current plane.
"               "
"               "
"               "
N999X1Y1;       This is the entry move for the previously
                programmed G41.
M30;
```

The system installer selects in AMP the maximum number of nonmotion blocks that are to be allowed during cutter compensation before the entry move must be reinitialized.

Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

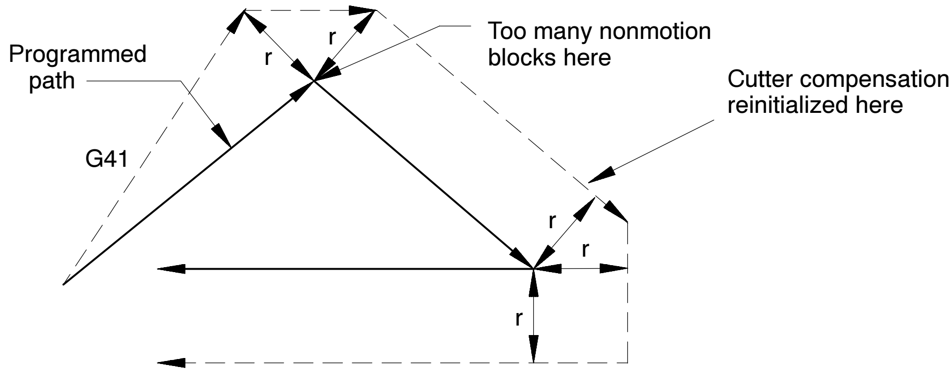
For example, assume that the system installer has designated that only two nonmotion blocks may be performed. If more than two blocks following the entry move do not contain axis motion in the current plane, then the entry moves are performed again at the next block containing axis motion in the current plane.

Example 20.8
Too Many Nonmotion Blocks After Entry Block

Assume current plane to be the XY plane and the system installer has designated that only two nonmotion blocks may be performed before cutter compensation is reinitialized.

```
N1X0Y0F500;
N2G41D1X1Y1;    Entry move cutter compensation left.
N3Z2;           No axis motion in current plane.
N4...;          No axis motion in current plane.
N5X4Y-4;        New entry move cutter compensation left.
M30;
```

Figure 20.20
Entry Move Followed By Too Many Nonmotion Blocks



Cutter Compensation Type B Exit Moves

The cutter compensation feature is cancelled by programming G40. The path that is taken when the tool leaves cutter compensation is referred to as the exit move. The path that the tool follows during an exit move is dependant on:

- The direction of compensation (G41 or G42).
- The angle between the last motion made in cutter compensation (in the current compensation plane) and the motion of the of the exit move.

Designating a tool offset number D00 in a program does not cancel cutter compensation and does not generate an exit move. Cutter compensation simply continues on as if a tool radius had been changed to a radius of zero (refer to page 20-43 on changing cutter radius). The exit move, if D00 is the active tool radius, is then equal to the programmed tool path.

Important: An exit move cannot be a circular move (G02 or G03). Any exit move must be programmed on a linear path. Any attempt to generate an exit move using a circular path generates a block format error.

Example 20.9 gives some sample exit move program blocks:

Example 20.9
Examples of Exit Move Blocks

Assume the current plane to be the XY plane.

```
N100X1Y1;
N110X3Y3G40;   Exit move.
N100X1Y1;
N110G40;
N120X3Y3;      Exit move.
N100X1Y1;
N110G40;
N120Z1;        No axis motion in the current plane.
N130...;       No axis motion in the current plane.
N140...;       No axis motion in the current plane.
”              ”
”              ”
N200X3Y3;      Exit move.
N100X1Y1;
N110Z1;        No axis motion in the current plane.
N120...;       No axis motion in the current plane.
N130...;       No axis motion in the current plane.
”              ”
”              ”
N200G40X3Y3;   Exit move.
```

All of the program blocks in Example 20.9 produce the same exit move provided that the number of nonmotion blocks in the compensation mode has not exceeded a value selected by the system installer in AMP.

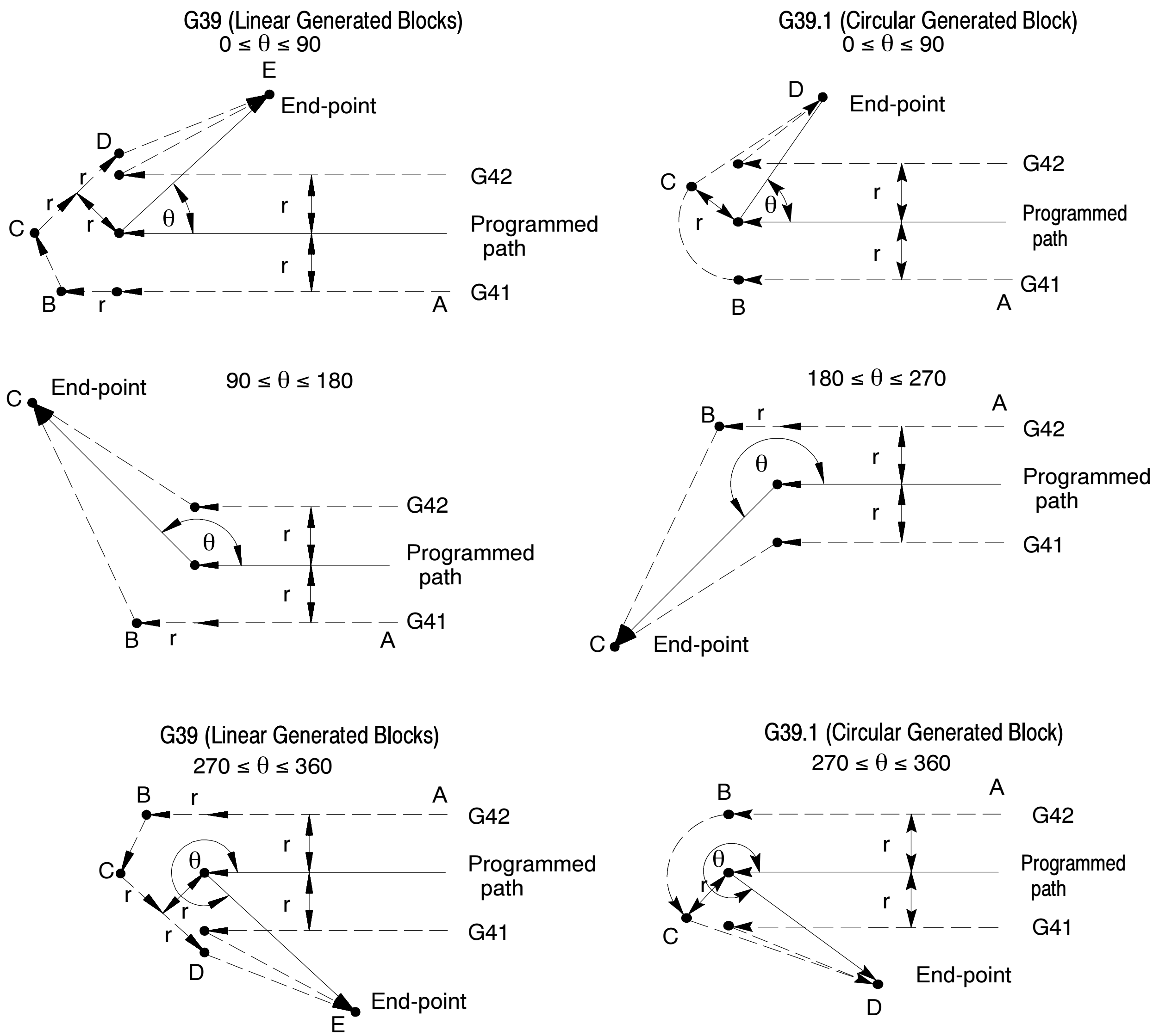
Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

The exit of the cutting tool for type B cutter compensation takes the shortest possible path to the end-point of the exit move only for inside corners. For outside tool corners, the cutting tool always remains at least the radius of the cutting tool away from the end-point of the last move in compensation. It is possible to redefine the start-point using an I, J, and/or K word as described later in this section. The end-point of the exit move is no longer offset to the left or right.

Figures 20.21 and 20.22 show examples of typical exit moves using type B cutter compensation. All examples assume that the number of nonmotion blocks before the designation of the G40 command has not exceeded the number allowed as determined by the system installer in AMP.

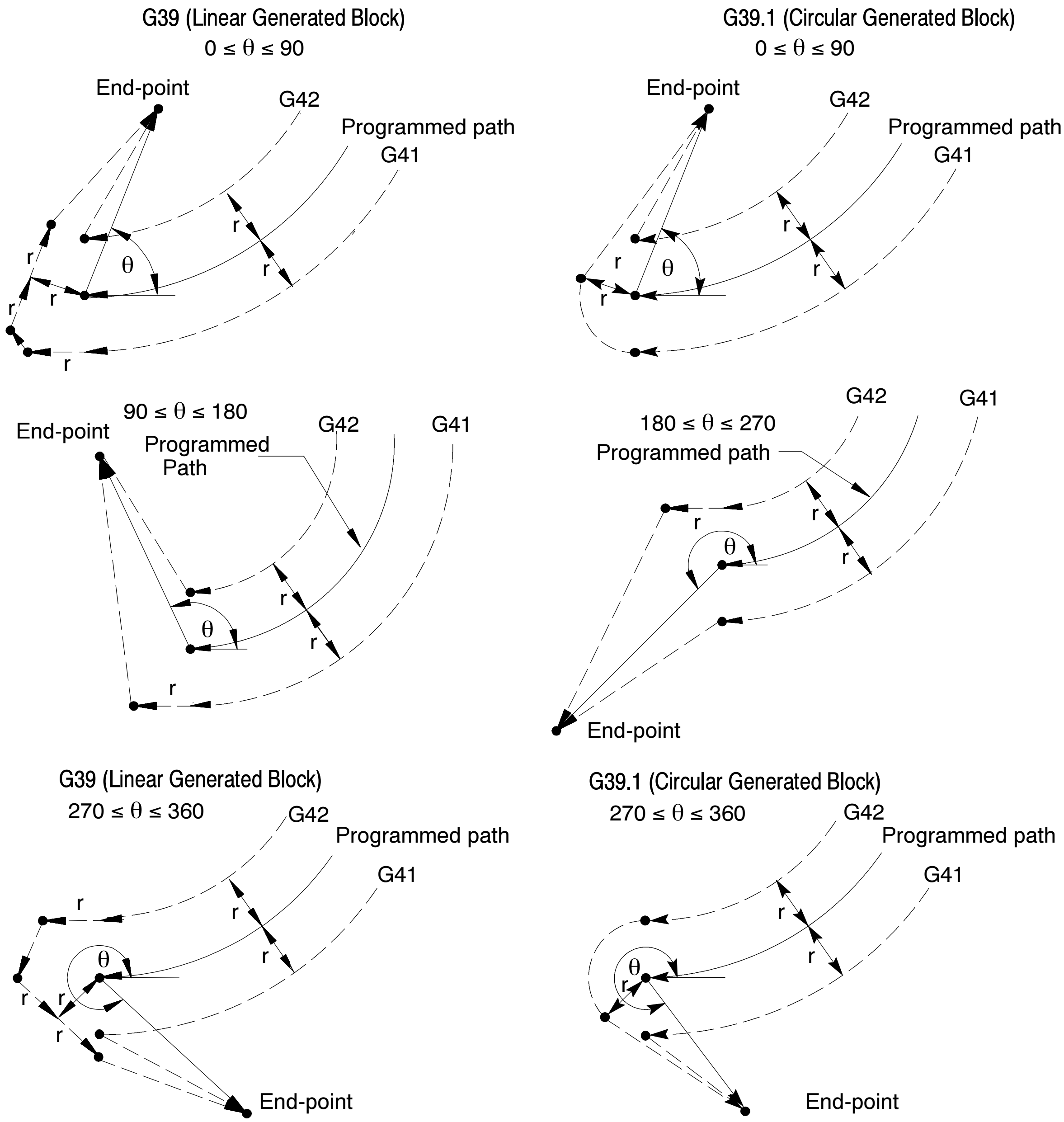
Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

Figure 20.21
Tool Path For Exit Move Straight Line-to-Straight Line



If the last programmed move is circular (an arc), the tool is positioned at right angles to a tangent line drawn from the end-point of that circular move.

Figure 20.22
Tool Path For Exit Move Arc-to-Straight Line



Figures 20.21 and 20.22 assume that the number of blocks not containing axes motion in the current plane, following G40 before the exit move takes place, does not exceed an amount selected in AMP by the system installer. If the number of nonmotion blocks following G40 exceeds the limit, the control generates its own exit move. This may often cause over-cutting of the part.

Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

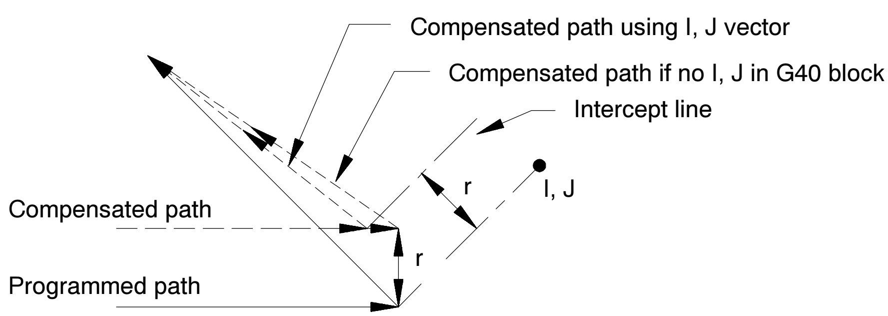
It is possible to modify the path that the tool takes for an exit move by including an I, J, and/or K word in the exit move. Only the I, J, or K words that represent values in the current plane are programmed in the block containing the exit move. I, J, and K correspond to the X, Y, and Z axes respectively.

The I, J, and K words in the exit move block define a vector that is used by the control to redefine the end-point of the previously compensated move. I, J, and K words are always programmed as incremental values regardless of the current mode (G90 or G91).

The vector defined by the I, J, and/or K words is along a line drawn from the end-point of the programmed path a distance as designated with the I, J, and/or K words. The I, J, and/or K words must be in the currently defined plane.

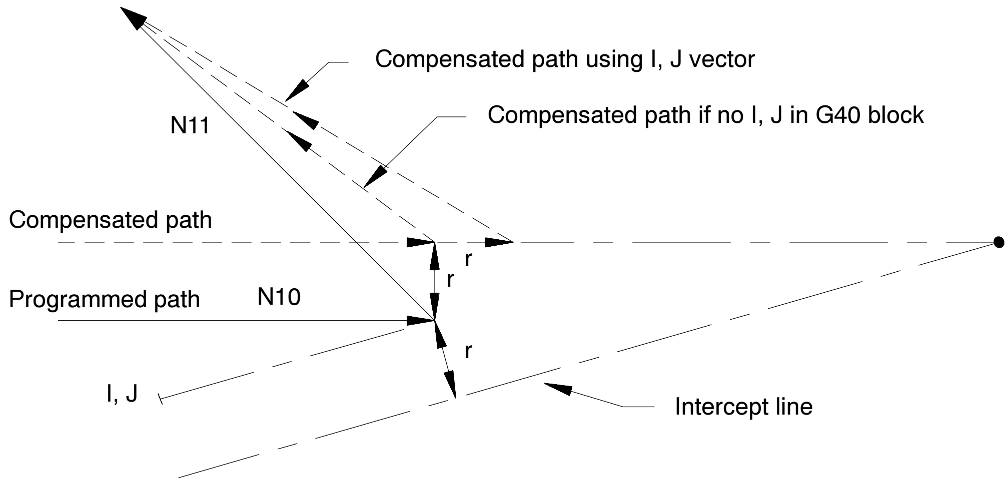
A new vector is then defined parallel to the vector defined by the I, J, and/or K word and offset from this vector in the direction and amount of the currently active offset (G41 or G42). The intersection of this new vector with the current compensated tool path define a point which is the new end-point of the last programmed compensation move.

Figure 20.23
Exit Move Defined by an I, J, K Vector



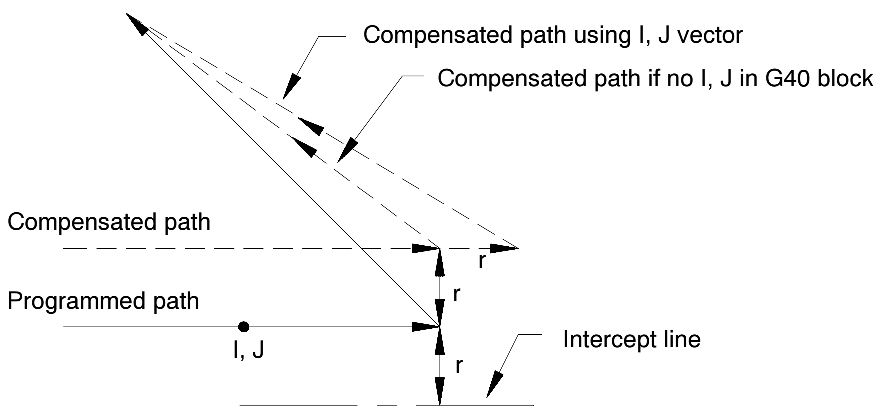
Exception is taken to the above figure when the change in length of the compensated path is more than one radius of the tool. In this special case, this offset is limited to one radius of the tool. The direction of the offset is towards the point of intersection of the I, J, or K vector and the current compensated tool path.

Figure 20.24
Exit Move Defined By An I, J, K Vector But Limited to Tool Radius



If the vector defined by I, J, and/or K is parallel to the programmed tool path, the resulting exit move are offset in the opposite direction of the I, J, K vector by one radius of the tool.

Figure 20.25
Exit Move When I, J, K Vector is Parallel to Programmed Tool Path



Important: If one I, J, and/or K value is programmed without the second one, the value of the second I, J, and/or K word defaults to 0.

Tool Path During Cutter Comensation

Except for entry and exit moves, the basic tool paths generated during cutter compensation are the same for types A and B cutter compensation. The paths taken are a function of the angle between tool paths (whether G41 tool-left or G42 tool-right is specified) and the radius of the cutting tool.

Important: If at any time during the execution of cutter compensation blocks a block reset is performed, the cutter compensation function is re-initialized, and the next move acts as an entry move as described in an earlier section.

Important: When cutting arcs with cutter compensation active, the control may need to adjust the programmed feedrate to maintain cutting speed. Refer to Chapter 17 for details on feedrates during cutter compensation.

The control generates extra motion blocks when necessary to keep the cutting tool in tolerance of the desired tool path. This becomes necessary when the intersection of tool paths is an outside tool path (as defined on page 20-1) that has an angle as follows:

- between 0 and 90 degrees during cutter compensation left (G41)
- between 270 and 360 degrees during cutter compensation right (G42)

Figures 20.26 through 20.29 illustrate the basic motion of the cutting tool as it executes program blocks during cutter compensation:

Figure 20.26
Cutter Compensation Tool Paths Straight Line-to-Straight Line

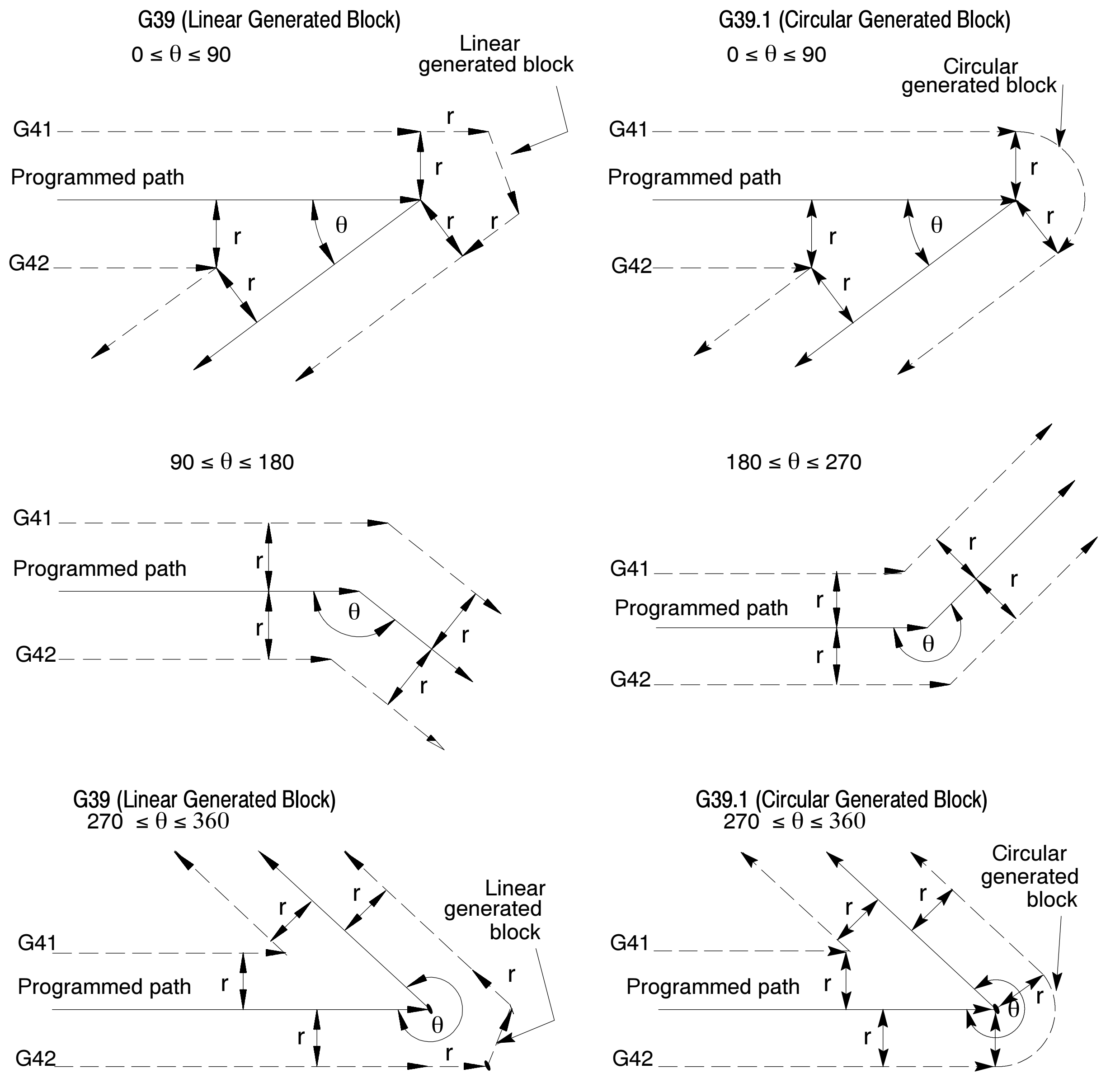


Figure 20.27
Cutter Compensation Tool Paths Straight Line-to-Arc

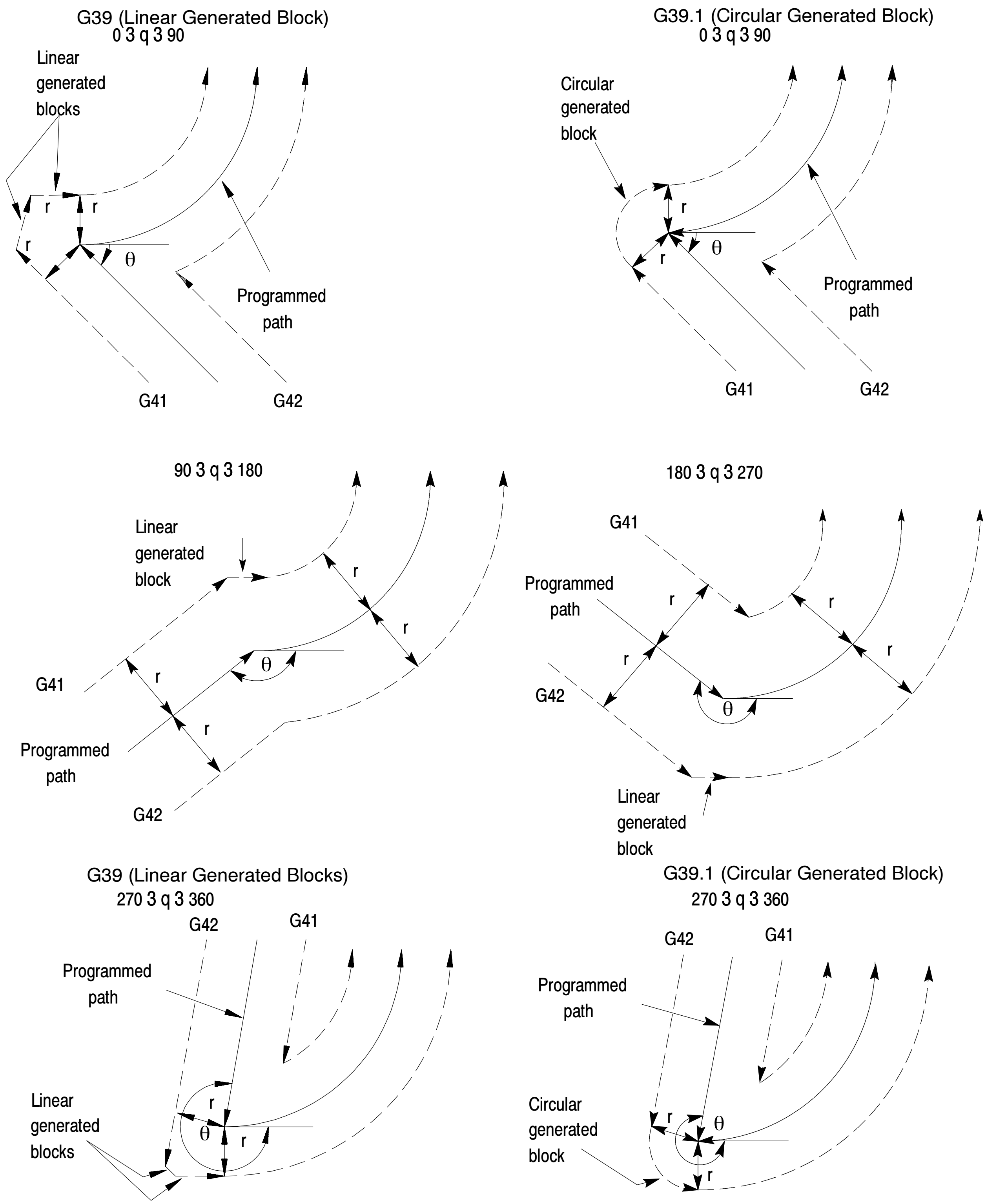


Figure 20.28
Cutter Compensation Tool Paths Arc-to-Straight Line

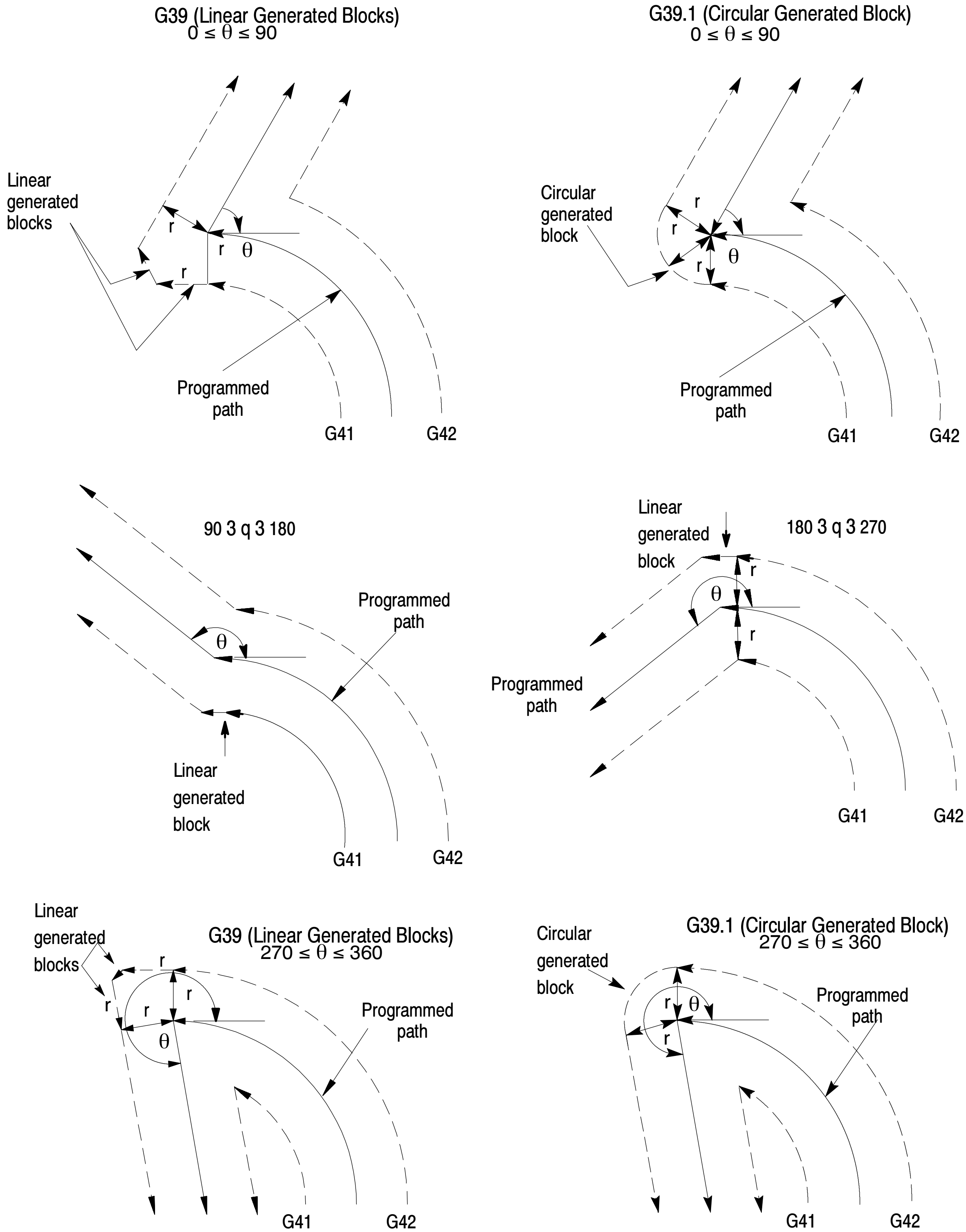
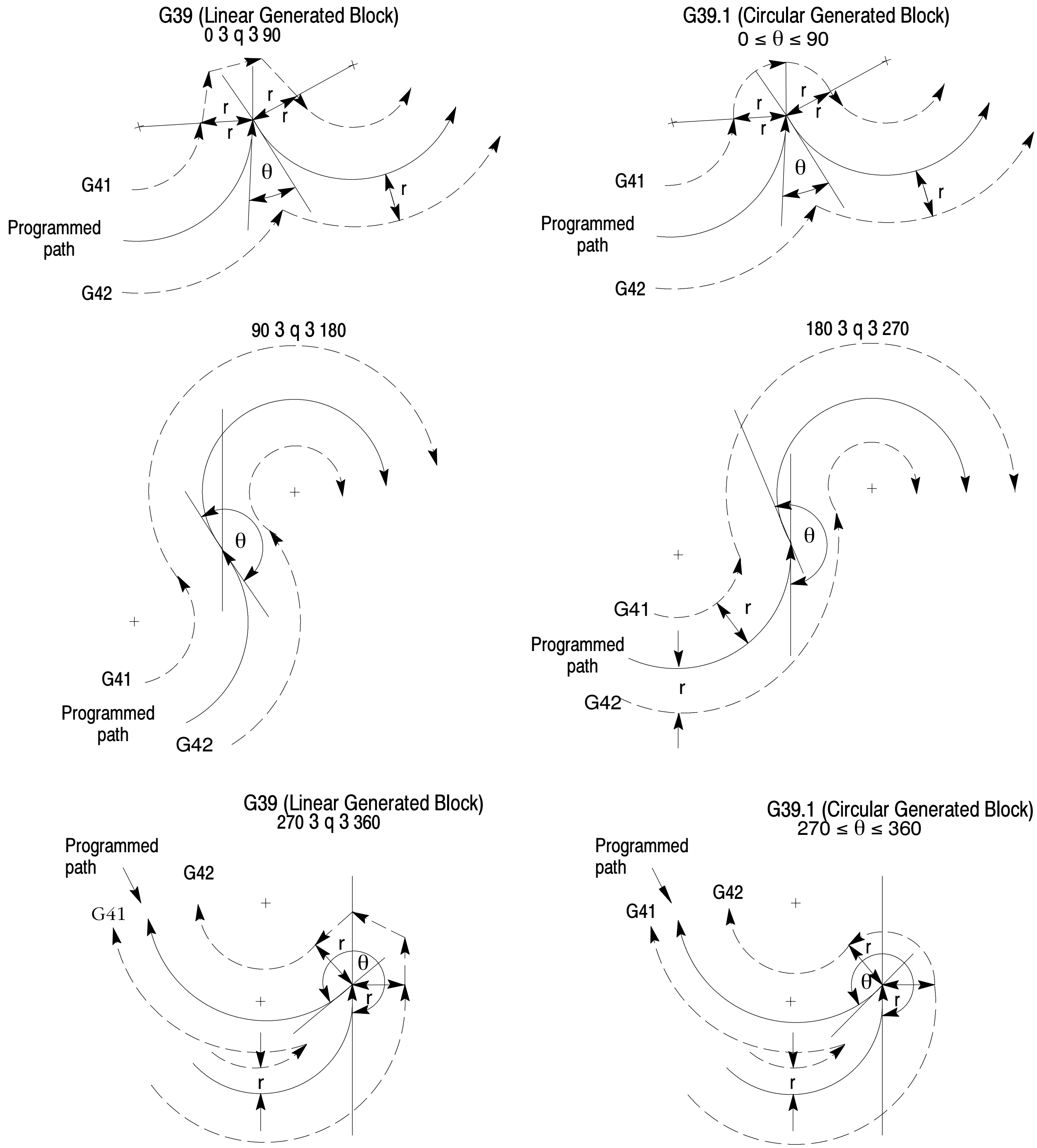


Figure 20.29
Cutter Compensation Tool Paths Arc-to-Arc



Cutter Compensation Special Cases

The following subsections describe possible tool paths that may be generated when programming one of the following during cutter compensation:

- changing cutter compensation direction (cross-over tool paths)
- exceeding the allowable number of consecutive nonmotion blocks during cutter compensation
- corner movement following a generated block
- changing cutter radius during cutter compensation
- effect on cutter compensation when interrupting a program to execute either a MDI program or a manual move
- changing or offsetting current work coordinate system during cutter compensation
- moving to and from machine home and secondary machine home

Changing Cutter Compensation Direction

This section describes the resulting tool path when a change in compensation direction (left or right) is programmed. This may result in the cutting tool crossing over the programmed tool path as compensation changes from left to right or right to left.

Linear Tool Path-to-Linear Tool Path

The following figures show the tool path taken when cutter compensation is changed from G41 to G42 during the execution of two linear program moves.

The control generates two points when changing cutter compensation direction, called point 1 and point 2. Point 1 is the final tool position before compensation direction is changed (at right angles to the end-point of the programmed tool path offset by one tool radius). Point 2 is the desired tool position for the start of the first block using the changed compensation direction (at right angles to the start-point of the motion block that changes compensation direction and offset by the tool radius). The control generates the motion block that connects point 1 to point 2 as shown in Figure 20.30 through 20.33.

Figure 20.30
Linear-to-Linear Change with Block Direction Reversed

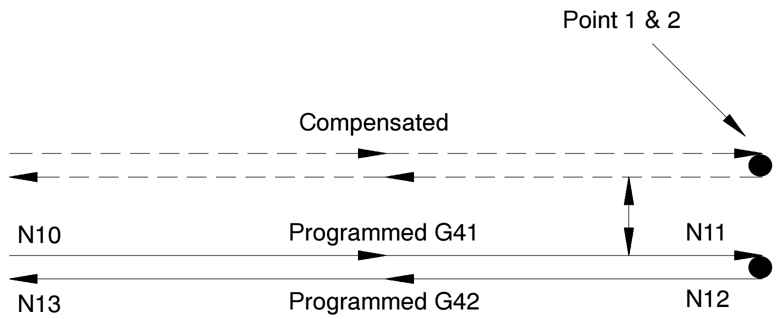


Figure 20.31
Linear-to-Linear Change with Tangential Motion Blocks

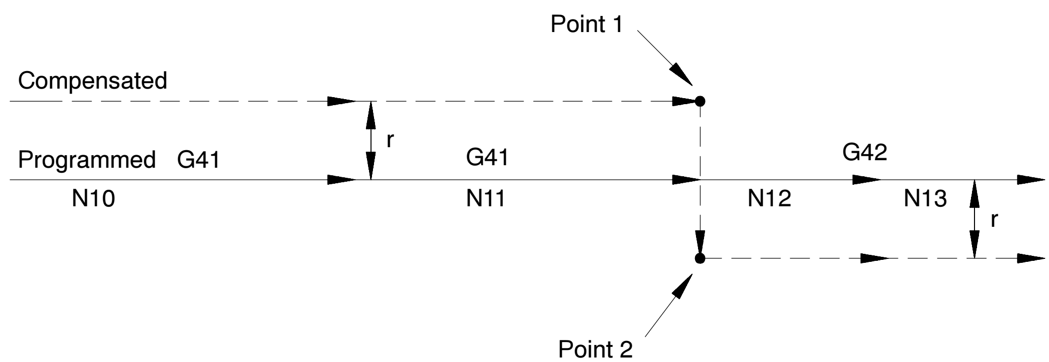


Figure 20.32
Linear-to-Linear Change with A Generated Block

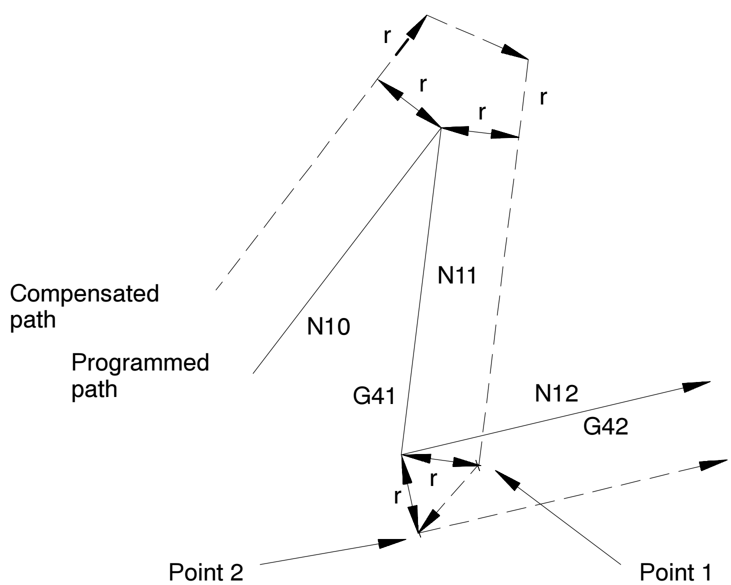
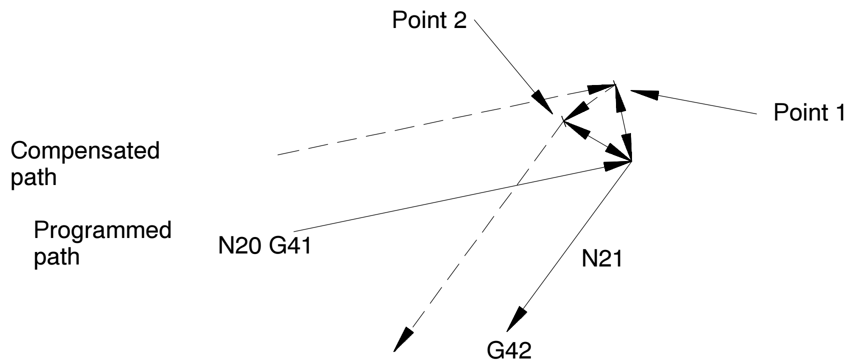


Figure 20.33
Linear-to-Linear Change with No Generated Block

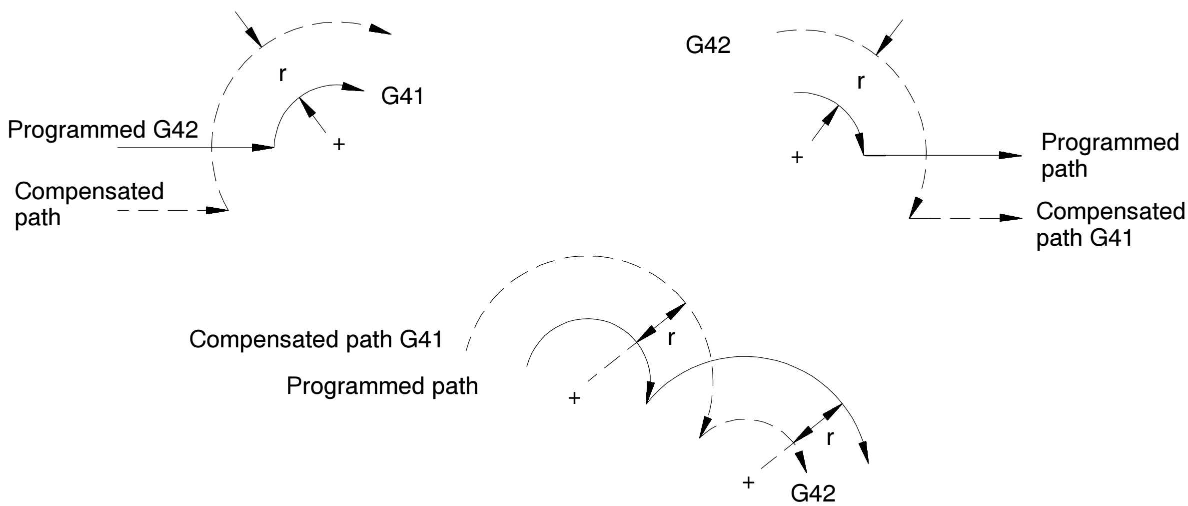


For one of the following cases that changes the cutter compensation direction, the control will attempt to find an intersection of the actual compensated tool paths:

Linear-to-Circular, Circular-to-Linear, or Circular-to-Circular Tool Paths

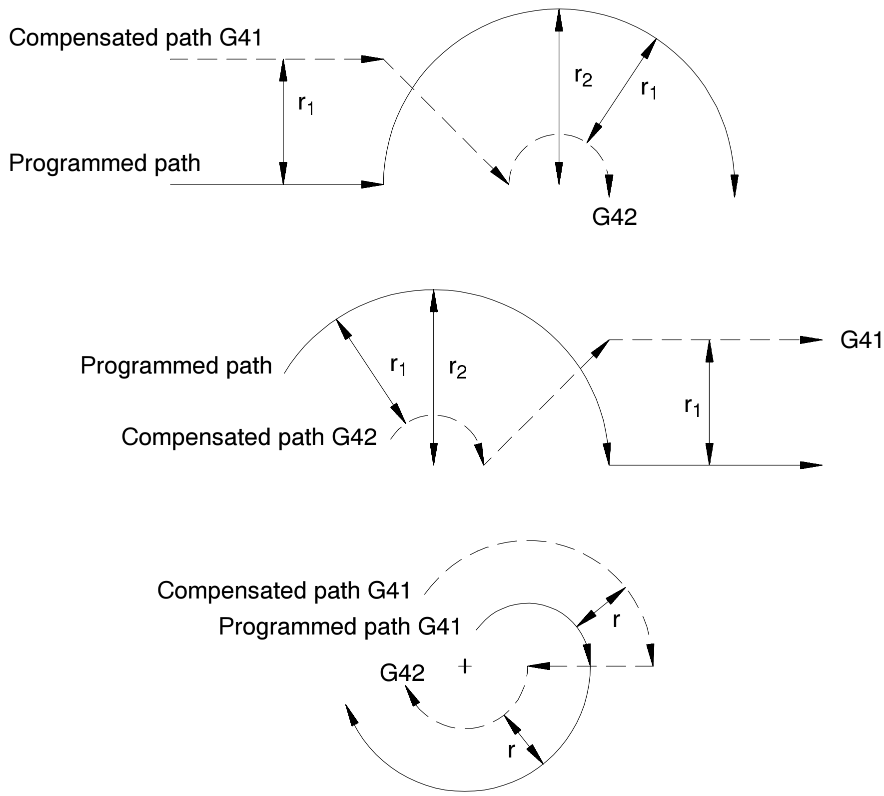
For the following cases that change the cutter compensation direction, the control attempts to find an intersection of the actual compensated tool paths. If the control finds an intersection, it modifies the end-point of the original compensated tool path and the start-point of the new compensated tool path to equal that intersection. (see Figure 20.34).

Figure 20.34
Change in Compensation With Actual Tool Path Intersection



If no intersections of the actual tool paths exist, the compensated tool path is as shown in Figure 20.35. The end-point of the last move in the original compensation direction is at right angles to that block's programmed tool path, and the start-point of the move in the new compensation direction is at right angles to that block's start-point.

Figure 20.35
Change in Compensation with No Possible Tool Path Intersections



Too Many Nonmotion Blocks

The control always looks ahead to the next motion block to determine the actual tool path taken for a motion block in cutter compensation. If the next block is not a motion block, the control continues to scan ahead for a motion block until it either detects one or the allowable number of nonmotion blocks, as set in AMP, has been exceeded. Refer to documentation prepared by the system installer for the allowable number of nonmotion blocks allowed in a specific system.

Important: The definition of a nonmotion block is any block within a program that does not actually generate the movement of one of the axes in the current compensated plane. Blocks that are skipped by the control because of the block-delete feature (/) discussed in chapter 7 are also counted as a nonmotion block in cutter compensation regardless of the content of the skipped block.

If the control, when scanning ahead, does not find a motion block before the number of nonmotion blocks has been exceeded, it will not generate the normal cutter compensation move. Instead the control sets up the compensation move with an end-point one tool radius away from and at right angles to the programmed end-point.

In many cases, this may cause unwanted overcutting of a work piece. Figures 20.36 and 20.37 are example tool paths of programmed motion blocks followed by too many nonmotion blocks before the next move was made:

Figure 20.36
Too Many Nonmotion Blocks Following a Linear Move

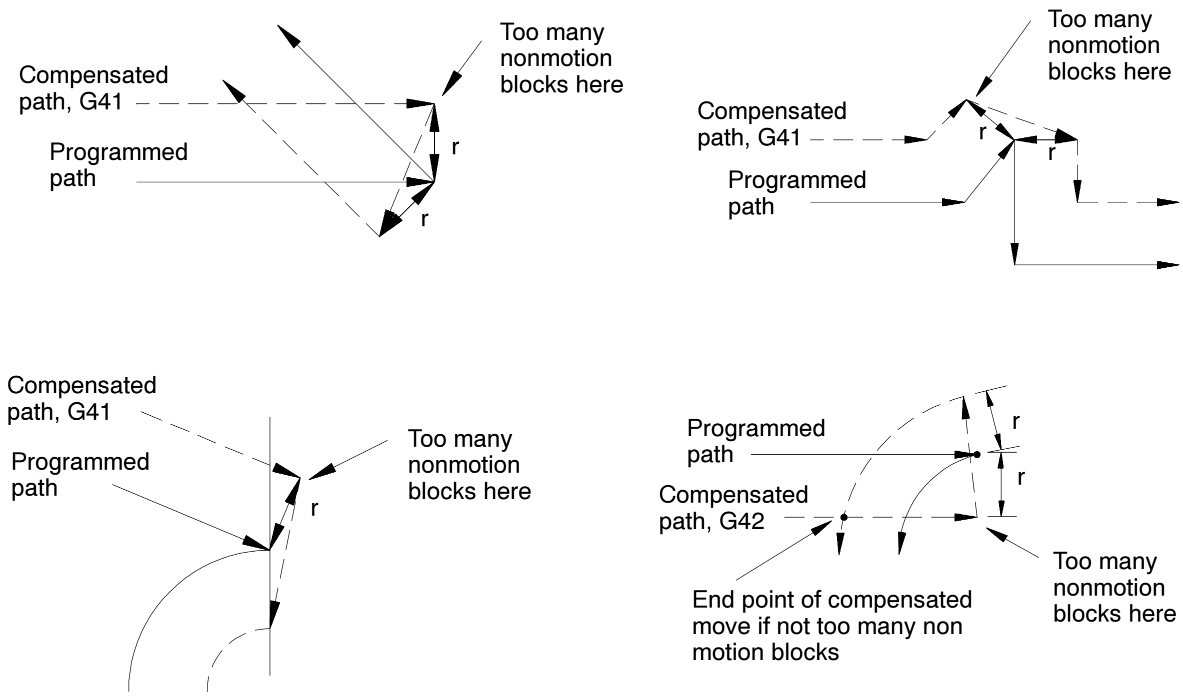


Figure 20.37
Too Many Nonmotion Blocks Following a Circular Move

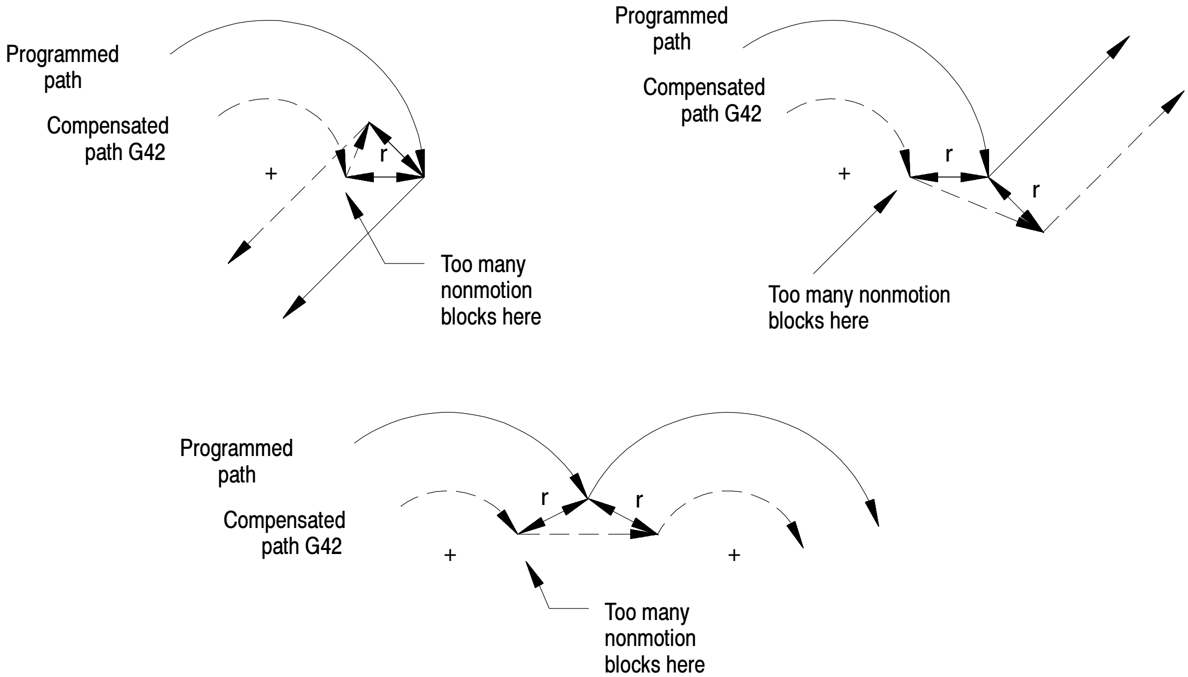


Figure 20.38
Too Many Nonmotion Blocks Following a Linear Move

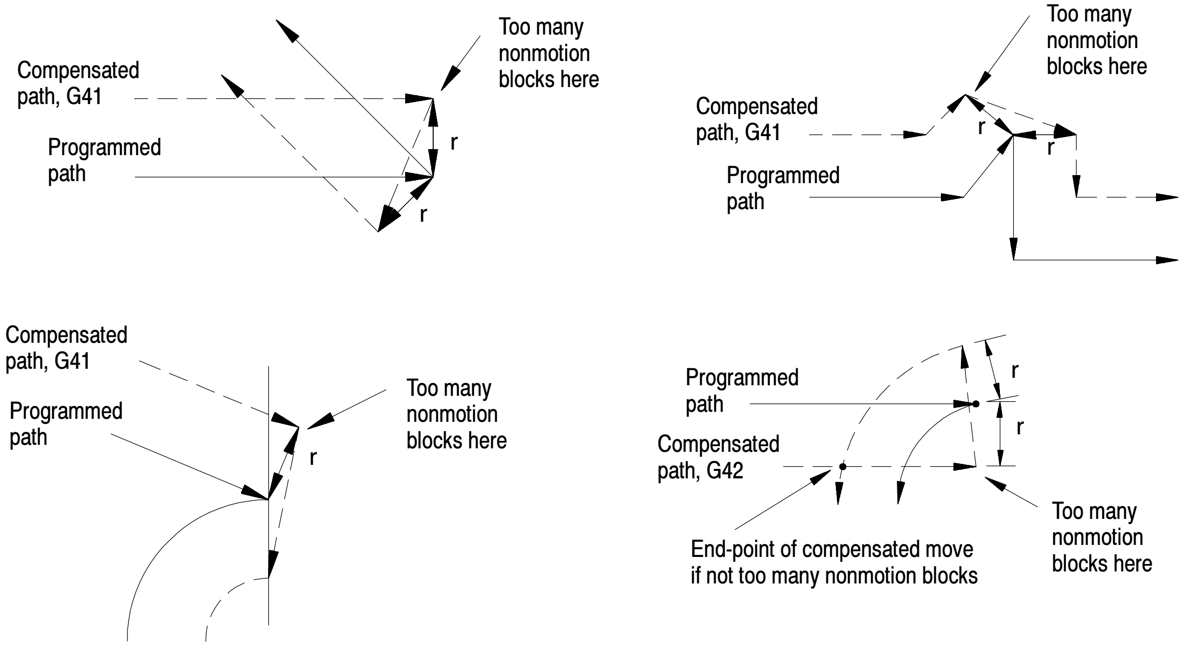
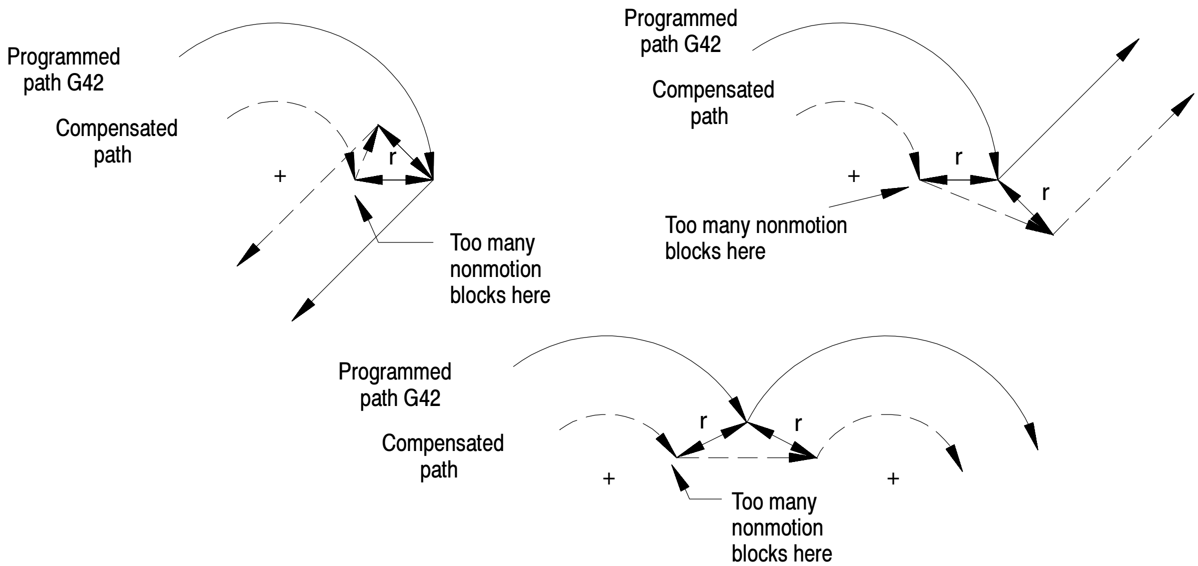


Figure 20.39
Too Many Nonmotion Blocks Following a Circular Move

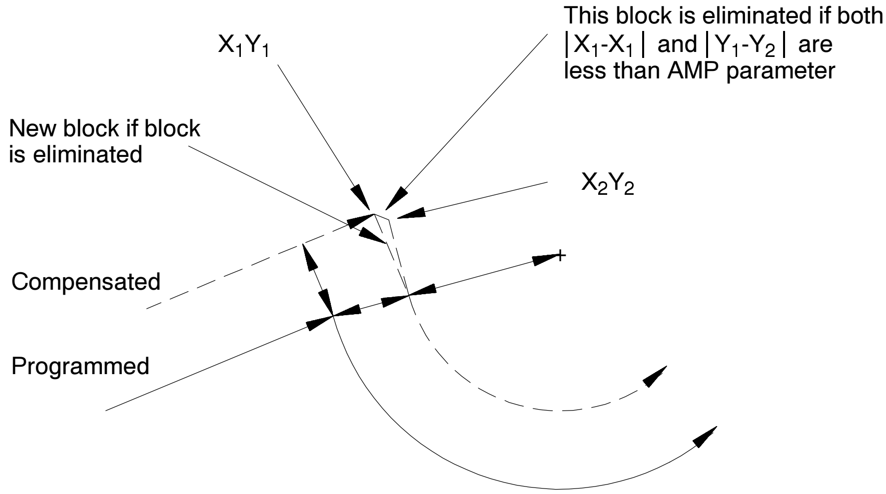


Corner Movement After Generated Blocks

Frequently the control must generate motion blocks to position the cutting tool in the proper alignment for a following compensated cutting move. These blocks are generated to make certain that the cutting tool remains at least one radius of the cutting tool away from the programmed cutting path at all times.

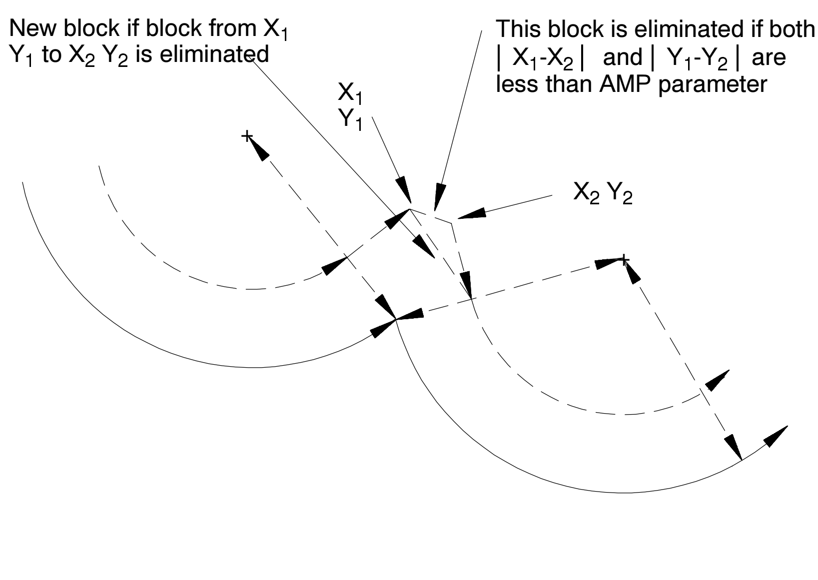
When the control generates two motion blocks, the length of the first generated block is checked against a minimum allowable length as determined in AMP by your system installer. The coordinate values for the current axes in the compensation plane are compared to the minimum allowed value. If both are less than the allowed value, then the control does not executes the first generated block. The path of the second generated block is then altered to position the cutting tool along a linear path to the original end-point of the second generated block. See Figure 20.40 for a pictorial representation.

Figure 20.40
Compensation Corner Movement for Two Generated Blocks



When the control generates three motion blocks, the length of the second generated block is checked against a minimum allowable length, 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, then the control does not execute the second generated block. The path of the third generated block is then altered to position the cutting tool along a linear path to the original end-point of the third generated block. See Figure 20.41 for a pictorial representation.

Figure 20.41
Compensation Corner Movement for Three Generated Blocks



Changing Cutter Radius During Compensation

If a tool becomes excessively worn, broken, or if any other reason requires the changing of the programmed tool radius, the cutter compensation should be cancelled and reinitialized after the tool has been changed. The following section describes the resulting tool path if, for some reason, it is desirable to program a change in cutter radius during cutter compensation.

Important: Slight overcutting may occur during Cutter Compensation, depending on the programmed path at the point where the change in cutter radius was made. To avoid overcutting, we recommend that you use a Mid-Start Program until the point of tool breakage.

Refer to chapter 3 for information about changing the active tool offset and page 20-3 on changing the programmed compensation diameter offset number.

Figures 20.42 through 20.44 are representations of the resulting tool paths after the programming of a change in the radius of the cutting tool. Assume in these figures that the programmed change to the tool radius is entered in block N11 which also contains the motion as described in the figure. The tool path taken when changing tool radius is dependant on the move immediately before the change in radius was programmed, the move that the change in radius was programmed in, and whether any generated motion blocks were made between these tool paths.

Figure 20.42 describes the tool path when the programmed moves are linear-to-linear.

Figure 20.42
Linear-to-Linear Change in Cutter Radius During Compensation

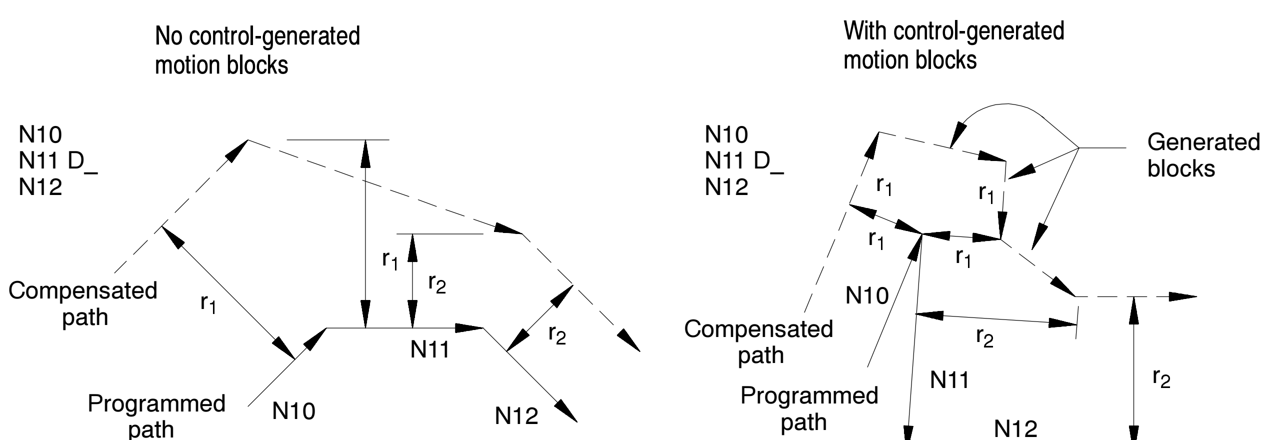


Figure 20.43 describes the tool path when the programmed moves are linear-to-circular.

Figure 20.43
Linear-to-Circular Change in Cutter Radius During Compensation

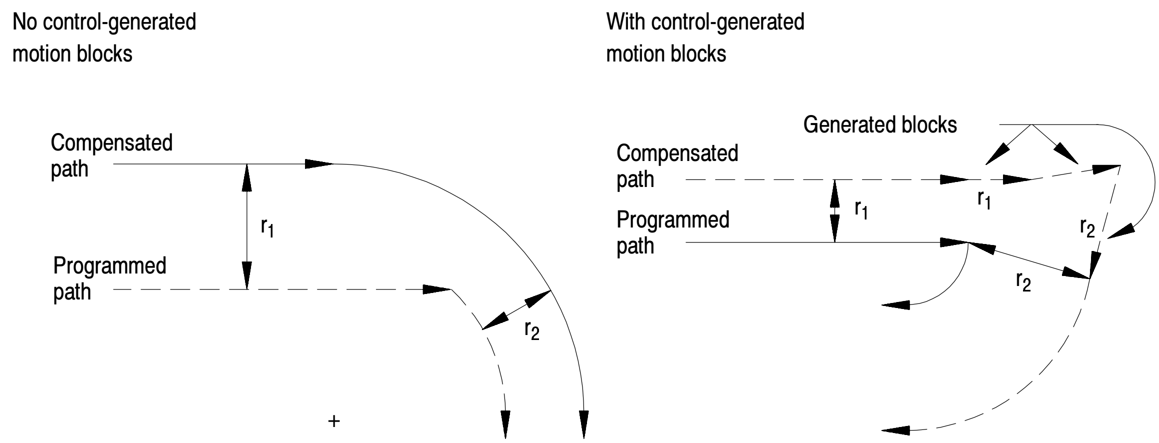
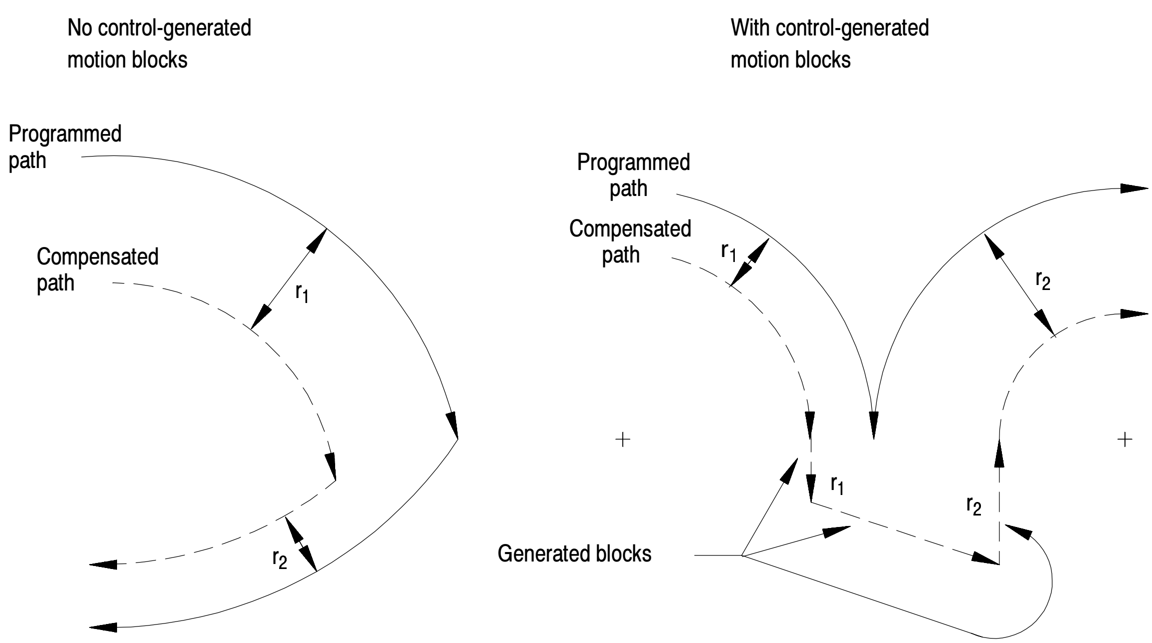


Figure 20.44 describes the tool path when the programmed moves are circular-to-circular.

Figure 20.44
Circular-to-Circular Change in Cutter Radius During Compensation



Change in Cutter Radius During Jog Retract.

This section describes a change in the cutter radius during a jog retract operation. This is a typical operation since the jog retract feature is often used when a tool becomes very worn or is broken. It may be necessary to replace the tool with a tool of a slightly different diameter. Cutter compensation is able to adjust to the new tool diameter.

Typically when the jog retract operation is performed, the tool is jogged away from the workpiece and then replaced. After it is replaced, you need to activate a different tool diameter offset value. This is done in either of two methods:

- The new offset number is activated by programming a new D-word in an MDI block.
- The new offset number is activated by using the **{ACTIVE OFFSET}** softkey found on the offset table screen. This feature is described in chapter 3.

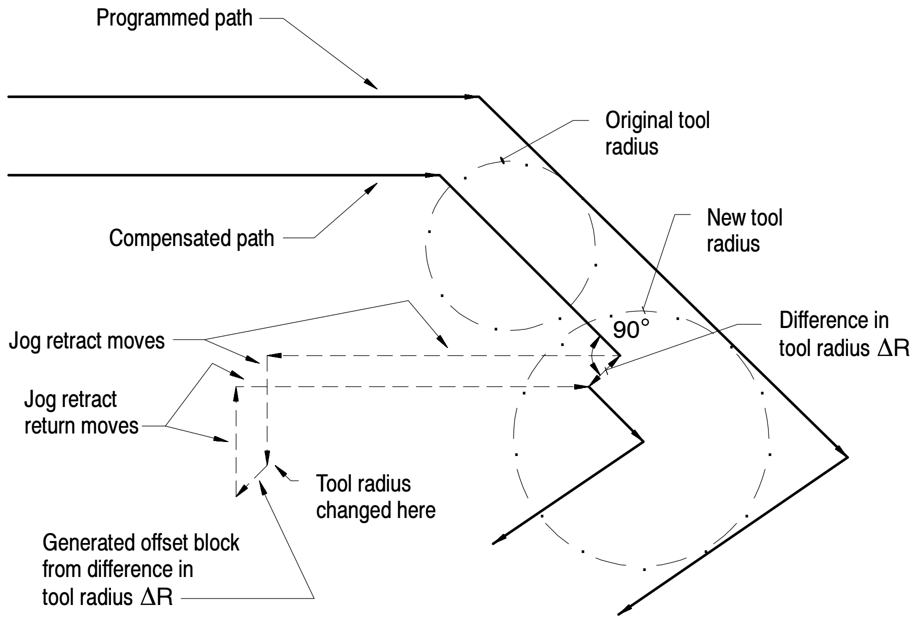
However the new offset is activated, cutter compensation is able to compensate for this new diameter by modifying the saved jogged path. This path is modified so that the new tool will cut the same part as the old tool. The absolute position of the machine, therefore, is different on the return path from what it was when jogging away from the part.

This jogged path is adjusted when you press the **<CYCLE STOP>** button to return from the jog retract. As soon as you press the **<CYCLE STOP>** button, the control generates a move that offsets the current tool position by the necessary distance. This distance is determined as the necessary distance the tool where would have to be positioned so that the exact same jog return paths can be used to return to the part and still have the end-point be offset from the original position by the difference in the cutter diameter.

ATTENTION: Make sure that this offset path will not cause any collisions with the part or the machine fixtures. The position of the tool when the tool change in jog retract is made should be a safe distance from the part and machine fixtures.

Figure 20.45 gives an example of a typical change in tool radius during jog retract with cutter compensation active.

Figure 20.45
Change in Cutter Radius During a Jog Retract



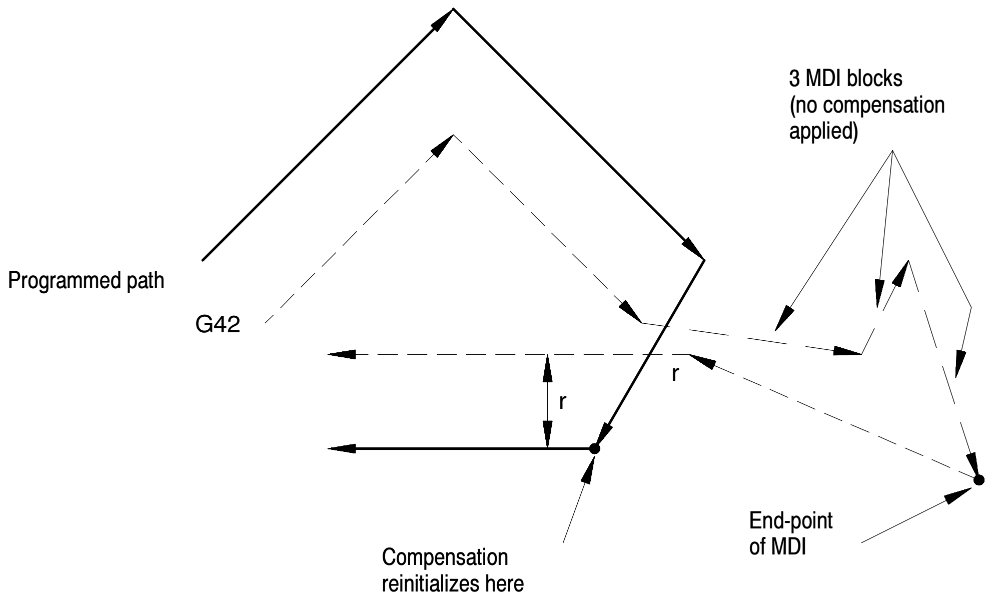
MDI or Manual Motion During Cutter Compensation

If exiting automatic mode and either a MDI motion block is executed or a manual jog motion is made, the cutter compensation feature, if active, reinitializes when the next motion block is executed in automatic mode. The compensation feature compensates the cutting tool one tool radius perpendicular to the tool path of the next motion block that is executed in automatic mode. In effect, the control generates its own entry move for compensation with the first compensated block being the next block executed in automatic operation.

Important: The cutter compensation feature is not available for any motion blocks that are programmed in MDI mode. The cutter compensation mode may be altered by programming either G41, G42, or G40; or the tool radius may be changed in an MDI program. However, none of the tool paths executed in MDI are compensated. No changes made to cutter compensation are applied until the next block executed in automatic mode.

Figure 20.46 is an example of the possible tool path that is taken when you interrupt an automatic operation during cutter compensation to execute MDI motion blocks. This same tool path applies also, if you interrupt cutter compensation to perform a manual jog move.

Figure 20.46
TTRC Interrupted with MDI Blocks



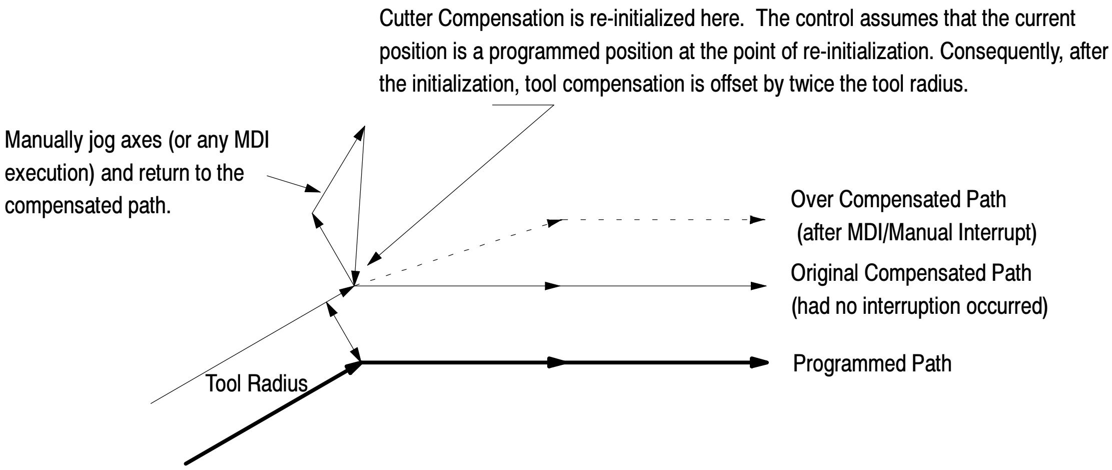
Important: If during cutter compensation, you switch out of automatic mode and either:

- generate axis motion in manual mode on an axis in the cutter compensation plane, or
- execute any block in MDI mode,

cutter compensation is re-initialized when you return to automatic mode.

This produces a path that is different from the path that would have been produced had the manual or MDI operation not been done, even if you returned the tool to the point of interrupt. In absolute mode the control returns to the originally compensated path after it executes a block that contains both axes in the compensation plane. In incremental mode, the compensated path remains offset by the additional tool radius. Figure 20.47 illustrates these conditions.

Figure 20.47
Cutter Compensation Re-Initialized after a Manual or MDI Operation.



Use the Jog Retract feature if you must jog the axes away from a compensated path. Jog retract prevents the overcompensation from occurring.

If you interrupt cutter compensation with a manual or MDI operation and the next programmed block is a circular block, the control generates an error when it tries to re-initialize cutter compensation. You can avoid this by using the jog retract feature instead of manual or MDI when you need to interrupt cutter compensation.

Unless **Cutter Compensation** is active, when a program recover is performed, the control automatically returns the program to the beginning of the block that was interrupted. In the case of power failure, the control will even reselect the program that was active prior to the interruption.

Moving to/from Machine Home

We recommend that you cancel cutter compensation using a G40 command before the execution of a return to or from machine home, or a return to or from the secondary machine home. This refers to the operations performed when the control executes either the G28, G29, or G30 commands as described on page 20-9.

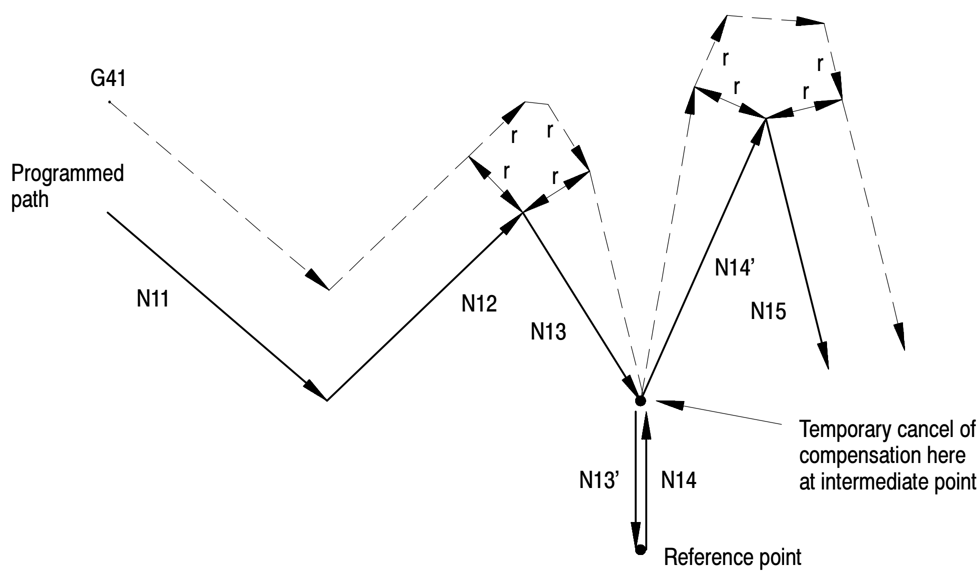
If compensation is not cancelled by using a G40 command, the control automatically, temporarily cancels compensation for the return to machine home or secondary machine home operations. This is done by using the move to the intermediate point, as designated when the operation was performed, as an exit move for compensation.

Important: An intermediate point should always be programmed for a return-to-home operation if cutter compensation is active. If no intermediate point is specified, the control executes the move prior to the return-to-home operation as an exit move. This may cause undesired over-cutting of the part.

If compensation was not cancelled using a G40 command before returning to machine or secondary home points, the control automatically re-initializes cutter compensation for the return from machine or secondary home points. This is done by using the move to the intermediate point, designated when the operation is performed, as an entry move for compensation.

Figure 20.48 gives an example of either a G28 or G30 block followed by a G29 block.

Figure 20.48
Cutter Compensation During G28, G30, and G29 Blocks



Changing or Offsetting Work Coordinate System

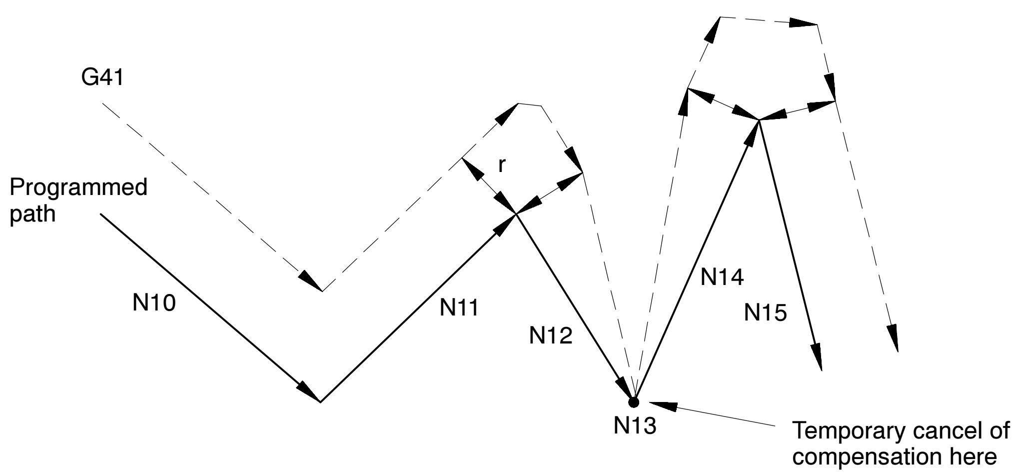
We recommend that cutter compensation be cancelled using a G40 command before any modifications to the current work coordinate system are made including any offsets or any change of the coordinate system (G54-G59.3).

Important: Changes can only be made, to axes in the cutter compensation plane, in the executing part program or using MDI. Changes to axes not in the current cutter compensation plane can be made manually in the offset tables (using the softkeys). Changes to work coordinate system offsets for axes that are in the active cutter compensation plane can not be performed manually in the offset tables.

If compensation is not cancelled using a G40 command, the control automatically, temporarily cancels compensation for the change in work coordinate system. This is done by using the last compensated move in the current coordinate system as an exit move for compensation. The control then automatically re-initializes cutter compensation after the new work coordinate system is established. This is done by using the first move in the new coordinate system that is in the compensation plane as a entry move for compensation. This repeat setup occurs even if the change to the coordinate system is not in the active cutter compensation plane.

Figure 20.49 gives an example of programming a G92 offset to the work coordinate system. The same figure, however, would apply to any change in the work coordinate system.

Figure 20.49
Cutter Compensation During G92 Offset to Work Coordinate System



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 9/PC control has a total of 21 set-up buffers. Different features require the use of some of these set-up buffers. One is always used for the currently executing block. Cutter compensation requires at least 3 of these buffers; other features also occasionally require the use of some of these buffers. Any remaining set-up buffers are used for Block Look-Ahead, one buffer for each block look-ahead.

At times (especially possible during cutter compensation) the control may not have enough look-ahead blocks to correctly execute the current block. When this happens, the control automatically starts disabling the block retrace feature. The block retrace feature uses one set-up buffer for every retraceable block. The number of re-traceable blocks is set in AMP by the system installer (a maximum of 15 is possible). As the control starts disabling the block-retrace feature, it decreases the number of available retraceable blocks until either there are sufficient set-up buffers available to successfully execute the current program, or until there are no more block-retrace blocks left. The control displays a message on line 2 of the CRT if it needs to disable some of the block-retrace feature.

Note that using too many buffers for block-retrace is not a recommended method of operating the control. The larger the number of look ahead blocks that the control is scanning, the more efficiently the control executes programs. It is recommended that the number of retrace blocks that is available to the block-retrace feature be limited to the minimum number required on the system (set the number of block-retrace blocks as low as possible for a specific application). This greatly improves program execution and prevent the necessity of the control deleting block retrace blocks.

Error Detection

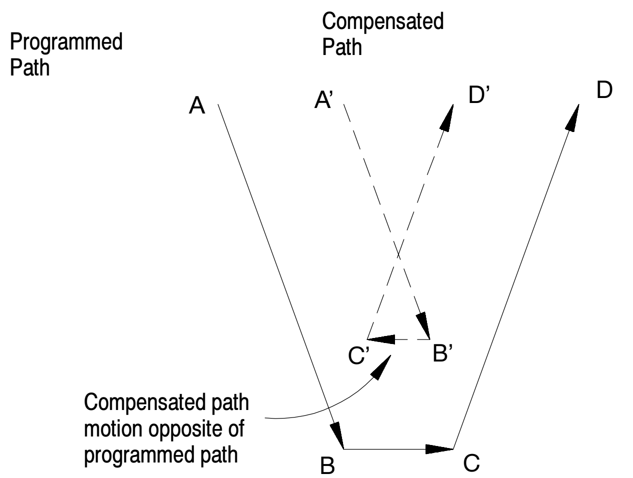
Error detection for cutter compensation blocks can be separated into three categories:

- Backwards motion detection
- Circular departure too small
- Interference

Backwards Motion Detection

The compensated tool path is parallel to, but in the opposite direction of the programmed tool path.

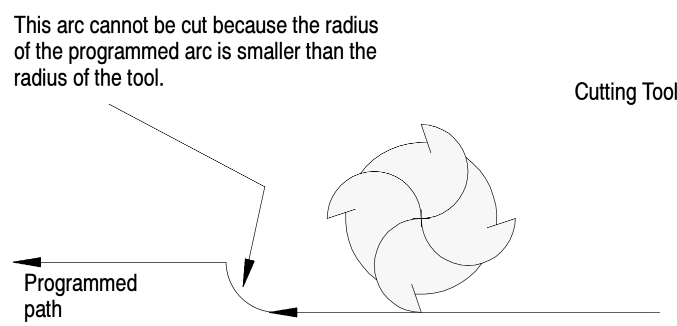
Figure 20.50
Typical Backwards Motion Error



Circular Departure Too Small

This error is generated when the cutter radius is larger than the radius of the programmed arc. Note this form of compensation error cannot be disabled with an M-code. Programming this contour with tool tip radius compensation on always generates an error.

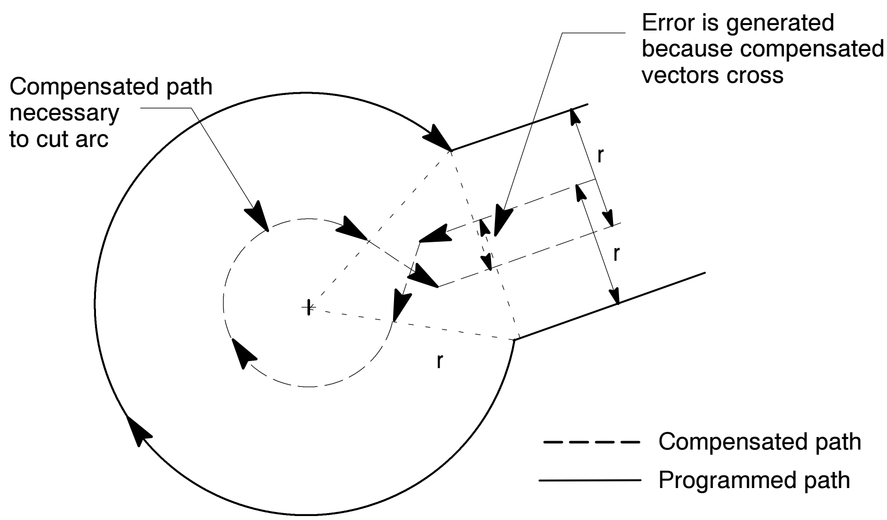
Figure 20.51
Typical Circular Departure Error



Interference

This error occurs when compensation vectors intersect. Normally when this intersection occurs, a backwards motion error is generated; however, a few special cases exist that are caught only by interference error detection.

Figure 20.52
Typical Interference Error



Disabling Error Detection

If so desired, all of the above error detection (with the exception of circular departure-too-small cases) can be disabled for a specific block or portion of a part program. To disable the error detection for a specific block, the system installer must have defined an M-code in AMP. By programming this M-code in a block, all error detection for cutter compensation may be disabled. Error detection is disabled until another M-code defined in AMP to re-enable error detection is programmed in a block.

Important: Circular departure too small cases cannot be disabled. The control cannot execute a compensated path when this error occurs.

The default condition is error detection enabled. Default values for these M-codes are:

M-code:	Error detection:
M800	disables
M801	enables

Error detection M-codes are only functional when cutter compensation is active. Cutter compensation is active when the control is in G41 or G42 mode and has already made the entry move into compensation. If an M800 or M801 is programmed in G40 mode or before the entry move into cutter compensation takes place, the M code is ignored.

If error detection is disabled in cutter compensation, and cutter compensation is exited (G40 programmed), the next time cutter compensation is reactivated error detection will be reactivated automatically. Error detection is always automatically enabled when cutter compensation is activated.

Refer to documentation prepared by your system installer for the M-codes used on your specific system.

END OF CHAPTER

Using Pocket Milling Cycles

Chapter Overview

Use pocket milling cycles to cut circular, rectangular, hemispherical pockets and posts, or irregular pockets and posts. Pocket milling cycles are cycles that make multiple passes along the X, Y, and Z axes to cut out a pocket in a workpiece. There are 8 pocket milling cycles. These include:

- five G88.1 Pocket Milling Roughing Cycles
- three G88.2 Pocket Milling Finishing Cycles

Important: You must turn cutter compensation off before executing any of these pocket cycles. An error "Illegal G Code During G41/G42" is displayed if cutter compensation is on when the control executes one of these pocket cycles.

Pocket Milling Roughing Cycle (G88.1)

Use the G88.1 pocket milling roughing cycle to rough out rectangular or circular pockets, slots, and to enlarge an existing rectangular or circular pocket.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane. If the current plane is not the XY plane, the operation of these cycles will rotate accordingly.

Important: Tool length and diameter offsets must be entered and active prior to the G88 block.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.1 roughing cycle for each of the possible pockets.

Rectangular Pocket Roughing Using G88.1

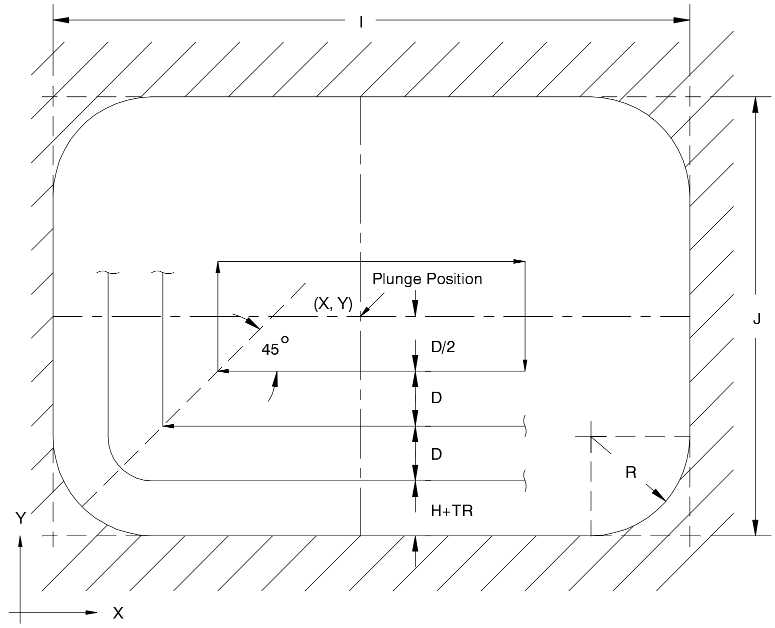
Use the G88.1 pocket milling roughing cycle to rough out a rectangular pocket in a workpiece. This cycle makes multiple rectangular cuts at a programmed width and depth.

The G88.1 block used to rough out a rectangular pocket has this format:

```
G88.1 X__Y__Z__I__J__(,R or,C)__P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the rectangular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the rectangular pocket's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. **Important:** It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively. **Important:** When roughing out a rectangular pocket, the tool diameter can not exceed the length of the shortest side of the rectangular pocket. If it does, the control enters Cycle-Stop mode and displays the error message "TOOL RADIUS TOO LARGE" on the CRT.
,R	Corner radius. This is an optional parameter that is used to program rounded interior corners in the rectangular pocket.
,C	Corner chamfer. This is an optional parameter that is used to program chamfered interior corners in the rectangular pocket. **Important:** In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.
P	Direction of roughing cut. This parameter determines whether the roughing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. **Important:** Cutter compensation (G41/G42) must be disabled prior to the G88.1 block. The control generates an error if compensation is not disabled.
H	The finish allowance that will be left on the sides of the pocket. This finish allowance can be removed later using a G88.2 finishing cycle. To leave a finish allowance on the pocket bottom, program a pocket depth (Z parameter) that is at the desired finish allowance above the actual pocket bottom. This finish allowance can be removed later using a G88.2 finishing cycle.
D	Roughing cut thickness. This parameter specifies the maximum width of any XY axis roughing cuts. This is an optional parameter. If not programmed, the control uses the default thickness, which is equal to half of the current tool diameter. **Important:** The roughing cut thickness can not be greater than the current tool diameter. If it is, the control will enter Cycle-Stop mode and display the error message "D-WORD LARGER THAN TOOL DIAMETER" on the CRT.
L	Incremental plunge depth of each cutting pass along the Z axis. If L is not programmed, the plunge amount will be equal to the programmed depth of the pocket. This is an optional parameter.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed, the roughing feedrate (F) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis moves. If not programmed, the existing (modal) feedrate will be used. **Important:** The rectangular pocket does not have to be parallel to the axes of the selected plane. It may be rotated by rotating the work coordinate system (G68). Refer to chapter 12 for additional information on rotating the work coordinate system.

Figure 21.1
Rectangular Pocket Roughing Using G88.1



Important: The tool should be positioned near the center of the pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the rectangular pocket is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate specified by the F parameter.

After the plunge operation, the control performs a single-axis roughing cut outward along either the -X or -Y axis. The roughing cut is performed along which ever axis intersects the longer side of the rectangular pocket. The control then moves the tool in a rectangular path, defined by the programmed sides of the rectangular pocket, that starts and ends at the same point on either the -X or the -Y axis.

After completing the rectangular path, the control makes another single-axis roughing cut outwards along the -X or -Y axis. The control then moves the tool in a rectangular path that starts and ends at the same point on -X or -Y axis.

This process is repeated until the side of the pocket, less the finish allowance H, is reached. The tool is then simultaneously raised to the clearance amount and moved to the plunge-position. This completes the machining of one L level.

The width of the first roughing cut is equal to the programmed rough cut thickness, D, divided by two then multiplied by 95% ((D/2) x (.95)). The width of the last roughing cut is equal to the tool radius plus the finish allowance (H + TR).

The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

If ,R or ,C is not programmed in the G88.1 block, each corner of the rectangular pocket is squared off as much as the tool radius will allow. If ,R or ,C is programmed in the G88.1 block, the corners of the rectangular pocket will either be rounded or chamfered. Refer to chapter 15, *Using Chamfers and Corner Radius*, for additional information on chamfers and corner rounding.

Rectangular Pocket Enlarging Using G88.1

Use the G88.1 pocket milling roughing cycle to enlarge an existing rectangular pocket in a workpiece. This cycle makes multiple rectangular cuts at a programmed width and depth.

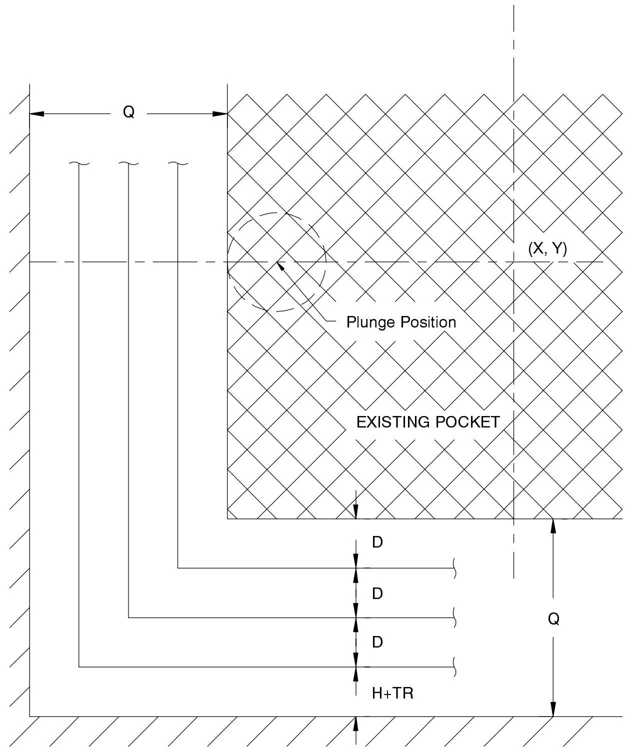
The G88.1 block used to enlarge an existing rectangular pocket has this format:

```
G88.1  X__Y__Z__I__J__Q__(,R or,C)__P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the original rectangular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket.
I J	Length of the original rectangular pocket's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. **Important:** It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
Q	Unsigned incremental value that specifies how much the original rectangular pocket should be enlarged. I and J are both enlarged by this amount.
,R	Corner radius.
,C	Corner chamfer.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Important: The rectangular pocket does not have to be parallel to the axes of the selected plane. It may be rotated by rotating the work coordinate system (G68). Refer to chapter 12 for additional information on rotating the work coordinate system.

Figure 21.2
Rectangular Pocket Enlarging Using G88.1



Important: The tool should be positioned near the center of the original pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to system installers literature) while moving it to the longest side of the original pocket. This move is always along the -X or –Y axis depending on whether the rectangular pocket is longer along the X or Y axis (shorter axis move is the first move). This position is the plunge- position, the position where the plunge to the programmed depth takes place.

Important: If the original pocket has rounded or chamfered corners, excess material may be in the corners that may have to be cleaned out before attempting to enlarge the original pocket, since the G88.1 cycle does not compensate for rounded or chamfered corners.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. The plunge takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate specified by the F parameter.

After the plunge operation, the control performs a single-axis cut outwards to a point on the longest side of the rectangular pocket. This move is always along the -X or -Y axis depending on whether the rectangular pocket is longer along the X or Y axis (shorter axis move is the first move). The control then moves the tool in a rectangular path that starts and ends at either the -X or -Y axis.

This process is repeated until the sides of the enlarged pocket, less the finish allowance H, are reached. The tool is then raised by the clearance amount and moved at rapid feedrate back to the plunge-position. This completes the machining of one L level.

The width of the last roughing cut is equal to the tool radius plus the finish allowance (H + TR). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

If ,R or ,C is not programmed in the G88.1 block, each corner of the rectangular pocket is squared off as much as the tool radius will allow. If ,R or ,C is programmed in the G88.1 block, the corners of the rectangular pocket will either be rounded or chamfered. Refer to chapter 15, *Using Chamfers and Corner Radius*, for additional information on chamfering and corner rounding.

Slot Roughing Using G88.1

Use the G88.1 pocket milling roughing cycle to rough out a slot in a workpiece. This cycle makes multiple cuts at a programmed length and depth.

The G88.1 block used to rough out a slot has this format:

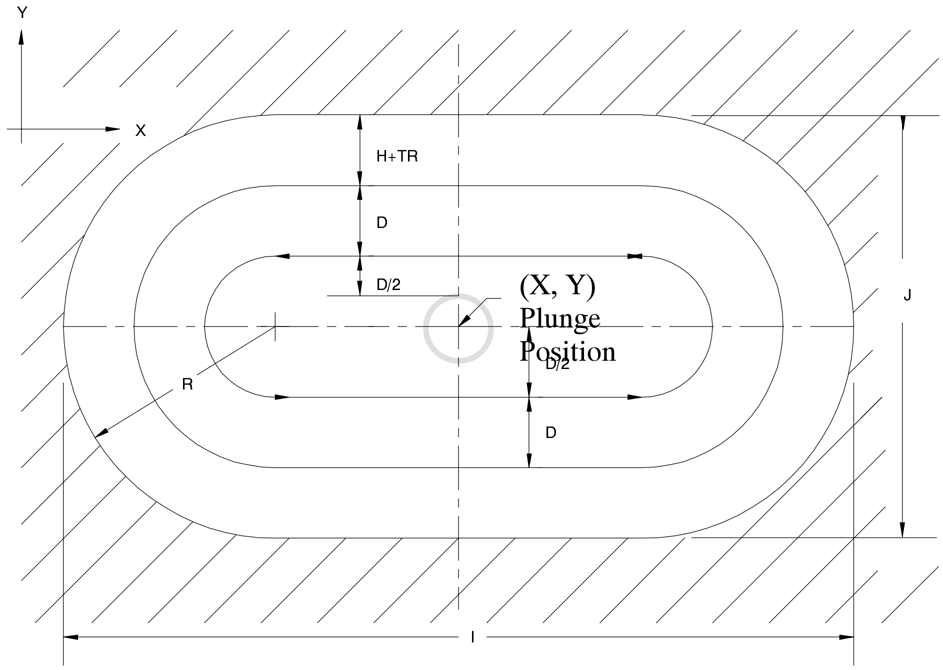
```
G88.1  X__Y__Z__I__R__P__H__D__L__E__F__; (X axis slot)
```

or

```
G88.1  X__Y__Z__J__R__P__H__D__L__E__F__; (Y axis slot)
```

Where :	Is :
X Y	The coordinates that specify the center of the slot.
Z	The coordinate (along the plunging axis) that specifies the bottom of the slot.
I J	The length of the slot as measured from the points where the axis intersects the arc at each end of the slot. I specifies the length of a X axis slot. J specifies the length of a Y axis slot. These are unsigned incremental values. **Important:** It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
R	The radius of the arc at each end of the slot. The radius equals the slot-width/2. If not programmed, the control uses the tool radius value. **Important:** When roughing out a slot, the tool radius can not exceed the radius of the arc at each end of the slot. If it does, the control enters Cycle-Stop mode and displays the error message "TOOL RADIUS TOO LARGE" on the CRT.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 21.3
Slot Roughing Using G88.1



Important: The tool should be positioned at the center of the slot prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

If the programmed R parameter is greater than the tool radius, this cycle is processed similar to a G88.1 roughing cycle for a rectangular pocket. The difference being that the R parameter programmed in a slot roughing cycle specifies the radius of the arc at the end of the slot verses the radius of the corners in a rectangular roughing cycle. The control cuts an arc at the short side of the rectangular pocket based on the programmed R parameter and the arc-center determined by the control.

If the programmed R parameter is equal to the tool radius, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the arc-center at the -X or -Y end of the slot. This simultaneous move takes place at the rapid feedrate. This position is the plunge-position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. The plunge takes place at the plunge feedrate specified by the E parameter. After the plunge operation a roughing cut is made at the feedrate specified by the F parameter to the arc-center at the +X or +Y end of the slot.

A plunge to the next incremental L level or to the programmed Z level is made. A roughing cut is made at the feedrate F to the arc-center at the -X or -Y end of the slot. This process is repeated at each L level until the slot is machined out. When the slot is machined out the control raises the tool to the initial Z level plus the clearance amount and then moves it to the pre-cycle position of the tool.

Circular Pocket Roughing Using G88.1

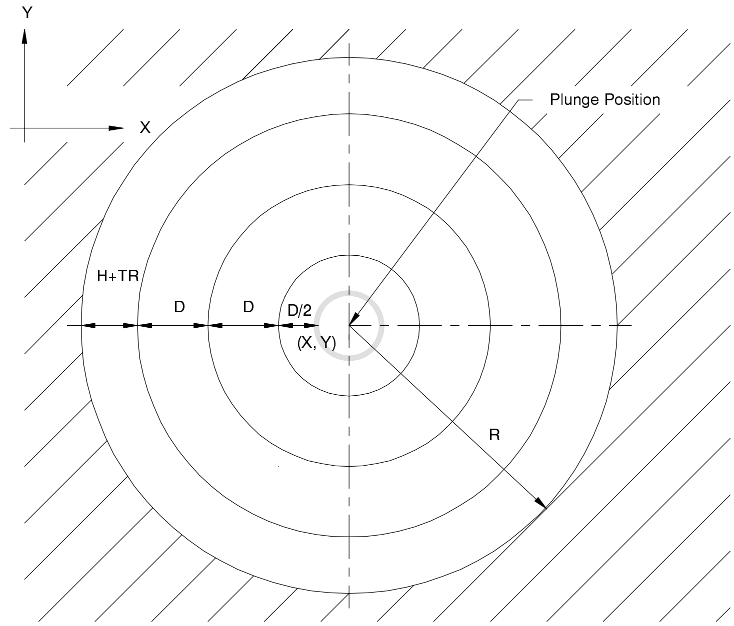
Use the G88.1 pocket milling roughing cycle to rough out a circular pocket in a workpiece. This cycle makes multiple circular cuts at a programmed width and depth.

The G88.1 block used to rough out a circular pocket has this format:

```
G88.1  X__Y__Z__R__P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the circular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the circular pocket. This parameter must be programmed.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 21.4
Circular Pocket Roughing Using G88.1



Important: The tool should be positioned near the center of the pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the circular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the pocket is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate.

After the plunge operation, the control performs a single-axis cut outwards along the -X axis of the circular pocket. The control then moves the tool in a 360 degree circular path, defined by the center of the circular pocket and it's radius, around the plunge- position.

After completing the 360 degree circular path, the control makes a single-axis rough cut outwards along the -X axis then cuts another 360 degree circular path. This process is repeated until the sides of the pocket, less the finish allowance H, are reached. The tool is then simultaneously raised by the clearance amount and moved at rapid feedrate back to the plunge-position. This completes machining of one L level.

The width of the first roughing cut is equal to the programmed rough cut thickness, D, divided by two then multiplied by 95% ((D/2) x (.95)). The width of the last roughing cut is equal to the tool radius plus the finish allowance (H + TR).

The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

Circular Pocket Enlarging Using G88.1

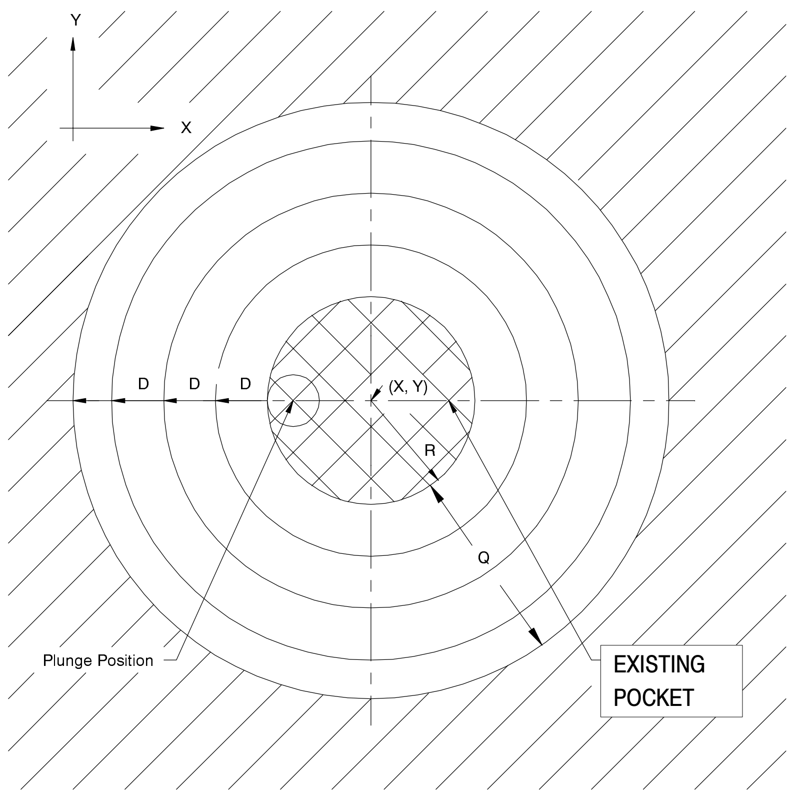
Use the G88.1 pocket milling roughing cycle to enlarge an existing circular pocket in a workpiece. This cycle makes multiple circular cuts at a programmed width and depth.

The G88.1 block used to enlarge an existing circular pocket has this format:

```
G88.1  X__Y__Z__R__Q__P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the original circular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the original circular pocket. This parameter must be programmed.
Q	Unsigned incremental value that specifies how much the original circular pocket should be enlarged. This parameter must be programmed. The radius of the enlarged circular pocket is equal to R + Q.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 21.5
Circular Pocket Enlarging Using G88.1



Important: The tool should be positioned near the center of the pocket prior to the G88.1 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer the literature provided by your system installer) while moving it to the center of the original circular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The tool is then moved along the -X axis to the left side of the original circular pocket. This position is the plunge-position of the cycle.

After the plunge operation, the control performs a single-axis cut outwards along the -X axis of the circular pocket. The control then moves the tool in a 360 degree circular path that starts and ends at the -X axis.

After completing the 360 degree circular path, the control makes a single-axis rough cut outwards along the -X axis then cuts another 360 degree circular path. This process is repeated until the sides of the pocket, less the finish allowance H, are reached. The tool is then simultaneously raised by the clearance amount and moved at rapid feedrate back to the plunge-position. This completes machining of one L level.

The width of the last roughing cut is equal to the tool radius plus the finish allowance (H + TR). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the pocket is machined out.

Once the pocket has been machined out, the control simultaneously raises the tool to the initial Z level plus the clearance amount while moving it away from pocket edge by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate simultaneously along the X, Y, and Z axes to the pre-cycle position of the tool.

Pocket Milling Finishing Cycle (G88.2)

Use the G88.2 pocket milling finishing cycle to remove the finish allowance left on the sides of the rectangular or circular pockets, and slots. Use this cycle to finish a pocket formed by using a G88.1 roughing cycle. Typically a tool change is made between the G88.1 and the G88.2 cycles.

Important: Remember:

- the active plane is selected using G17, G18, or G19
- In this chapter it is assumed that G17, the XY plane, is selected as the active plane.
- tool length and diameter offsets must be entered and active prior to the G88 block
- if the radius of the finishing tool is larger than the radius of the roughing tool, some material may be left in the corners of the pocket after the finishing pass

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.2 finishing cycle for each of the possible pockets.

Rectangular Pocket Finishing Using G88.2

Use the G88.2 pocket milling finishing cycle to finish a rectangular pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a rectangular pocket during a G88.1 roughing cycle.

The G88.2 block used to finish a rectangular pocket has this format:

```
G88.2  X__Y__Z__I__J__(,R or ,C)__P__H__L__F__;
```

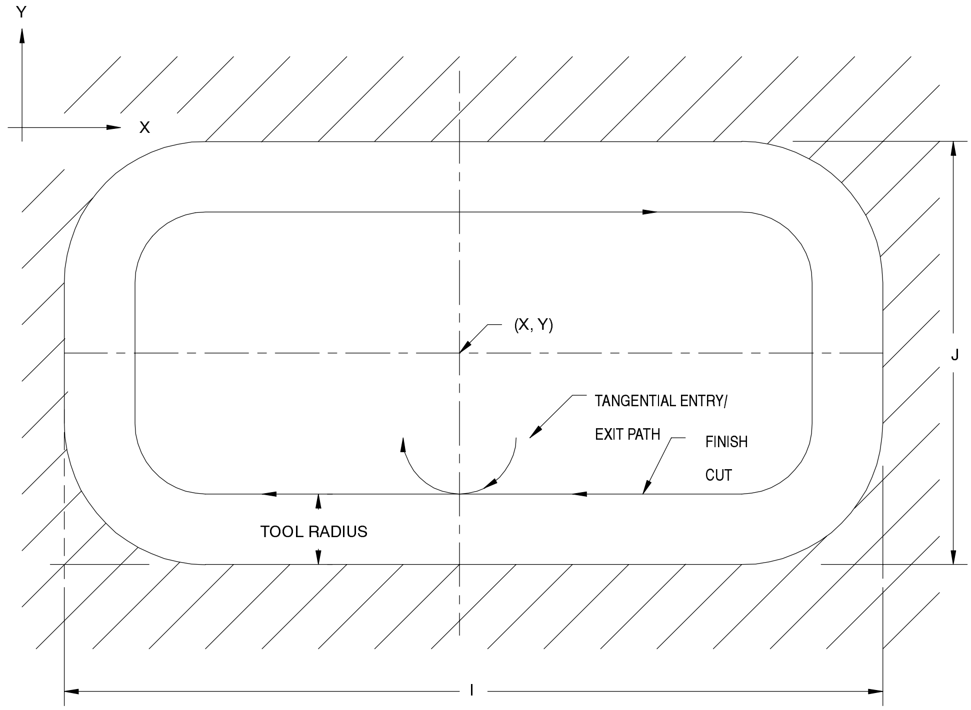
Where :	Is :
X Y	The coordinates that specify the center of the rectangular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the rectangular pocket's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. **Important:** It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
,R	Corner radius. This is an optional parameter that is used to program rounded interior corners in the rectangular pocket.
,C	Corner chamfer. This is an optional parameter that is used to program chamfered interior corners in the rectangular pocket. **Important:** In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.

Where :	Is :
`P`	Direction of finishing cut. This parameter determines whether the finishing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. **Important:** Cutter compensation (G41/G42) must be disabled prior to the G88.2 block. The control generates an error if compensation is not disabled.
`H`	The finish allowance that will be left on the sides of the pocket. This is an optional parameter that is provided to allow for multiple finishing cuts.
`L`	Incremental plunge depth of each cutting pass plunge along the Z axis. If L is programmed, a finish pass is made at each L level. If L is not programmed, only one finishing pass is made at the programmed Z depth. This is an optional parameter. It is typically programmed when a very deep pocket is being finished.
`F`	Finishing feedrate. If not programmed the existing (modal) feedrate will be used.

Important: The rectangular pocket does not have to be parallel to the axes of the selected plane. It may be rotated by rotating the work coordinate system (G68). Refer to chapter 12 for additional information on rotating the work coordinate system.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool diameter by the control. The tangential entry/exit point is along the -X or -Y axis depending on which axis intersects the center of the longer side of the rectangular pocket. If the pocket is square, the tangential entry/exit point is along the -X axis.

Figure 21.6
Rectangular Pocket Finishing Using G88.2



Important: The tool should be positioned near the center of the pocket prior to the G88.2 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular pocket specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

The control starts the finish pass by moving the tool from the pocket center to the start point of the tangential entry/exit path. This start point is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter.

After the plunge operation, the tool is moved along the tangential entry/exit path to the tangential entry/exit point on the side of the pocket. A finish pass that ends at the tangential entry/exit point is then cut around the sides of the rectangular pocket. The control ends the finish pass by moving the tool from the tangential entry/exit point to the end point of the tangential entry/exit path.

If the programmed Z depth of the pocket has not been reached, the tool is simultaneously raised by the clearance amount and moved back to the tangential entry/exit path start point at the rapid feedrate. A plunge to the next level is then made. A finish pass is made at each L level until the pocket bottom is reached.

When all finish passes are completed, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the pocket by the clearance amount. This simultaneous move takes place at the rapid feedrate. The control then retracts the tool at rapid feedrate in all three axes back to the pre-cycle position of the tool.

Circular Pocket Finishing

Use the G88.2 pocket milling finishing cycle to finish a circular pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a circular pocket during a G88.1 cycle.

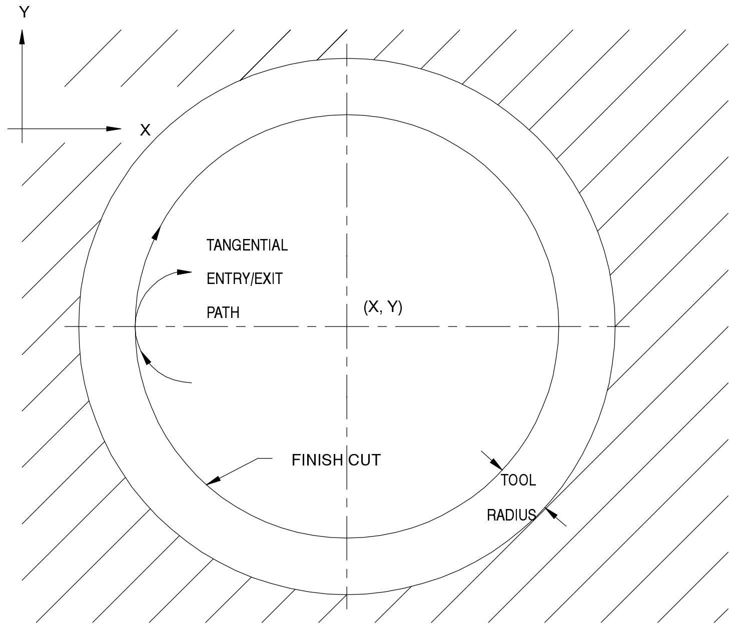
The G88.2 block used to finish a circular pocket has this format:

```
G88.2  X__Y__Z__R__P__H__L__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the circular pocket.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the circular pocket. This parameter must be programmed.
P	Direction of finishing cut.
H	Finish allowance.
L	Incremental plunge depth.
F	Finishing feedrate.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool radius by the control. The tangential entry/exit point will always be at the left side of the circular pocket along the -X axis.

Figure 21.7
Circular Pocket Finishing Using G88.2



Important: The tool should be positioned near the center of the pocket prior to the G88.2 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

Except for the shape of the pocket, the rest of the circular pocket finishing cycle is identical to that of a rectangular pocket finishing cycle.

Slot Finishing Using G88.2

Use the G88.2 pocket milling finishing cycle to finish a slot in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a slot during a G88.1 cycle.

The G88.2 block used to finish a slot has this format:

```
G88.2  X__Y__Z__I__R__P__H__L__F__;  (X axis slot)
```

or

```
G88.2  X__Y__Z__J__R__P__H__L__F__;  (Y axis slot)
```

Where :	Is :
X Y	The coordinates that specify the center of the slot.
Z	The coordinate (along the plunging axis) that specifies the bottom of the slot.
I J	The length of the slot as measured from the point where the axis intersects the arc at each end of the slot. I specifies the length of a X axis slot. J specifies the length of a Y axis slot. These are unsigned incremental values.
R	The radius of the arc at each end of the slot. The radius equals the slot-width/2. If not programmed, the control uses the tool radius value.
P	Direction of finishing cut.
H	Finish allowance.
L	Incremental plunge depth.
F	Finishing feedrate.

Important: The tool should be positioned near the center of the pocket prior to the G88.2 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

If the programmed R parameter is greater than the tool radius, this cycle is processed similar to a G88.2 finishing cycle for a rectangular pocket. The difference being that the R parameter programmed in a slot finishing cycle specifies the radius of the arc at the end of the slot verses the radius of the corners in a rectangular finishing cycle. The control cuts an arc at the short side a the rectangular pocket based on the programmed R parameter and the arc-center determined by the control.

If the programmed R parameter is equal to the tool radius, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it at the rapid feedrate to the arc-center at the -X or -Y end of the slot. This position is the plunge-position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. After the plunge operation a finishing cut is made at the finishing feedrate specified by the F parameter to the opposite arc-center at the +X or +Y end of the slot.

The tool is then raised to the initial Z level plus the clearance amount. If the programmed Z depth of the slot has not been reached, the tool is moved at rapid feedrate back to the plunge-position. A plunge to the next incremental L level or to the programmed Z level is made. Another finishing cut is made at the finishing feedrate to the arc-center at the +X or +Y end of the slot.

This process is repeated at each L level until the sides of the slot are finished. When all finish passes are completed, the control retracts the tool at rapid feedrate in all three axes back to the pre-cycle position of the tool.

END OF CHAPTER

Using Post Milling Cycles

Chapter Overview

This chapter describes how to use G88.3 and G88.4 to program post milling cycles. Use this table to find the information:

Post Milling Roughing Cycle (G88.3)

Use the G88.3 post milling roughing cycle to rough out material outside of a specified area or post.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length, work coordinates, and diameter offsets must be entered and active prior to the G88 block. The radius/diameter of the tool can not exceed the length of the shortest side of the rectangular pocket. If it does, the control enters Cycle-Stop mode and displays the error message “TOOL RADIUS TOO LARGE” on the CRT.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey
- Work coordinate offset changes through the offset softkeys

The following subsections cover using the G88.3 roughing cycle for rectangular or circular posts.

Rectangular Post Roughing Using (G88.3)

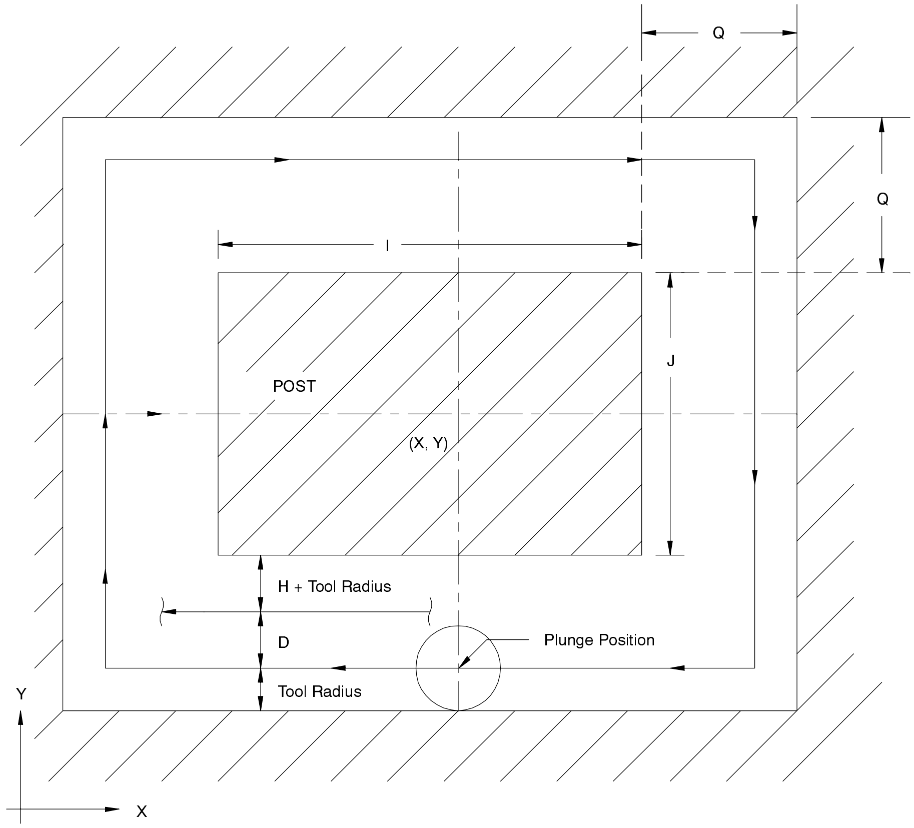
Use the G88.3 post milling roughing cycle to rough out a rectangular post in a workpiece. This cycle makes multiple cuts at a programmed width and depth.

The G88.3 block used to rough out a rectangular post has this format:

```
G88.3  X__Y__Z__I__J__Q__(,R or,C)__P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the rectangular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the post's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. **Important:** It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
Q	Unsigned incremental value that specifies the distance from the sides of the post to the outer perimeter of the pocket. This distance is the same whether measured along the X or the Y axis.
,R	Corner radius. This is an optional parameter that is used to program rounded interior corners in the rectangular pocket.
,C	Corner chamfer. This is an optional parameter that is used to program chamfered interior corners in the rectangular pocket. **Important:** In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.
P	Direction of roughing cut. This parameter determines whether the roughing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. **Important:** If cutter compensation (G41/G42) was enabled prior to the G88.3 block, it is disabled when G88.3 is enabled.
H	The finish allowance that will be left on the sides of the post. This finish allowance can be removed later using a G88.4 finishing cycle. To leave a finish allowance on the pocket bottom, program a pocket depth (Z parameter) that is at the desired finish allowance above the actual pocket bottom. This finish allowance can be removed later using a G88.4 finishing cycle.
D	Roughing cut thickness. This parameter specifies the maximum width of any XY axis roughing cuts. This is an optional parameter. If not programmed, the control uses the default thickness, which is equal to half of the current tool diameter. **Important:** The roughing cut thickness can not be greater than the diameter of the current tool. If it is, the control enters Cycle-Stop mode and displays the error message “D-WORD LARGER THAN TOOL DIAMETER” on the CRT.
L	Incremental plunge depth of each cutting pass along the Z axis. This is an optional parameter. If not programmed, the plunge amount will be equal to the programmed depth of the pocket.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis moves. If not programmed the existing (modal) feedrate will be used.

Figure 22.1
Rectangular Post Roughing Using G88.3



Important: The tool should be positioned near the center of the post prior to the G88.3 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

Depending on which axis intersects the longer side of the pocket, the tool is moved along either the -X or the -Y axis to the outer perimeter of the pocket to be machined around the post. This position is the plunge- position of the cycle, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate.

After the plunge operation, the control moves the tool in a rectangular path, defined by the programmed sides of the rectangular post, that starts and ends at the same point on either the -X or the -Y axis. After completing a rectangular path, the control makes a single-axis rough cut along the -X or -Y axis towards the center of the rectangular post. Another rectangular path is cut that ends at the -X or -Y axis. This process is repeated until the sides of the post, less the finish allowance H, are reached.

The width of the first roughing cut is equal to the tool radius. The width of the last roughing cut is equal to the tool radius plus the finish allowance
(H + TR). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

The tool is then simultaneously raised by the clearance amount and moved along either the -X or -Y axis at rapid feedrate back to the plunge-position. This completes machining of one L level.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

Once the post has been machined out, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the post by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate along the X, Y, and Z axes to the pre-cycle position of the tool.

If ,R or ,C is not programmed in the G88.3 block, each corner of the rectangular post is squared off as much as the tool radius will allow. If ,R or ,C is programmed in the G88.3 block, the corners of the post will either be rounded or chamfered. Refer to chapter 15, Using Chamfers and Corner Radius, for additional information on chamfering and corner rounding.

Circular Post Roughing Using G88.3

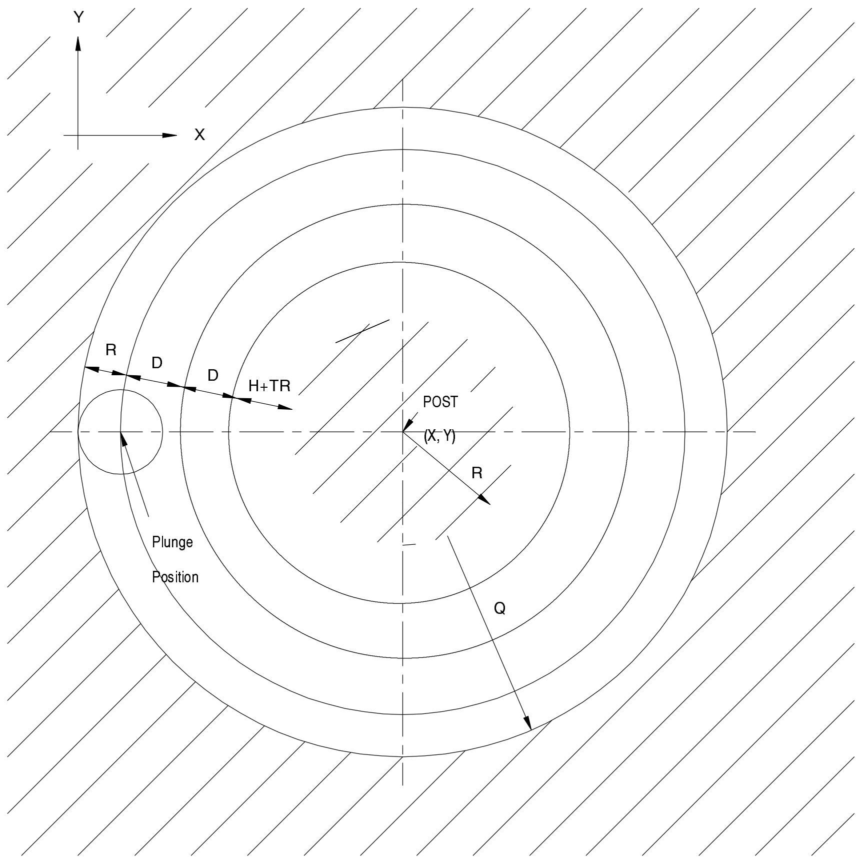
Use the G88.3 post milling roughing cycle to rough out a circular post in a workpiece. This cycle makes multiple circular cuts at a programmed width and depth.

The G88.3 block used to rough out a circular post has this format:

```
G88.3  X__Y__Z__R__Q__P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the circular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
R	The radius of the circular post. This parameter must be programmed.
Q	Unsigned incremental value that specifies the distance from the sides of the circular post to the outer perimeter of the circular pocket.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Incremental plunge depth.
E	Plunge feedrate.
F	Roughing feedrate.

Figure 22.2
Circular Post Roughing Using G88.3



Important: The tool should be positioned near the center of the post prior to the G88.3 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

The tool is then moved along the -X axis to the outer perimeter of the pocket. This position is the plunge- position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter. This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate.

After the plunge operation, the control moves the tool in a circular path that starts and ends at the same point on the -X axis. After completing a circular path, the control makes a single-axis rough cut towards the post along the -X axis. Another circular path is cut that start and ends at the -X axis. This process is repeated until the sides of the pocket, less the finish allowance H, are reached.

The width of the first roughing cut is equal to the tool radius. The width of the last roughing cut is equal to the tool radius plus the finish allowance (H + TR). The width of the remaining roughing cuts is calculated by the control based on the remaining area to be roughed-out and the programmed rough cut thickness, D. The control divides the remaining area by D to calculate the number of roughing cuts needed to rough out this area. The control then adjusts the width and number of these cuts until an even number of roughing cuts is achieved. The width of these cuts will always be equal to or less than the programmed rough cut thickness, D.

The tool is then simultaneously raised by the clearance amount and moved at rapid feedrate along the -X axis back to the plunge-position. This completes the machining of one L level.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

Once the post has been machined out, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the post by the clearance amount. This simultaneous move takes place at the rapid feedrate. The tool is then moved at rapid feedrate along the X, Y, and Z axes to the pre-cycle position of the tool.

Pocket Milling Finishing Cycle (G88.4)

Use the G88.4 post milling finishing cycle to remove the finish allowance left on the sides of a rectangular or circular post. You can use this cycle to finish a post formed by using a G88.3 roughing cycle. Typically a tool change is made between the G88.3 and the G88.4 cycles.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length, work coordinates, and diameter offsets must be entered and active prior to the G88 block. The radius/diameter of the tool can not exceed the length of the shortest side of the pocket. If it does, the control enters Cycle-Stop mode and displays the error message "TOOL RADIUS TOO LARGE."

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey
- Work coordinate offset changes through the offset softkeys

The following subsections cover using the G88.4 finishing cycle for rectangular or circular posts.

Rectangular Post Finishing Using G88.4

Use the G88.4 post milling finishing cycle to finish a rectangular post in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a rectangular post during a G88.3 cycle.

The G88.4 block used to finish a rectangular post has this format:

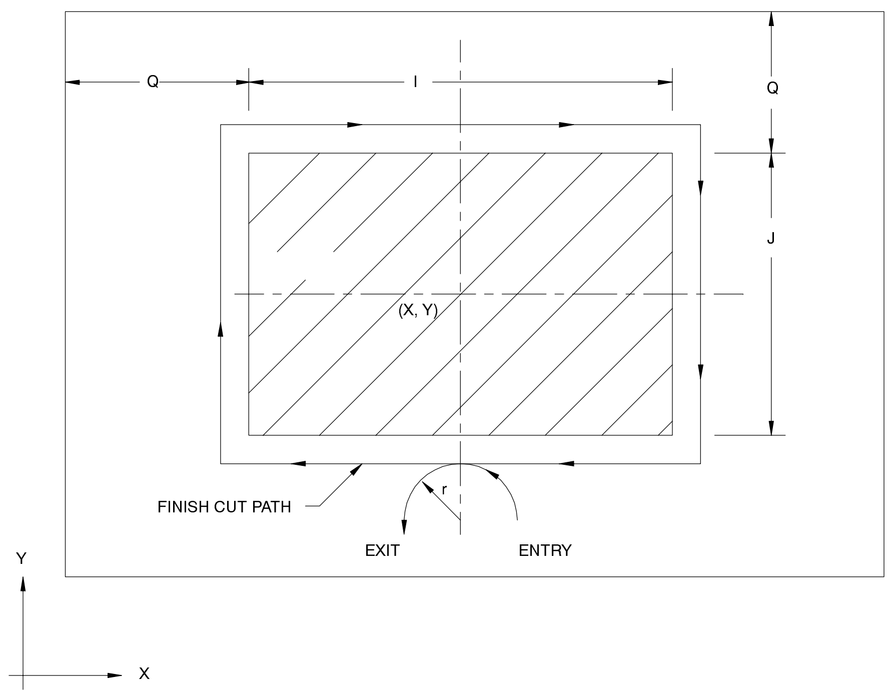
```
G88.4  X__Y__Z__I__J__Q__(,R or ,C)__P__H__L__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the rectangular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the rectangular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
I J	The length of the post's sides. I specifies the length of the side parallel to the X axis. J specifies the length of the side parallel to the Y axis. These are unsigned incremental values. **Important:** It is assumed that I and J are assigned in AMP as the integrand axis names that correspond to the X and Y axes respectively.
Q	Unsigned incremental value that specifies the distance from the sides of the post to the outer perimeter of the pocket. This distance is the same whether measured along the X or the Y axis.
,R	Corner radius. This is an optional parameter that is used to program rounded post corners.
,C	Corner chamfer. This is an optional parameter that is used to program chamfered rounded post corners. **Important:** In order to program rounded or chamfered corners the Chamfering and Corner Radius option must be installed in the control.
P	Direction of finishing cut. This parameter determines whether the finishing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. **Important:** If cutter compensation (G41/G42) was enabled prior to the G88.4 block, it is disabled when G88.4 is enabled.

Where :	Is :
H	The finish allowance that will be left on the sides of the post. This is an optional parameter that is provided to allow for multiple finishing cuts.
L	Incremental plunge depth of each cutting pass along the Z axis. If L is programmed, a finish pass is made at each L level. If L is not programmed, only one finishing pass is made at the programmed Z depth. This is an optional parameter. It is typically programmed when a very deep pocket is being finished.
F	Finishing feedrate. If not programmed the existing (modal) feedrate will be used.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool radius by the control. The tangential entry/exit point is along the -X or -Y axis depending on which axis intersects the center of the longer side of the rectangular post. If the rectangular post is square, the tangential entry/exit point will be along the -X axis.

Figure 22.3
Rectangular Post Finishing Using G88.4



Important: The tool should be positioned near the center of the post prior to the G88.4 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The pre-cycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the rectangular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

The control starts the finish pass by moving the tool from the post center to the start point of the tangential entry/exit path. This start point is the plunge position, the position where the plunge to the programmed depth takes place.

If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter. If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter.

After the plunge operation, the tool is moved along the tangential entry/exit path to the tangential entry/exit point on the side of the post. A finish pass that ends at the tangential entry/exit point is then cut around the sides of the rectangular post. The control ends the finish pass by moving the tool from the tangential entry/exit point to the end point of the tangential entry/exit path.

If the programmed Z depth of the post has not been reached, the tool is simultaneously raised by the clearance amount and moved back to the tangential entry/exit path start point at the rapid feedrate. A plunge to the next level is then made. A finish pass is made at each L level until the pocket bottom is reached.

When all finish passes are completed, the control simultaneously raises the tool to the initial Z level while moving it away from the side of the post by the clearance amount. This simultaneous move takes place at the rapid feedrate. The control then retracts the tool at rapid feedrate in all three axes back to the pre-cycle position of the tool.

Circular Post Finishing Using G88.4

Use the G88.4 post milling finishing cycle to finish a circular post in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of a circular post during a G88.3 cycle.

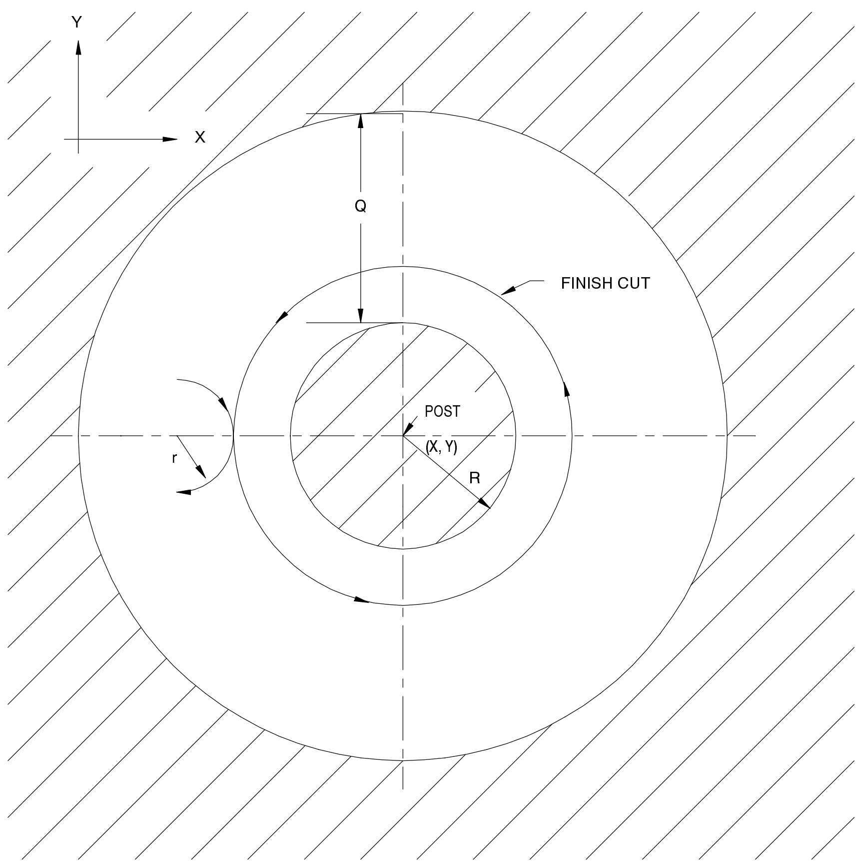
The G88.4 block used to finish a circular post has this format:

```
G88.4  X__Y__Z__Q__R__P__H__L__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the circular post.
Z	The coordinate (along the plunging axis) that specifies the bottom of the circular pocket.
Q	Unsigned incremental value that specifies the distance from the sides of the circular post to the outer perimeter of the circular pocket.
R	The radius of the circular post. This parameter must be programmed.
P	Direction of finishing cut.
H	Finish allowance.
L	Incremental plunge depth.
F	Finishing feedrate.

In a finishing cycle, a smooth entry to and exit from the finish contour is accomplished by having the tool approach and leave the finish contour along a tangential arc. The radius of this arc is set equal to the tool radius by the control. The tangential entry and exit will always occur on the left side of the circular post at the -X axis.

Figure 22.4
Circular Post Finishing Using G88.4



Important: The tool should be positioned near the center of the post prior to the G88.4 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool. The precycle position must be at some depth other than the cycles programmed final depth or an error is generated.

From the precycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the circular post specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate.

Except for the shape of the post, the rest of the circular post finishing cycle is identical to that of a rectangular post finishing cycle.

END OF CHAPTER

Using Hemisphere Milling Cycles

Chapter Overview

This chapter describes how to use G88.5 and G88.6 to program hemisphere milling cycles. Use this table to find information:

Hemisphere Milling Roughing Cycle (G88.5)

Use the G88.5 hemisphere milling roughing cycle to rough out concave or convex hemispherical pockets.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length and diameter offsets must be entered and active prior to the G88 block.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.5 roughing cycle for concave or convex hemispheres.

Concave Hemisphere Roughing Using G88.5

Use the G88.5 concave milling roughing cycle to rough out a concave pocket in a workpiece. This cycle makes multiple concentric circular cuts at a programmed width and depth.

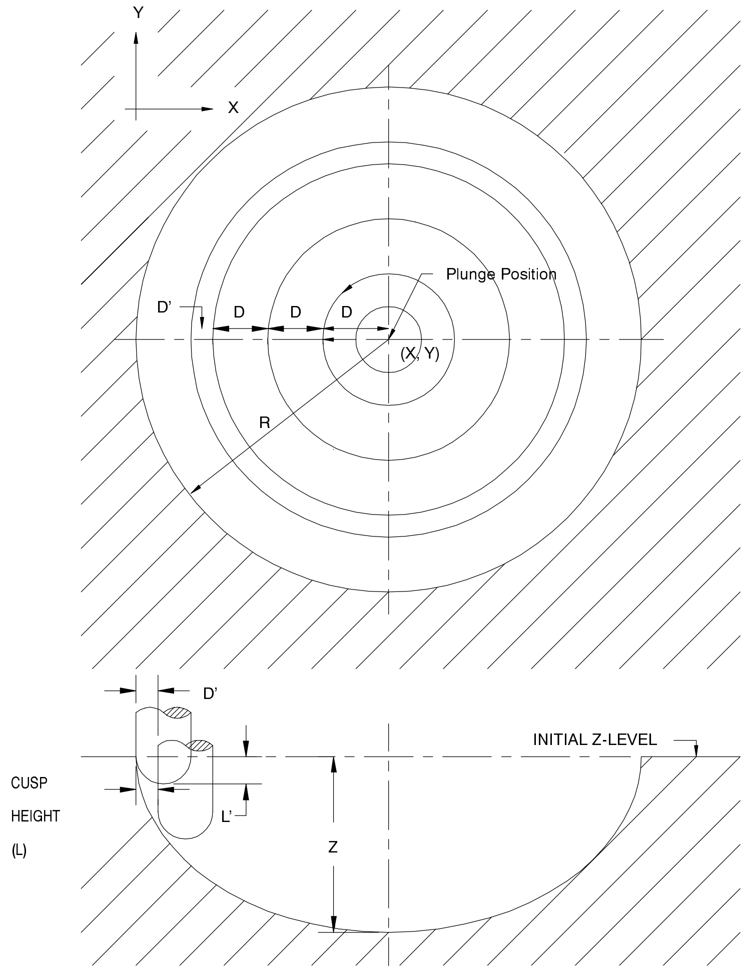
The G88.5 block used to rough out a concave pocket has this format:

```
G88.5  X__Y__Z__R__Q0_P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the concave hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the bottom of the hemispherical pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
R	Radius of the concave hemisphere.
Q0	Code specifying a concave hemisphere.
P	Direction of roughing cut. This parameter determines whether the roughing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. **Important:** If cutter compensation (G41/G42) was enabled prior to the G88.5 block, it is disabled when G88.5 is enabled.
H	The finish allowance that will be left on the sides of the concave hemisphere. This finish allowance can be removed later using a G88.6 finishing cycle.
D	Roughing cut thickness. This parameter specifies the maximum width of any XY axis roughing cuts. This is an optional parameter. If not programmed, the control uses the default thickness, which is equal to half of the current tool diameter. **Important:** The roughing cut thickness can not be greater than the diameter of the current tool. If it is, the control enters Cycle-Stop mode and displays the error message “D-WORD LARGER THAN TOOL DIAMETER” on the CRT.
L	Roughing cusp height. This value of this parameter is a measurement that corresponds to the height of the material left along the sides of the hemisphere after each X-Z roughing plunge.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis moves. If not programmed the existing (modal) feedrate will be used.

If Q0 is programmed, the control generates a concave hemisphere. The control makes multiple concentric circular cuts at each level of the concave hemisphere. The depth of the pocket increases and the diameter of each level decreases until the bottom center of the concave hemisphere is reached.

Figure 23.1
Concave Hemisphere Roughing Using G88.5



Important: The tool should be positioned near the center of the concave hemisphere prior to the G88.5 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the concave hemisphere specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the concave hemisphere is the plunge position, the position where the plunge to the programmed depth takes place.

Prior to each plunge, the control computes a delta rough cut thickness, D', and a delta plunge depth, L'. These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.5 block, and the tool radius programmed prior to the G88.5 block.

With the axis positioned at the plunge-position, a plunge along the Z axis, of depth L' is performed. This plunge takes place at the plunge feedrate specified by the E parameter.

After the plunge operation, the control performs a single-axis rough cut outwards towards the left on the -X axis of the hemisphere. The width of the first roughing cut is equal to the programmed rough cut thickness, D, divided by two then multiplied by 95% ((D/2) x (.95)). The control then moves the tool in a 360 degree circular path around the plunge-position.

After completing a 360 degree circular path, the control makes another single-axis rough cut outwards along the -X axis by the rough cut thickness, D. Another 360 degree circular path is cut that ends at the -X axis. This process is repeated until the side of concave hemisphere, less the finish allowance H, is reached.

Important: The thickness of the last rough cut is D' not the programmed D parameter.

The tool is then raised by the clearance amount (AMP selectable, refer to system installers literature) and moved at rapid feedrate back to the plunge-position. This completes machining of one L level.

If the programmed Z depth of the pocket has not been reached, another plunge takes place along the Z axis to the next L level. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position.

Convex Hemisphere Roughing Using G88.5

Use the G88.5 convex milling roughing cycle to rough out a convex pocket in a workpiece. This cycle makes multiple concentric circular cuts at a programmed width and depth from the top center of the convex hemisphere to the outermost diameter of the convex hemisphere.

The G88.5 block used to rough out a convex pocket has this format:

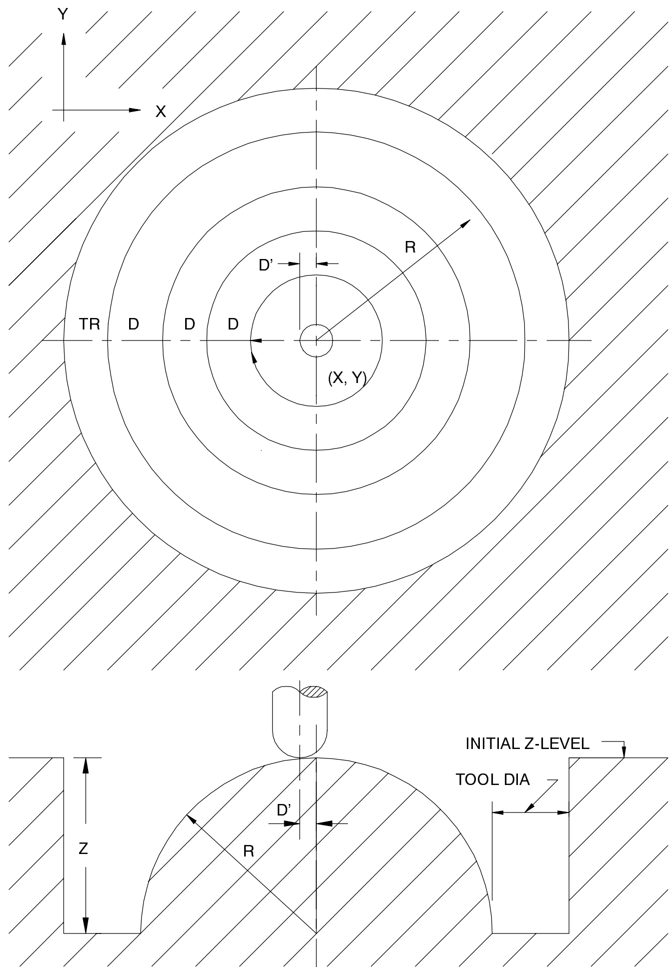
```
G88.5  X__Y__Z__R__Q1_P__H__D__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the convex hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the base of the convex hemisphere.
R	Radius of the convex hemisphere.
Q1	Code specifying a convex hemisphere.
P	Direction of roughing cut.
H	Finish allowance.
D	Roughing cut thickness.
L	Roughing cusp height.
E	Plunge feedrate.
F	Roughing feedrate.

If Q1 is programmed, the control generates a convex hemisphere. The control makes multiple concentric circular cuts at each level of the convex hemisphere. The depth of the pocket and the diameter of each level increases until the outer diameter of the convex hemisphere is reached.

Important: The tool should be positioned near the center of the convex hemisphere prior to the G88.5 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

Figure 23.2
Convex Hemisphere Roughing Using G88.5



From the pre-cycle position, the control simultaneously raises the tool by the clearance amount (AMP selectable, refer to the literature provided by your system installer) while moving it to the center of the convex hemisphere specified by the X and Y parameters. This simultaneous move takes place at the rapid feedrate. The center of the convex hemisphere is the plunge position, the position where the plunge to the programmed depth takes place.

Prior to each plunge, the control computes a delta rough cut thickness, D', and a delta plunge depth, L'. These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.5 block, and the tool radius programmed prior to the G88.5 block.

With a convex hemisphere, the plunge is actually a contour move to the outward along the -X axis. This move cuts along the spherical contour, axes X and Z, at the plunge feedrate specified by the E parameter. This plunge simultaneously moves the X and Z axes by the D' and L' amounts.

After the plunge, the control moves the tool in a 360 degree circular path around the plunge-position. After completing this circular path, the control makes a rough cut, D parameter, outwards along the -X axis. Another 360 degree circular path is cut that ends at the -X axis. This process is repeated until the outer diameter of the pocket, plus the finish allowance H, is reached. The tool is then raised by the clearance amount and moved at rapid feedrate back to the initial cut at the current level. This completes machining of one L level.

Important: The outer diameter of the pocket is equal to the radius of the hemisphere plus the tool diameter and the finish allowance.

If the programmed Z depth of the pocket has not been reached, another plunge takes place simultaneously along the X and Z axes to the next L' level. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth of the convex hemisphere is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position.

Hemisphere Milling Finishing Cycle (G88.6)

Use the G88.6 hemisphere milling finishing cycle to remove the finish allowance left on the sides of a concave or convex hemisphere. You can use this cycle to finish a hemisphere formed by using a G88.5 roughing cycle. Typically a tool change is made between the G88.5 and the G88.6 cycles.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length and diameter offsets must be entered and active prior to the G88 block.

These features are prohibited during execution of pocket milling cycles:

- MDI mode
- Tool offset changes through the offset softkey

The following subsections cover using the G88.6 finishing cycle for concave or convex hemispheres.

Concave Hemispher Finishing Using G88.6

Use the G88.6 concave milling finishing cycle to finish a concave pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of the concave hemisphere during the G88.5 roughing cycle.

The G88.6 block used to finish a concave pocket has this format:

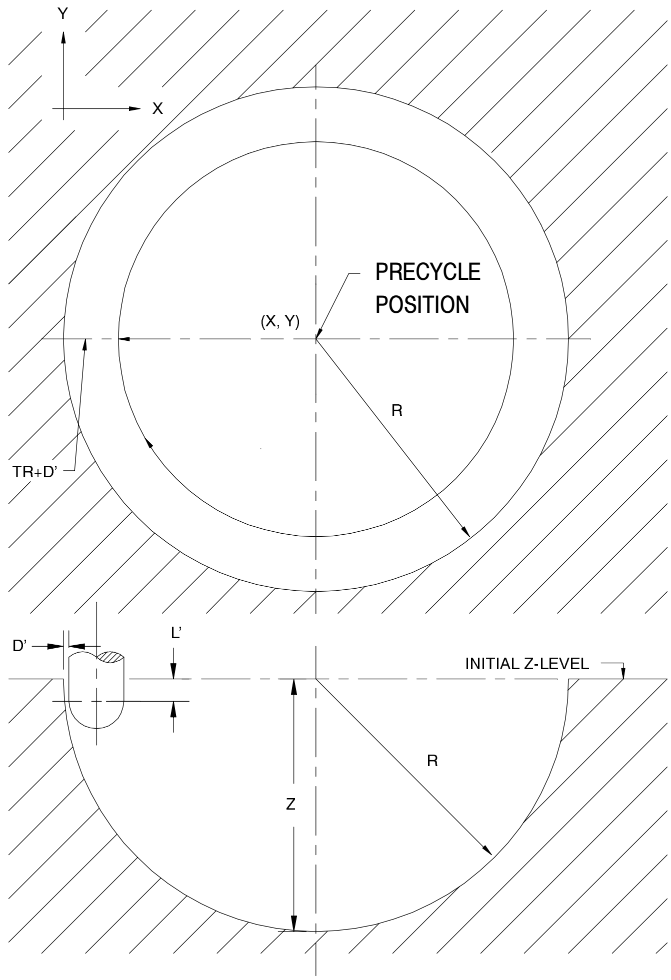
```
G88.6  X__Y__Z__R__Q0_P__H__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the concave hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the bottom of the hemispherical pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start level to the pocket bottom. This parameter must be programmed.
R	Radius of the concave hemisphere.
Q0	Code specifying a concave hemisphere.
P	Direction of finishing cut. This parameter determines whether the finishing cuts are performed in a clockwise or counter-clockwise motion. P0 specifies clockwise. P1 specifies counter-clockwise. **Important:** If cutter compensation (G41/G42) was enabled prior to the G88.6 block, it is disabled when G88.6 is enabled.
H	The finish allowance that will be left on the sides of the concave hemisphere. This is an optional parameter that is provided to allow for multiple finishing cuts.
L	Finishing cusp height. This value of this parameter is a measurement that corresponds to the height of the material left along the sides of the hemisphere after each X-Z roughing plunge.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F) will be used.
F	Finishing feedrate. If not programmed the existing (modal) feedrate will be used.

If Q0 is programmed, the control makes a finish pass around the sides of a concave hemisphere. The control makes multiple circular finish cuts at each level of the concave hemisphere. The depth of the pocket increases and the diameter of each level decreases until the bottom center of the concave hemisphere is reached.

Prior to each finish plunge, the control computes a delta finish cut thickness, D', and a delta plunge depth, L'. These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.6 block, and the tool radius programmed prior to the G88.6 block.

Figure 23.3
Concave Hemisphere Finishing Using G88.6



Important: The tool should be positioned near the center of the concave hemisphere prior to the G88.6 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

The control raises the tool by the clearance amount (AMP selectable, refer to system installers literature) and moves it at the rapid feedrate to the center of the concave hemisphere, X and Y coordinates.

The control initially moves the tool down L' from the initial Z level and offsets it from the side of the concave hemisphere by D'. A 360 degree circular path is cut, defined by the center of the concave hemisphere and the radius of the hemisphere, that ends at the -X axis.

If the programmed Z depth of the pocket has not been reached, another plunge takes place simultaneously along the X and Z axes to the next L' level. Another 360 degree circular path is cut. This process is repeated until the programmed Z depth of the concave hemisphere is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position. This completes the finishing cycle.

Convex Hemisphere Finishing Using G88.6

Use the G88.6 convex milling finishing cycle to finish a convex pocket in a workpiece. This cycle is typically used to remove the finish allowance that was left on the sides of the convex hemisphere during the G88.5 roughing cycle.

The G88.6 block used to finish a convex pocket has this format:

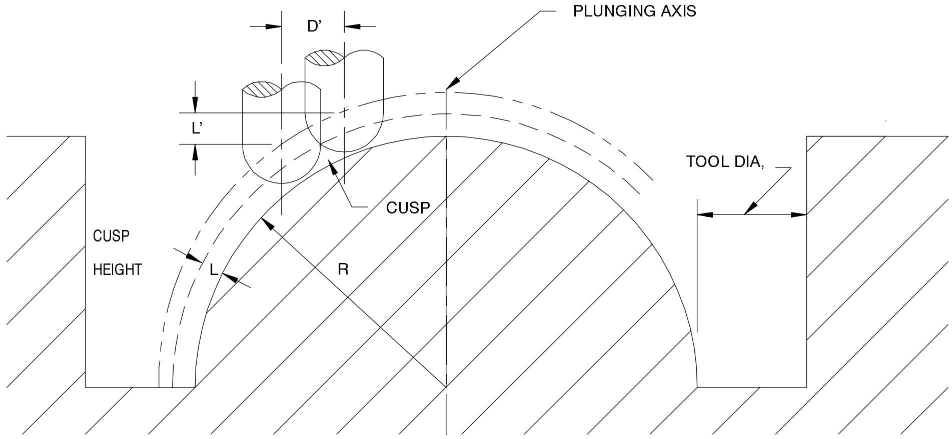
```
G88.6  X__Y__Z__R__Q1_P__H__L__E__F__;
```

Where :	Is :
X Y	The coordinates that specify the center of the convex hemisphere in the selected plane.
Z	The coordinate (along the plunging axis) that specifies the bottom of the hemispherical pocket.
R	Radius of the convex hemisphere.
Q1	Code specifying a convex hemisphere.
P	Direction of finishing cut.
H	Finish allowance.
L	Finishing cusp height.
E	Plunge feedrate.
F	Finishing feedrate.

If Q1 is programmed, the control makes a finish pass around the sides of a convex hemisphere. The control makes multiple circular finish cuts at each level of the convex hemisphere. The depth of the pocket and the diameter of each level increases until the outer diameter of the convex hemisphere is reached.

Prior to each finish plunge, the control computes a delta finish cut thickness, D', and a delta plunge depth, L'. These computations are based on the cusp height (L parameter) and the hemisphere radius (R parameter) programmed in the G88.6 block, and the tool radius programmed prior to the G88.6 block.

Figure 23.4
Convex Hemisphere Finishing Using G88.6



Important: The tool should be positioned near the center of the convex hemisphere prior to the G88.6 block. The Z coordinate of this position determines the initial Z level or top of the pocket. This is the pre-cycle position of the tool.

The control raises the tool by the clearance amount (AMP selectable, refer to system installers literature) and moves it at the rapid feedrate to the center of the convex hemisphere, X and Y coordinates.

With a convex hemisphere, the plunge is actually a contour move to the left along the -X axis. This move cuts along the spherical contour, axes X and Z, at the plunge feedrate specified by the E parameter. This plunge simultaneously moves the X and Z axes by the D' and L' amounts.

After the plunge, the control moves the tool in a 360 degree circular path, defined by the center of the convex hemisphere and the radius of the hemisphere, around the plunge-position.

Important: The thickness of the first rough cut is D' not the programmed D parameter.

If the programmed Z depth of the pocket has not been reached, another plunge takes place simultaneously along the X and Z axes to the next L' level. This plunge simultaneously moves the X and Z axes by the D' and L' amounts. This level is then finished as described in the previous paragraphs. This process is repeated until the programmed Z depth of the convex pocket is reached, at which time the tool is moved at rapid feedrate along the Z axis back to the initial Z level. The tool is then moved at rapid feedrate along the X and Y axes to its pre-cycle position. This completes the finishing cycle.

Important: The radius of the pocket surrounding the convex hemisphere is equal to the radius of the hemisphere plus the tool radius and the finish allowance.

END OF CHAPTER

Irregular Pocket Milling Cycles

Chapter Overview

This chapter describes how to use G89.1 and G89.2 to program irregular pocket milling cycles. Use this table to find information:

Irregular Pocket Milling

Use the G89.1 irregular pocket milling roughing cycle to rough out irregular pockets. Irregular pockets are pockets that are not rectangular, circular, or hemispherical. It is possible to form an irregular post inside an irregular pocket by combining two or more irregular pocket cycles.

You can use the irregular pocket milling finishing cycle (G89.2) to finish an irregular pocket in a workpiece. This cycle is typically used to finish an irregular pocket formed using a G89.1 irregular pocket roughing cycle. A tool change may be performed between the G89.1 and G89.2 cycles.

Important: The active plane is selected using G17, G18, or G19. In this chapter it is assumed that G17, the XY plane, is selected as the active plane.

Important: Tool length and diameter offsets must be entered and active prior to the G89 block.

These features are prohibited during execution of irregular pocket milling cycles:

- MDI mode
- Tool offset changes

The following subsection covers using the G89.1 roughing cycle for irregular pockets.

Irregular Pocket Roughing (G89.1)

Use the irregular pocket milling roughing cycle (G89.1) to rough out an irregular pocket in a workpiece. This cycle makes multiple cuts at a programmed depth, one cutter radius in width.

Important: You may only execute an irregular pocket roughing cycle (G89.1) in a main program.

The G89.1 block used to rough out an irregular pocket has this format:

```
G89.1  X__Y__Z__P__Q__H__E__F__L__;
```

Where :	Is :
X Y	The coordinates that specify the start/end corner of the irregular pocket in the selected plane. These parameters must be programmed.
Z	The coordinate (along the plunging axis) that specifies the bottom of the irregular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start-point to the pocket bottom. This parameter must be programmed.
P	The sequence number of the first block in the set of blocks that define the pocket shape. The coordinates in this block specify the end-point of the pocket's first wall.
Q	The sequence number of the last block in the set of blocks that define the pocket shape. The coordinates in this block specify the end-point of the pocket's last wall. This endpoint must correspond to the start/end corner specified by the X and Y coordinates. This parameter must be programmed. **Important:** The set of blocks specified by P and Q must be at least 5 blocks long to qualify as an irregular pocket. The set of blocks specified by P and Q may be located anywhere in the same program as the G89.1 block. These blocks can not be called from a subprogram or a macro unless the G89.1 block is in that subprogram or macro.
H	The finish allowance that will be left on the sides of the irregular pocket. This finish allowance can be removed later using a G89.2 finishing cycle. To leave a finish allowance on the pocket bottom, program a pocket depth (Z parameter) that is at the desired finish allowance above the actual pocket bottom. This finish allowance can be removed later using a G89.1 roughing cycle programmed with the actual pocket bottom.
E	Plunge feedrate. This parameter determines the feedrate of any Z axis moves. If not programmed the roughing feedrate (F parameter) will be used.
F	Roughing feedrate. This parameter determines the feedrate of any XY axis cutting moves. If not programmed the existing (modal) cutting feedrate will be used.
L	Incremental plunge depth of each cutting pass along the Z axis. If not programmed, the plunge amount will be equal to the programmed depth of the pocket.

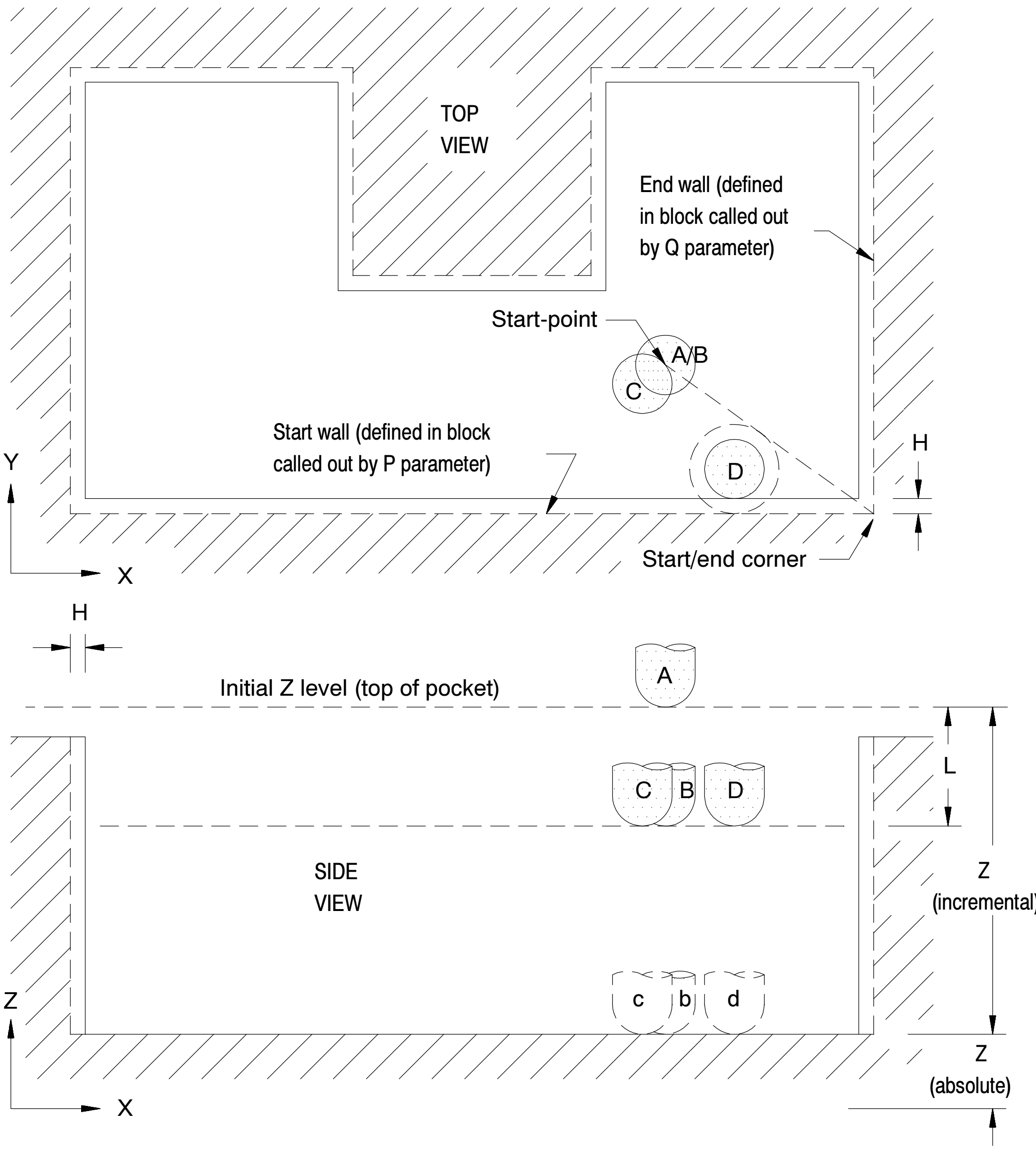
Before invoking the G89.1 cycle, the programmer must activate cutter compensation left or right by programming G41 or G42. This allows the control to begin interpreting the blocks that define the contour of the pocket as they are encountered.

Prior to the G89.1 block, the tool should be positioned near the start/end corner of the pocket and should be just above but not touching the part. This position is referred to as the start-point of the cycle (A in figure 16.16). From the start-point the cutter must be able to move down into the part and then directly over to the start/end corner of the pocket without cutting into any wall of the pocket. The Z coordinate of this position determines the initial Z level or top of the pocket.

Once the axes are positioned at the start-point, a plunge along the Z axis takes place. If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter (B in Figure 24.1). If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter (B in Figure 24.1). This move takes place at the plunge feedrate specified by the E parameter. If E is not programmed, the plunge takes place at the roughing feedrate specified by the F parameter.

After the plunge, the control uses the active cutter compensation mode (G41/G42) to offset the cutter one cutter radius perpendicular to the line from the start-point to the start/end corner (C in Figure 24.1). The control then moves the cutter from this offset position to a point where it is located one cutter radius plus the finish allowance away from both the first wall of the pocket and the line from the start-point to the start/end corner (D in Figure 24.1). This move takes place at the roughing feedrate.

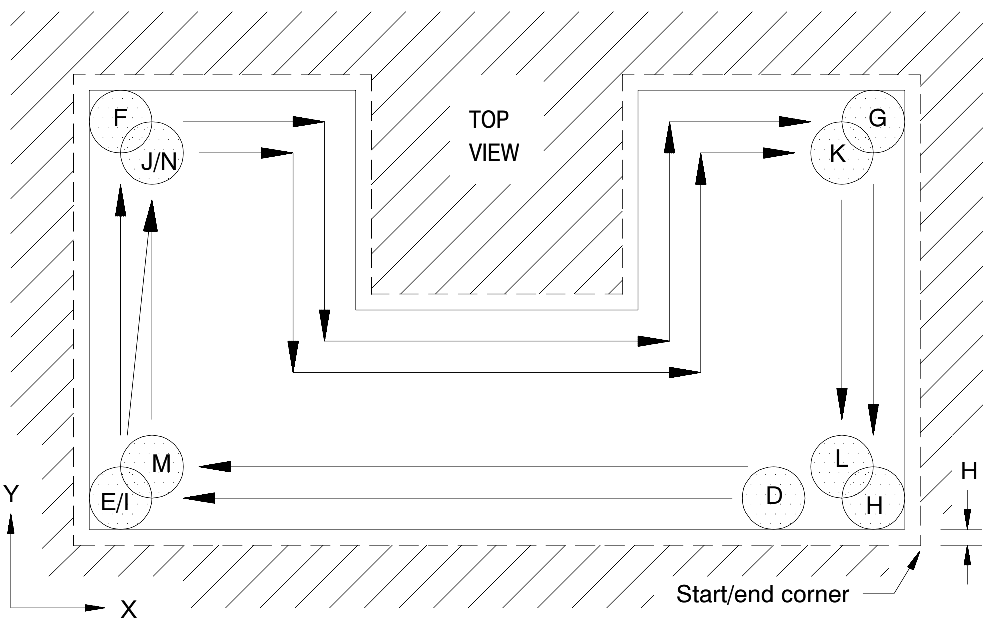
Figure 24.1
Irregular Pocket Roughing Cycle Entry Moves



From the final cutter position of Figure 24.1, the control moves the cutter twice around the programmed contour of the pocket as shown in Figure 24.2. The first pass around the pocket removes material twice the cutter radius (cutter diameter) in width (D through I in Figure 24.2). The second pass removes material that is one cutter radius in width (I through N in Figure 24.2).

These two passes cut a channel around the inside perimeter of the pocket that provides clearance for the cutter to be raised and lowered as necessary at the beginning and end of the rest of the roughing passes. While cutting this channel, the control automatically adjusts the roughing feedrate so that the volume of material being removed per unit time is the same as will be removed later during normal roughing passes.

Figure 24.2
Irregular Pocket Initial Roughing Passes

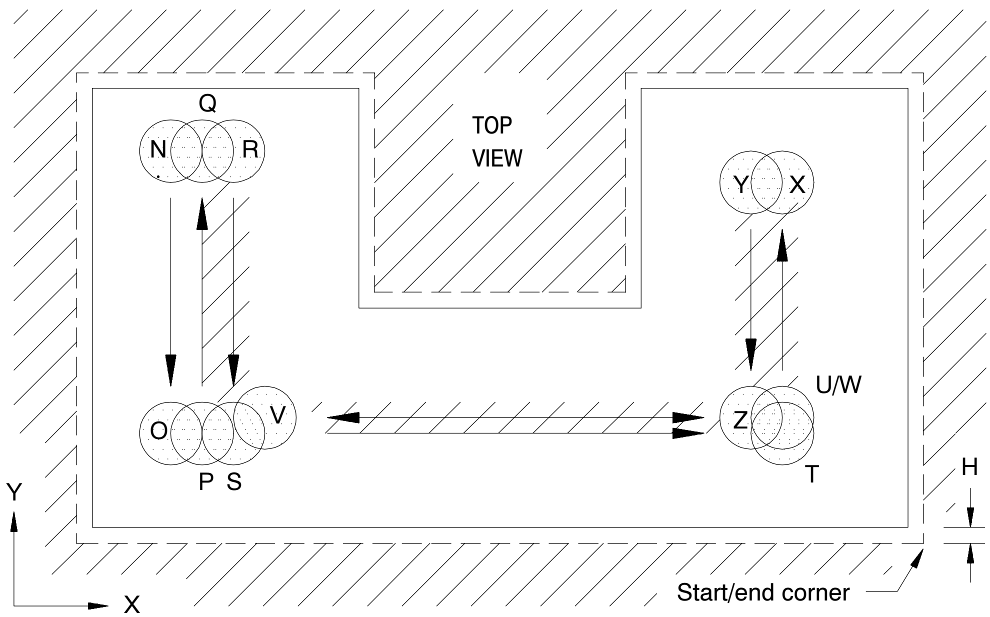


After the initial roughing passes are complete, the cutter is near the pocket corner that corresponds to the endpoint of the block that follows the block specified by the P parameter in the G89.1 block (point N in Figure 24.3).

The control analyzes the remaining pocket area to be machined-out and determines which area to machine first. Any undone area within one cutter radius of the current cutter position will be machined-out first.

The area within one cutter radius of the current cutter position will be machined-out in a series of straight-line step-over roughing passes. The orientation of these passes, either parallel to the X or the Y axis, is dependent upon the pocket contour. The control determines whether passes parallel to the X or to the Y axis will machine-out the most material in the fewest amount of passes and orients the passes accordingly (N through Z in Figure 24.3).

Figure 24.3
Roughing-Out Adjacent Areas in an Irregular Pocket

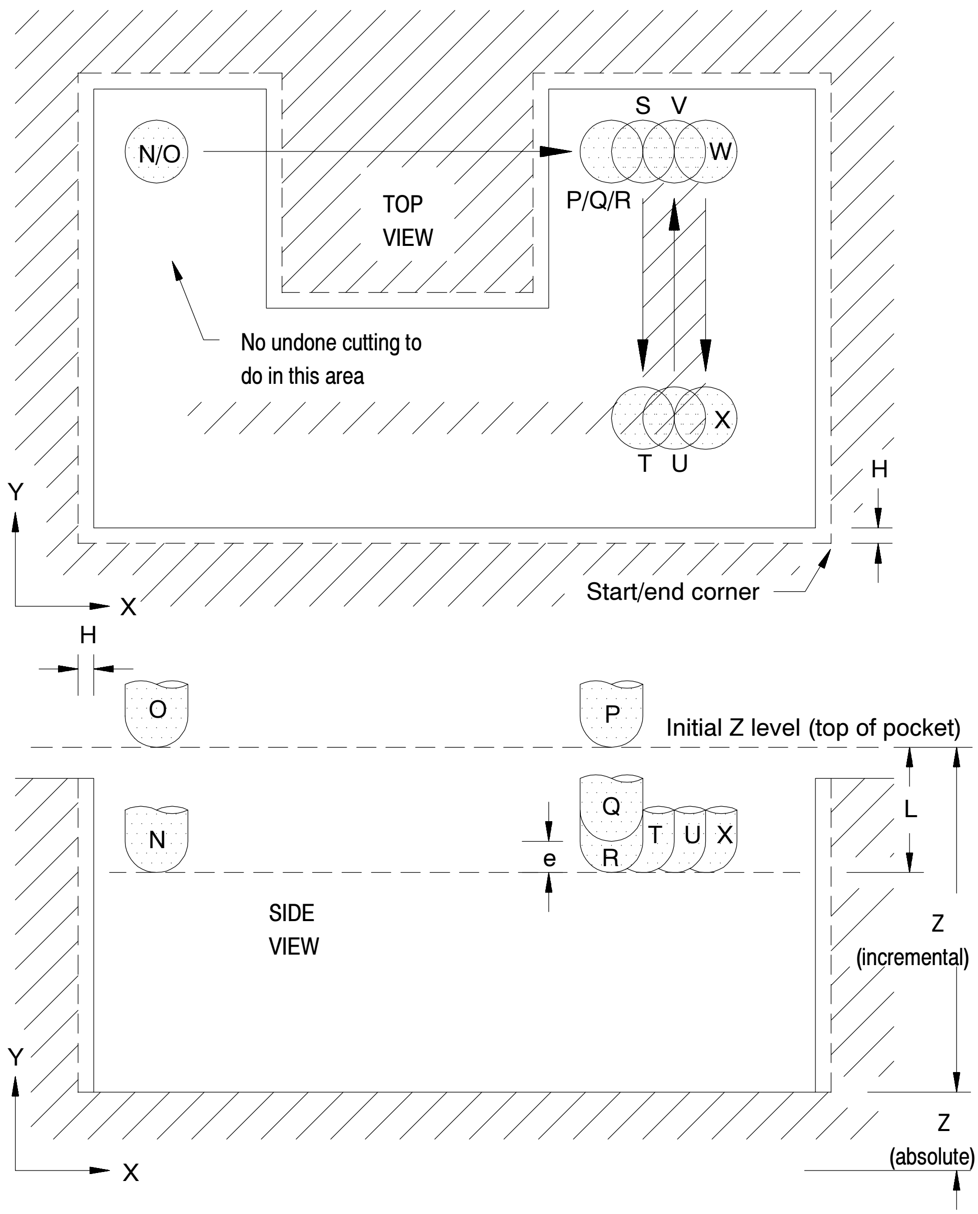


If there is no undone area within one cutter radius of the current cutter position, the control raises the cutter to the initial Z level (point O in figure 16.19). The cutter is then moved at the rapid feedrate to the nearest position just outside the nearest undone area (point P) and lowered to the current plunge-level minus the clearance amount (point Q) at the rapid feedrate. The clearance amount (e) is AMP selectable, refer to the literature provided by your system installer. The cutter is then lowered to the current plunge-level (point R) at the plunge feedrate.

Important: The control will move the cutter to the nearest undone area without raising the cutter to the initial Z level if it can do so without interfering with a pocket wall. In this case, the control will raise the cutter the clearance amount (e) and move it to the new position. The cutter is then lowered to the current plunge-level at the plunge feedrate.

The area within one cutter radius of the current cutter position will be machined-out in a series of straight-line step-over roughing passes. The orientation of these passes, either parallel to the X or the Y axis, is dependent upon the pocket contour. The control determines whether passes parallel to the X or to the Y axis will machine-out the most material in the fewest amount of passes and orients the passes accordingly (R through X in Figure 24.4).

Figure 24.4
Roughing-Out Non-Adjacent Areas in an Irregular Pocket



Once the current plunge-level has been machined-out , the cutter is moved back to the start-point. The control raises cutter to the initial Z level then moves it to the start-point.

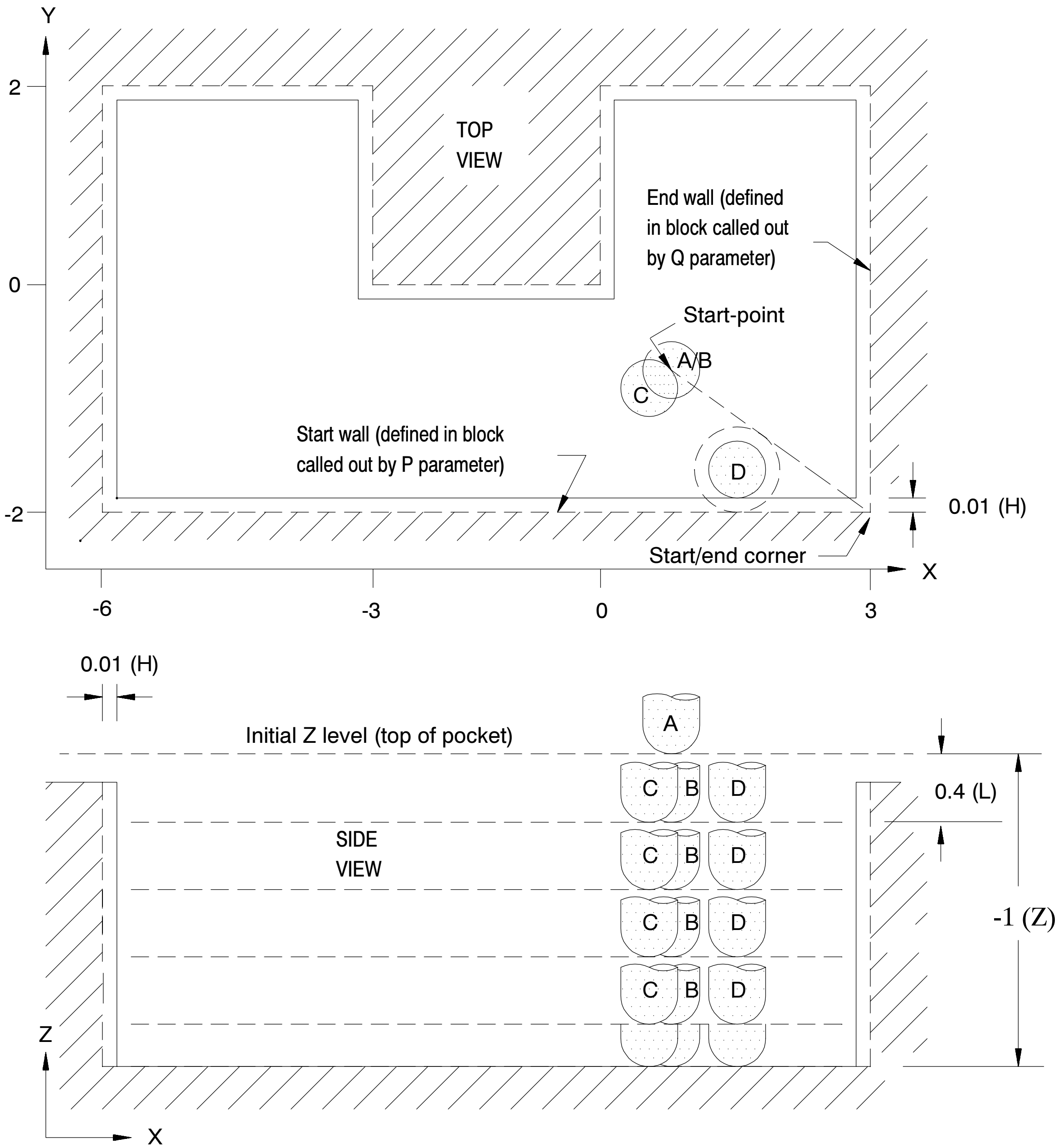
If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then machined as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

Once the programmed depth is reached, the control raises the cutter to the initial Z level then moves it to the start-point. This completes the irregular pocket roughing cycle. Example 24.1 shows an irregular pocket roughing cycle.

Example 24.1
Irregular Pocket Roughing Cycle

Program Block
N46 G92 X1 Y-1 Z0;
N47 G10 L12 P1 R.125;
N48 G90 G42 D1;
N49 G89.1 X3 Y-2 Z-1 P50 Q57 H.01 E5 F100 L.4;
N50 X-6;
N51 Y2;
N52 X-3;
N53 Y0;
N54 X0;
N55 Y2;
N56 X3;
N57 Y-2;
N58 M30;

Figure 24.5
Results of Example 24.1



Irregular Pocket Finishing (G89.2)

Use the irregular pocket milling finishing cycle (G89.2) to finish an irregular pocket in a workpiece. This cycle is typically used to finish an irregular pocket formed using a G89.1 irregular pocket roughing cycle. A tool change is usually performed between the G89.1 and G89.2 cycles. You can use this cycle to finish a post that was formed by combining two pocket cycles.

Important: You may only execute an irregular pocket finishing cycle (G89.2) in a main program.

The G89.2 block has this format:

```
G89.2  X__Y__Z__P__Q__H__F__L__;
```

Where :	Is :
X Y	The coordinates that specify the start/end corner of the irregular pocket in the selected plane. These parameters must be programmed.
Z	The coordinate (along the plunging axis) that specifies the bottom of the irregular pocket. In incremental mode this parameter specifies the depth of the pocket as measured from the start-point to the pocket bottom. This parameter must be programmed.
P	The sequence number of the first block in the set of blocks that define the pocket shape. The coordinates in this block specify the start-point of the pocket's first wall. This start-point must correspond to the start/end corner specified by the X and Y coordinates. This parameter must be programmed.
Q	The sequence number of the last block in the set of blocks that define the pocket shape. The coordinates in this block specify the endpoint of the pocket's last wall. This endpoint must correspond to the start/end corner specified by the X and Y coordinates. This parameter must be programmed. The set of blocks specified by P and Q may be located anywhere after the calling block (even after an end of program command), as long as the calling block is in the same program as the set of blocks. This means that blocks defining the pocket shape can not be called from a subprogram or a macro unless the calling block is in that subprogram or macro.
H	The finish allowance that will be left on the sides of the irregular pocket. This is an optional parameter that provides for multiple finishing cycles.
F	Finishing feedrate. This parameter determines the feedrate of any XY axis cutting moves. If not programmed the existing (modal) cutting feedrate will be used.
L	Incremental plunge depth of each cutting pass along the Z axis. If L is programmed, a finish pass is made at each L level. If L is not programmed, only one finishing pass is made at the programmed Z depth. This is an optional parameter. It is typically programmed when a very deep pocket is being finished.

Before invoking the G89.2 cycle, the programmer must activate cutter compensation left or right by programming G41 or G42. This allows the control to begin interpreting the blocks that define the contour of the pocket as they are encountered.

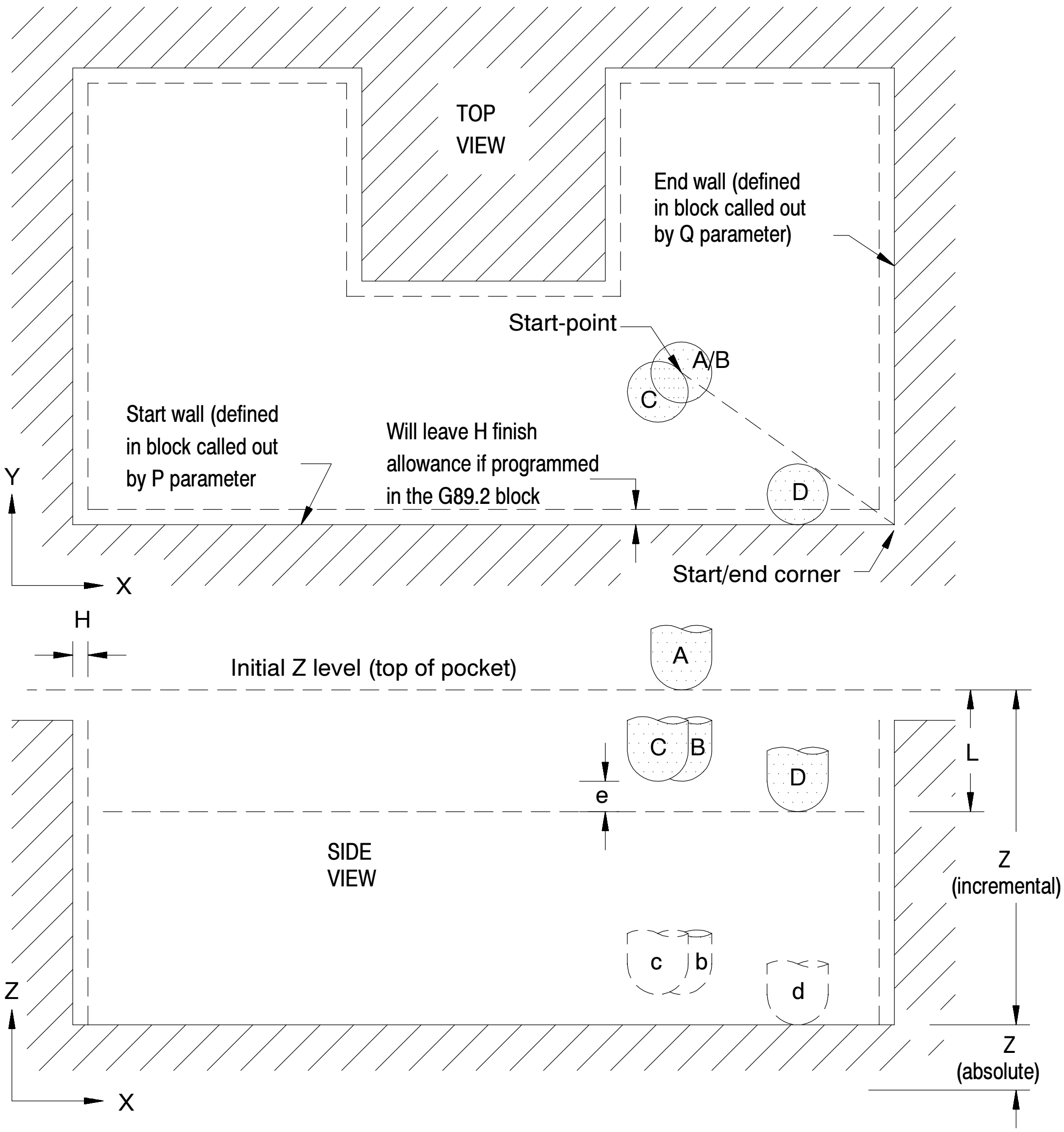
ATTENTION: From the start-point the cutter must be able to move down into the part and then directly over to the start/end corner of the pocket (A through D in Figure 24.6) without cutting into any wall of the pocket.

Also, the cutter must be able to move from the end-point of the P block to the start-point (I through K in Figure 24.7) without cutting into any wall of the pocket.

The cutter should be positioned at the start-point as described for the G89.1 cycle (A in Figure 24.6). Once the axes are positioned at the start-point, a plunge along the Z axis takes place. If L is programmed, the tool plunges along the Z axis to the incremental depth specified by the L parameter minus the clearance amount (B in Figure 24.6). If L is not programmed, the tool plunges along the Z axis to the pocket depth specified by the Z parameter minus the clearance amount (B in Figure 24.6). The plunge takes place at the rapid feedrate.

After the plunge, the control uses the active cutter compensation mode (G41/G42) to offset the cutter one cutter radius perpendicular to the line from the start-point to the start/end corner (C in Figure 24.6). The control then moves the cutter from this offset position to a point where it is located one cutter radius plus the finish allowance (if H is programmed) away from both the first wall of the pocket and the line from the start-point to the start/end corner. At the same time, the cutter is is lowered the clearance amount along the Z axis (D in Figure 24.6). This move takes place at the finishing feedrate.

Figure 24.6
Irregular Pocket Finishing Cycle Entry Moves

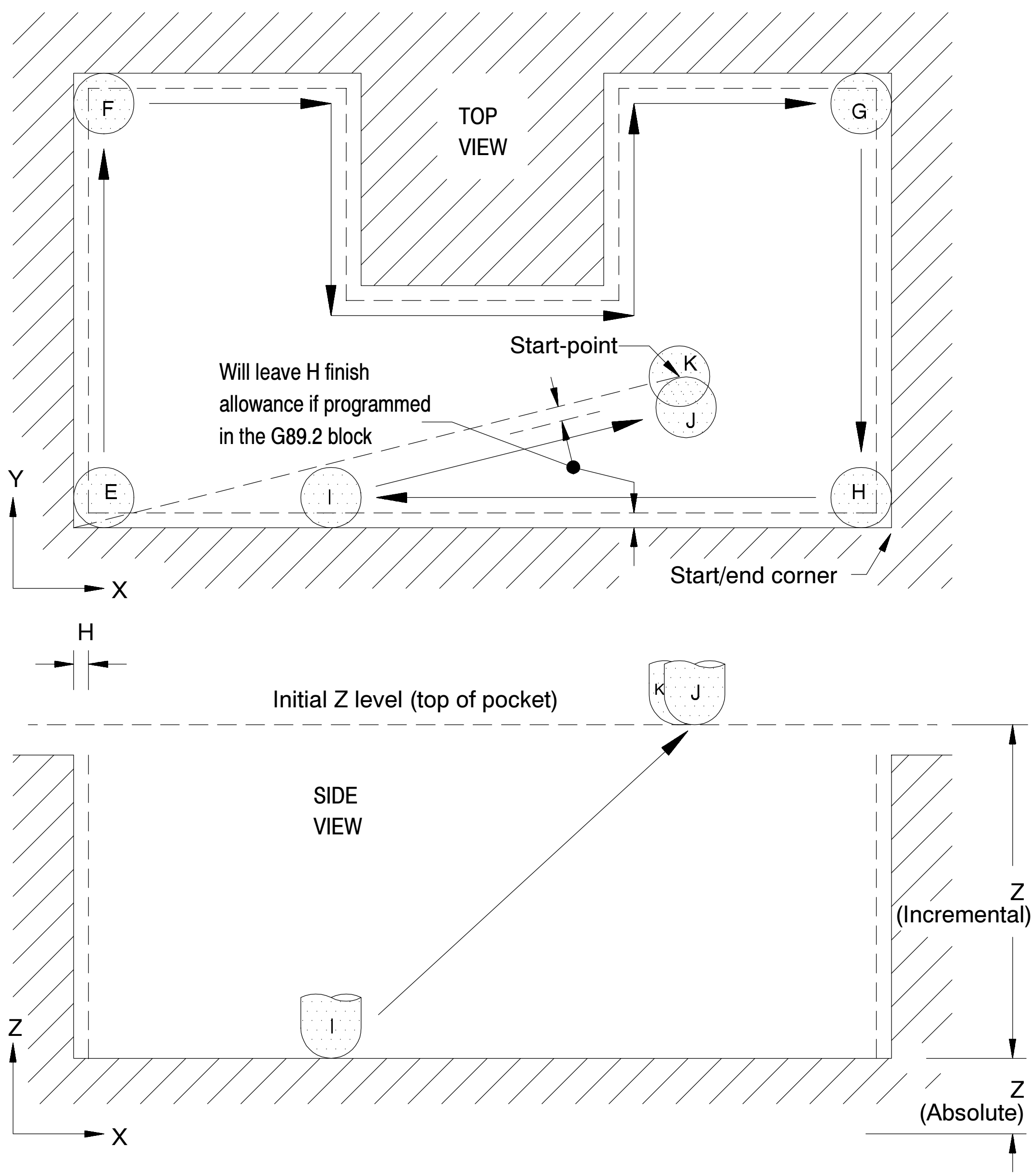


If H is programmed in the G89.2 block, an initial finish pass that leaves a finish allowance is made around the sides of the pocket. This typically would be done when it is desirable to clean out the corners in an irregular pocket before actually removing the finish allowance.

If H is not programmed in the G89.2 block, a finish pass is made around the sides of the pocket that removes all the finish allowance that was left during a G89.1 roughing cycle.

The finish pass ends at a point along the start-wall that is determined by the angle formed by the start-wall and a line drawn from the endpoint of the start-wall to the start-point. An example of this is shown in the following figure. From this point the cutter is moved back to the start-point of the cycle.

Figure 24.7
Irregular Pocket Finishing Cycle Exit Moves



ATTENTION: The cutter must be able to move from the end-point of the P block to the start-point (I through K in Figure 24.7) without cutting into any wall of the pocket.

If the programmed Z depth of the pocket has not been reached, another plunge along the Z axis to the next L level takes place. This level is then finished as described in the previous paragraphs. This process is repeated until the programmed Z depth is reached.

Once the programmed depth is reached, the control simultaneously raises the cutter and moves it towards the start-point. The control shuts off cutter compensation prior to reaching the start-point (J and K in Figure 24.7). This completes the irregular pocket finishing cycle.

END OF CHAPTER

Milling Fixed Cycles

Chapter Overview

This chapter covers the G-word data blocks in the milling fixed-cycle group. The operations of the milling fixed cycles are explained in these sections:

For this chapter, as well as this manual, make the following assumptions:

- X and Y axes are the positioning axes (G17 plane).
- Z axis is the hole machining axis for drilling, boring, and tapping applications.

Milling Fixed Cycles

Milling fixed cycles (sometimes referred to as canned cycles or autocycles cycles) repeat a series of basic machining operations, such as, boring, drilling or tapping. These operations, designated by a single block command, usually consist of a fixed series of steps that are dependent on the type of machining application.

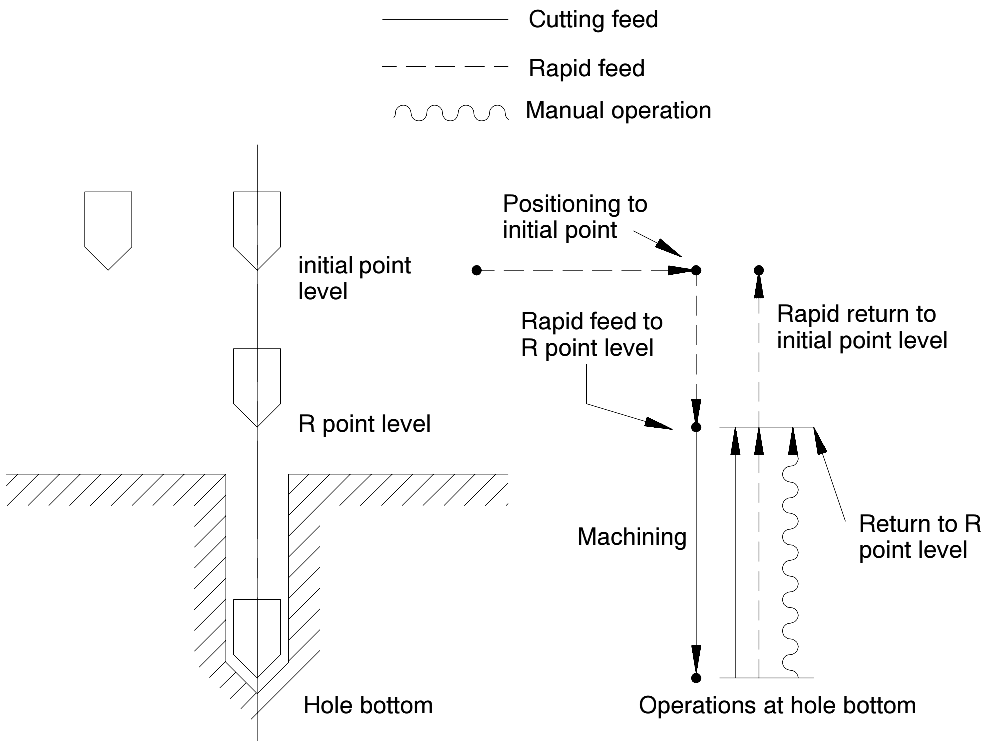
The control provides the milling fixed cycles shown in Table 25.A.

Table 25.A
Milling Fixed Cycles

G-code	Application	Tool Movement	Operation At Hole Bottom	Retraction Movement
G73	Deep Hole Peck Drilling Cycle with Dwell	Intermittent Feed	Retract	Rapid Traverse
G74	Left-Hand Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G74.1	Left-Hand Solid-Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G76	Boring Cycle, Spindle Shift	Feed	Oriented Spindle Stop / Retract	Rapid Traverse
G80	Cancel Or End Fixed Cycle	N/A	N/A	N/A
G81	Drilling Cycle, No Dwell/Rapid Out	Feed	Retract	Rapid Traverse
G82	Drilling Cycle, Dwell/Rapid Out	Feed	Dwell /Retract	Rapid Traverse
G83	Deep Hole Drilling Cycle	Intermittent Feed	Retract	Rapid Traverse
G84	Right-Hand Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G84.1	Right-Hand Solid-Tapping Cycle	Feed	Spindle Reversed / Retract	Feed
G85	Boring Cycle, No Dwell/Feed Out	Feed	Retract	Feed
G86	Boring Cycle, Spindle Stop/Rapid Out	Feed	Spindle Stop /Retract	Rapid Traverse
G87	Back Boring Cycle	Feed	Oriented Spindle Stop / Retract	Rapid Traverse
G88	Boring Cycle Spindle Stop/ Manually Out	Feed	Dwell / Retract Spindle Stop / Retract	Manual/Rapid Traverse
G89	Boring Cycle, Dwell/Feed Out	Feed	Dwell / Retract	Feed

In general, milling fixed cycles consist of the following operations (see Figure 25.1):

Figure 25.1
Milling Fixed Cycle Operations



The system installer determines if the positioning to initial point is always a rapid move, or if it is necessary to program a G00 or G01 to select a mode. This manual assumes rapid positioning.

Positioning and Hole Machining Axes

This section assumes that the programmer can determine the hole machining axis using the plane select G-codes (G17, G18, and G19). Refer to the system installer's documentation to make sure that a specific axis has not been selected in AMP to be the hole machining axis.

G-codes, G17, G18 or G19, determine the plane, the positioning axes and the hole machining axis. The two axes that define the selected plane are used as positioning axes; the axis **perpendicular** to the plane is the hole machining axis.

Table 25.B below assumes that the system installer has not altered the default values defining the G17, G18 or G19 plane select codes.

Table 25.B
Plane Selection vs Machining Axis

Plane	Hole Machining Axis	Positioning Axes
XY (G17)	Z axis or its parallel axis	X and Y axes or their parallel axes
ZX (G18)	Y axis or its parallel axis	Z and X axes or their parallel axes
YZ (G19)	X axis or its parallel axis	Y and Z axes or their parallel axes

Example 25.1 shows you how to change the hole machining axis to a parallel axis. A G80 should be executed to cancel any active milling mode, prior to changing the hole machining axis.

Example 25.1
Altering the Machining Axis to a Parallel Axis

Program Block	Comment
The W axis is parallel to the Z axis.	
G17;	XY plane active
G81X ___ Y ___ ;	Drilling cycle, Z is the hole machining axis
.	
.	
G80;	Cancel milling cycle mode
G81X ___ Y ___ W ___;	Drilling cycle, W is the hole machining axis
. . .	W must be programmed in every subsequent block to remain the drilling axis. If it is not programmed, Z becomes the drilling axis.

The plane selection codes (G17-G19) can be included in the milling fixed cycle block, or can be programmed in a previous block.

Figure 25.2 shows typical milling fixed cycle motions in absolute (G90) or incremental (G91) modes. Note the changes in how the R point and Z level are referenced.

Figure 25.2
Milling Fixed Cycle Parameters in G90 and G91 Modes

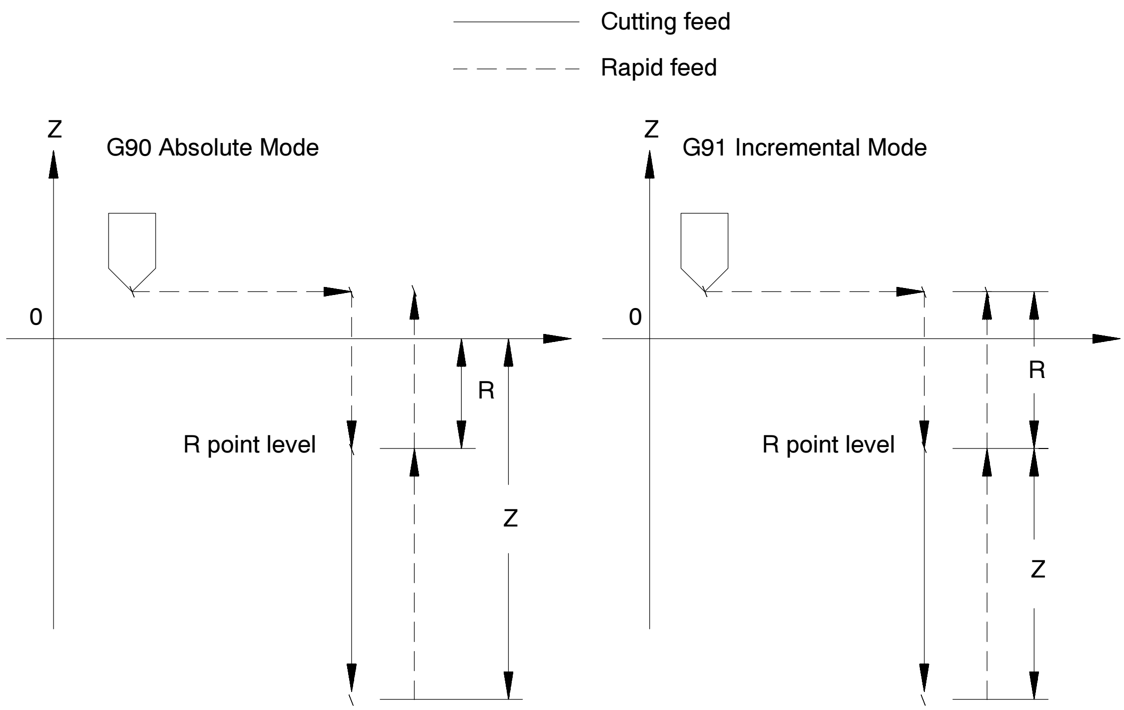
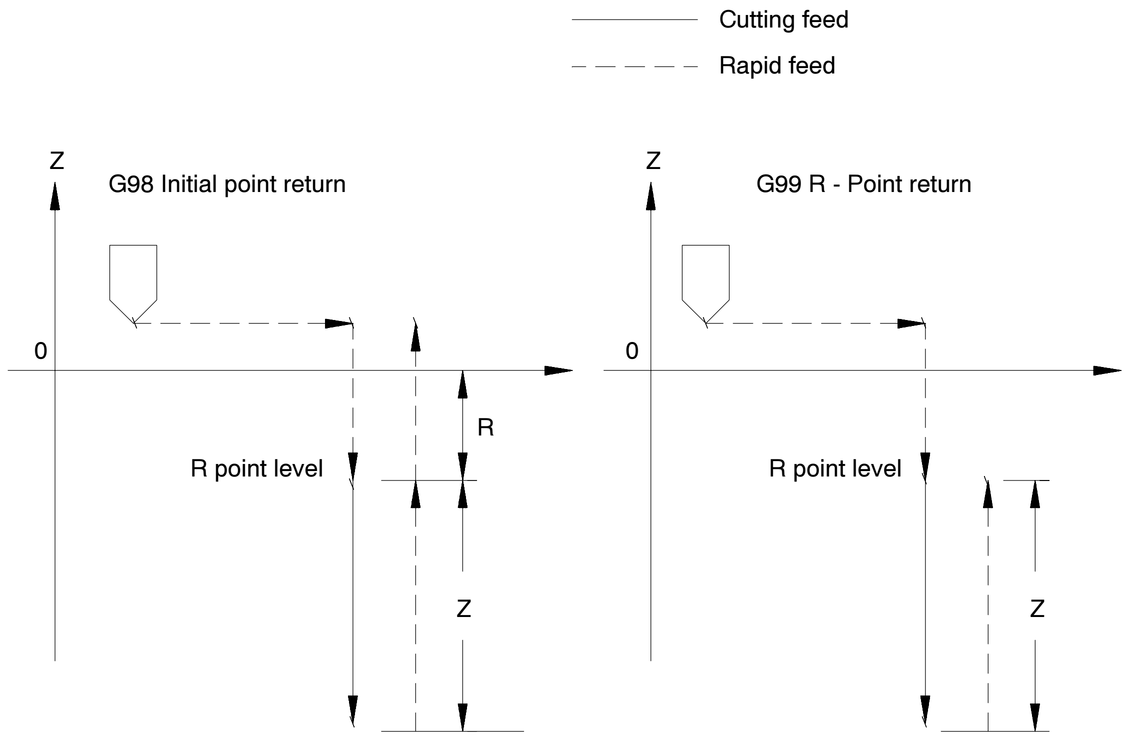


Figure 25.3 shows the two different modes available for selecting the return level in the Z axis after the hole has been drilled. These two modes are selected with G98 (which returns to the same level the cycle started at) and G99 (which returns to the level defined by the R point).

Figure 25.3
Milling Fixed Cycle Parameters in G98 or G99 Modes



Important: In the following sections, figures and examples are assumed to be programmed in the incremental mode (G91) and initial point return mode (G98).

Parameters

The following section provides a detailed explanation of each parameter that can be programmed for the milling fixed cycles. Some of these parameters are not valid with all cycles. Refer to the specific description of each cycle on page 25-8. To alter milling cycle operation parameters, refer to page 25-34.

We describe these milling fixed-cycle parameters below.

```
X__Y__Z__R__{I__J__K__}P__F__L__Q__D__S__;
```

Where :	Is :
X,Y	**specifies the location of the hole position in the selected plane.** In the absolute mode (G90), program the hole position using the coordinate values in the active coordinate system. In incremental mode (G91), program the hole position using the distance from the current tool position to the required hole position.
Z	**defines the hole bottom.** In absolute mode (G90), program the hole bottom level using the coordinate value in the active coordinate system. In the incremental mode (G91), program the distance from the R point level to the hole bottom level.
R	**defines the R point level.** In the absolute mode (G90), program the R point level as a coordinate value in the active coordinate system. In the incremental mode (G91), program the R point level by the distance from the initial point level to the R point level.
I, J, K, or Q	**Q defines the infeed amount** for each move made in the hole in G83**; I, J, and K, or Q defines the shift amount** for G76 and G87.
P	**defines the dwell period at hole bottom.** P programs the dwell in the same way as G04: seconds if in feedrate mode (G94), spindle revolutions if in revolution mode (G95). (The allowable dwell time range in seconds is 0.001-99999.99. The allowable dwell range in revolutions is also 0.001-99999.999.) The P-word does not apply in all milling fixed cycles.
F	**defines the cutting feedrate. If** this parameter is not specified, the control will use the currently active feedrate for the cutting feedrate. For G74.1 and G84.1, F = tap thread lead in inches/mm per revolution.
L	**defines the number of times the milling fixed cycle is repeated.** The maximum number of repeats is 9999. • In absolute mode, the control drills in the same location the number of times specified by the L-word. • In incremental mode, the L-word drills the number of holes specified by the L-word at equally spaced positions, determined by axis positioning parameters X and Y. • If an L0 is programmed, the control stores the milling cycle information but does not execute the drilling cycle. If no L-word is programmed, the control defaults to L1.
Q	In G83, **Q defines the infeed amount** for each move made in the hole. In G86.1 and G87, **Q defines the shift amount** (as do I, J, and K). In G74.1 and G84.1, **Q defines the angle** at which to **orient** the spindle before starting the tap. If you don't program the Q-word, the spindle is not oriented before the tap begins. This means that the hole is not re-tappable unless a Q-word is programmed in the cycle block. The spindle is brought to a stop prior to the initiation of the tapping phase even if Q is not programmed; this happens after the move to the R-plane.
D	**defines the return spindle speed** so that, if you want, the tap-out move can be performed faster or slower than the tap-in. Tool selection by D-word is not possible while in the solid tapping mode.
S	**defines spindle speed in rpm.**

Important: After programming a milling fixed cycle block, parameters X, Y, Z and R can be programmed in later blocks with different values. This, of course, permits axis motion to be changed. Parameters Q, P, I and K can only be programmed in the calling block for the milling fixed cycle. They cannot be programmed following the calling block. If they are, the control will ignore them.

Milling Fixed Cycle Operations

This section describes how the control executes each milling fixed cycle. The following is assumed for each cycle:

- initial point level is the return level (G98 is active)
- incremental mode is active (G91 is active)
- the X and Y axes are the positioning axes
- the Z axis is the hole machining axis

The milling fixed cycles are modal, which means they remain active until a G-code that cancels the milling fixed cycle is programmed. A milling fixed cycle can, therefore, be repeated at different positions, without having to re-program all the parameters associated with a given operation.

ATTENTION: The controlling spindle code determines which spindle and its related spindle M-codes (modal) will be active during milling cycles. When spindle is mentioned in relation to milling fixed cycles, we are referring to the controlling spindle. For more information on controlling spindles, refer to chapter 16.

Similarly, any parameters specified in the block with the G-code of the milling fixed cycle remain active until the cycle is cancelled, or until they are programmed again in a following block. L-words do **not** remain active and, instead, are active only for the block which contains the actual L-word.

G00, G01, G02, and G80 will cancel milling fixed cycle modes.

(G73): Deep Hole Peck Drilling Cycle with Dwell

The format for the G73 cycle is as follows:

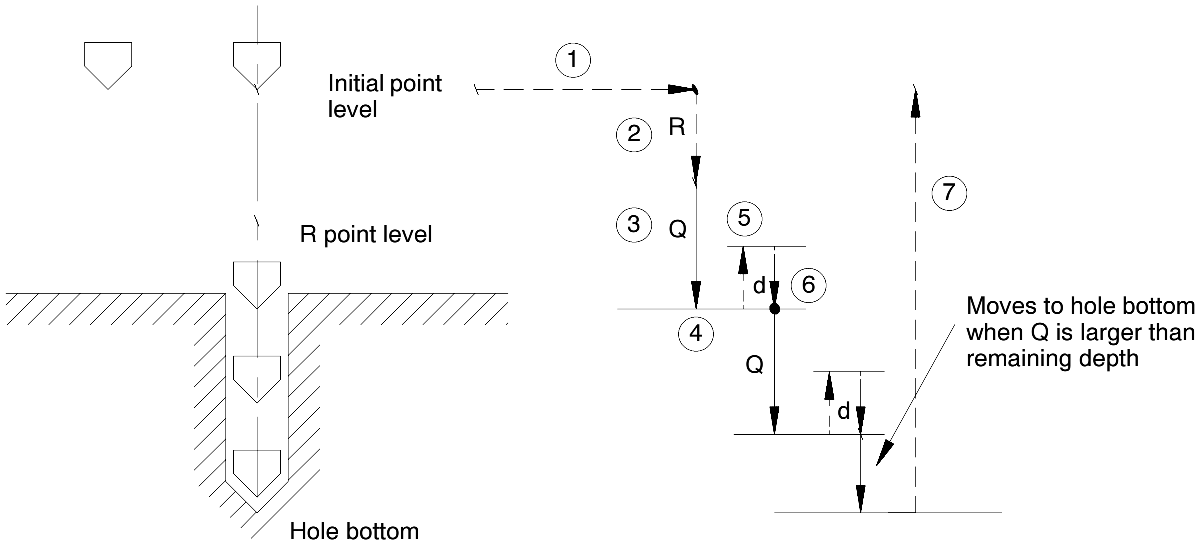
```
G73X__Y__Z__R__Q__P__F__L__;
```

Where :	Is :
X,Y	specifies the location of the hole position in the selected plane.
Z	defines the hole bottom.
R	defines the R point level.
Q	defines the infeed amount for each step into the hole.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.4
G73: Deep Hole Peck Drilling Cycle with Dwell



In the G73 peck drilling cycle, the control moves the axes in this manner:

1. The tool rapids to the initial point level above the hole location.
2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the drilling operation.
3. During the drilling operation, the control infeeds the drilling tool by an amount Q, as programmed in the drilling cycle.
4. If a value was programmed for the P parameter, the drilling tool will dwell after it reaches the bottom of the hole.
5. It then retracts by an amount d at a rapid feedrate. The amount d is specified by the system installer, or can be set by the operator as described on page 25-34. This intermittent feed simplifies chip disposal and lets a small retraction amount to be set in peck drilling.
6. After the drilling tool retracts an amount d, it then resumes drilling at the cutting feedrate to a depth d + Q.

This retraction and extension continues until the drilling tool reaches the depth of the hole as programmed with the Z-word in the drilling cycle block.

7. The drilling tool then retracts at a rapid feedrate to the initial point level as determined by G98.

When the **single block** function is active, the control stops axis motion and awaits "cycle start" after steps 1, 2 and 7.

(G74): Left-hand Tapping Cycle

This cycle is used to cut left-handed threads.

ATTENTION: The programmer or operator must set the direction of spindle rotation for tap-in. The control forces the proper spindle direction for the tap-out, but uses the programmed spindle direction for the tap-in.

The format for the G74 cycle is as follows:

```
G74X__Y__Z__R__P__F__L__;
```

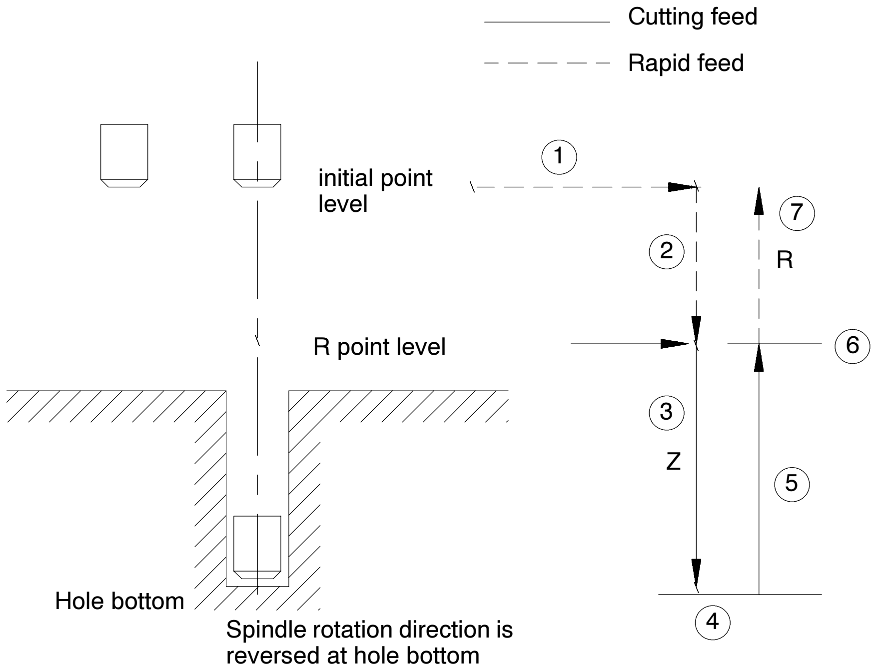
Where :	Is :
X,Y	specifies the location of the hole position in the selected plane.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the tapping feedrate. This should be programmed as close as possible to the rate in which the tap will be moving into the part (calculated from the tap thread pitch and the active spindle speed). Enter the feedrate in either IPM or IPR modes. No special spindle synchronization occurs with this cycle.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: When programming a G74 tapping cycle, consider this:

- The programmer or operator must start spindle rotation.
- **Override** usage- the control ignores the feedrate override switch and clamps override at 100 percent.
- During tapping the feedrate override switch, and the feedhold feature are both disabled. Cycle stop is not acknowledged until the end of the return operation.

Figure 25.5
G74: Left-Hand Tapping Cycle



In the G74 left-hand tapping cycle, the control moves the axes in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The threading tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the tapping operation.
3. During the tapping operation, the control infeeds the threading tool to the depth and at the feedrate programmed in the tapping cycle.
4. If a value was programmed for the P parameter, the threading tool dwells after it reaches the bottom of the hole, and after the spindle has been commanded to reverse.

The spindle reverses to the clockwise direction.

5. The threading tool retracts at the cutting feedrate to the R point.
6. If a value was programmed for the P parameter, the threading tool will dwell after it reaches the R point. (Dwells may be ignored if the system installer has chosen to do so in AMP.)

Then the spindle direction is reversed to counterclockwise.

7. With G98 active, the cutting tool will then accelerate to the rapid feedrate and retract to the initial point level.

When the **single block** function is active, the control stops axis motion and awaits “cycle start” after steps 1, 2 and 7.

If the operator activates a **feedhold** during steps 3, 4, or 5, axis motion stops after step 7. Axis motion will also stop during steps 1, 2 and 7. However, if feedhold is activated during step 7, axis motion will stop immediately.

Important: Your system installer can enable a tap retract feature for this cycle through PAL. Tap retract enables you to retract the tapping tool and resume the cycle, or completely abort the tapping operation. Refer to your system installers documentation for details.

(G74.1): Left-hand Solid-tapping Cycle

Use this cycle to cut left-handed threads.

The format for the G74.1 cycle is:

```
G74.1X__Z__R__F__L__Q__D__S__;
```

Where :	Is :
X	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	represents the thread lead along the drilling axis (Z in this manual). It is mandatory and modal in any subsequent solid tapping cycle blocks until a new F-word is programmed. The control interprets the F-word as the thread lead in inches per revolution or millimeters per revolution, depending on the inch/metric mode active.
L	defines the number of times the drilling cycle is repeated.
Q	defines the angle at which to orient the spindle before starting the tap. It is modal in any subsequent solid tapping blocks until a new Q-word is programmed or the tapping cycle is cancelled by a G80. To retap a hole, a Q-word must have been programmed when the hole was originally tapped.
D	defines the return spindle speed, but cannot exceed the maximum tapping spindle speed set in AMP. This will adjust your Z-axis feedrate according to the thread lead defined in F.
S	defines spindle speed in rpm.

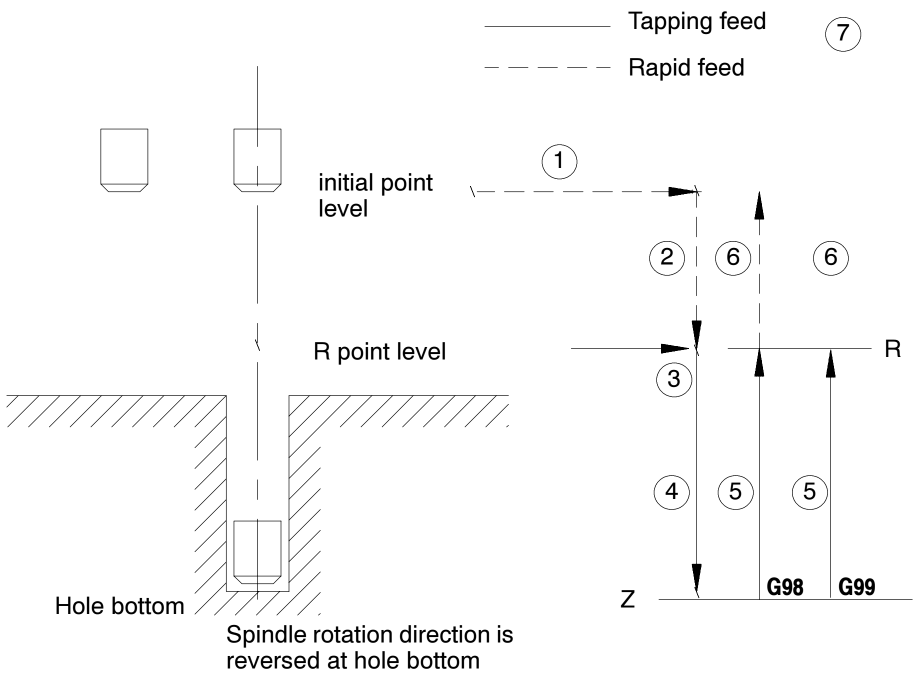
(Refer to page 25-6 for a detailed description of these parameters.)

Important: When programming and executing a G74.1 solid tapping cycle, remember:

- the feedrate of the tapping phases is derived as
  `(spindle {RPM} * F-lead {IPR}) = IPM`

- the spindle speed that is active at the start of the cycle determines the effective Z feedrate
- the direction of spindle rotation for tap-in and tap-out phases will be automatically generated by the control
- spindle speed override has no effect on the solid tapping cycle; you can use feedrate override to adjust the tapping operation
- `D` cannot exceed the maximum tapping spindle speed set in AMP
- you cannot select tools via `D`-word while in solid tapping mode
- gear changes are locked out
- cycle stop is acknowledged throughout the cycle, but can be disabled by G63
- you can use active reset to abort the cycle after the cycle stop request has been acknowledged
- to re-tap a hole, a `Q`-word must have been programmed when the hole was originally tapped
- block retrace is possible during the tap-in portion of the cycle, but not during the tap-out

Figure 25.6
G74.1: Left-Hand Solid-Tapping Cycle



In the G74.1 left-hand solid-tapping cycle, the control moves the axes in this manner:

1. The tool rapids to the tapping position above the hole location.
2. The threading tool then rapids to the R point.
3. The control either orients or stops the spindle.

If a Q-word was programmed:	the control:
yes	orients the spindle
no	stops the spindle

4. *Tap-in:* The counterclockwise rotation of the spindle ramps up to the programmed `S` spindle speed and linear motion of the Z axis moves synchronously to reach the Z position.
5. *Tap-out:* The spindle and linear motion reverse to the clockwise direction and retract to the R point.

The tap-out speed is determined by `F * S` unless you programmed `D` (tap-out rpm), in which case tap-out speed is `F * D`.

At the R point, spindle rotation has ramped to zero.

6. With G98 active, the cutting tool then accelerates to the rapid feedrate and retracts to the initial point level.

With G99 active, the cutting tool remains at R point; no movement occurs.

In single-block mode, the control stops axis motion after phases 1, 2, 3, and 6 of the cycle (Figure 27.M1).

Important: When it is active, S-Curve Acc/Dec (G47.1) will be applied to the rapid feedrate portions of the solid-tapping cycle.

(G76): Boring Cycle Spindle Shift

The format for the G76 cycle is as follows:

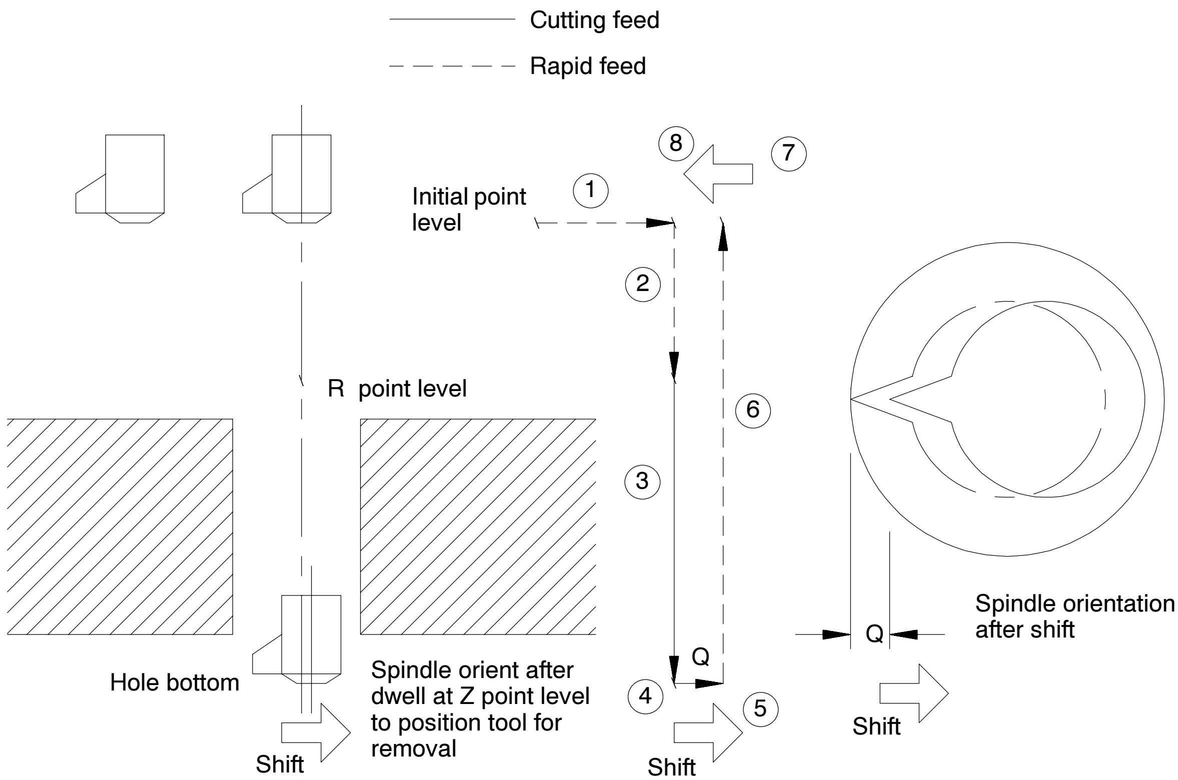
$$\text{G76X\_\_Y\_\_Z\_\_}\left\{\begin{matrix}\text{I\_\_J\_\_K\_\_}\\ \text{Q\_\_}\end{matrix}\right\}\text{R\_\_F\_\_L\_\_;}$$

Where :	Is :
`X,Y`	specifies location of the hole.
`Z`	defines the hole bottom.
`Q or I, J, K`	defines the tool shift amount and direction.
`R`	defines the R point level.
`F`	defines the cutting feedrate.
`L`	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.7
G76: Boring Cycle, Spindle Shift



In the G76 boring cycle, the control moves the axes in this manner:

1. The tool rapids to the initial point level above the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool bores at the programmed feedrate to the pre-programmed depth of the hole (defined by the Z-word in the boring cycle block).
4. The control positions the spindle at the bottom of the hole in a particular orientation, determined by the system installer in AMP.
5. To prevent the boring tool from damaging the sides of the bored hole during retraction, the boring tool is shifted in either of two methods, which are explained on the following page and illustrated in Figure 25.7.

   The shift direction is determined by two possible methods.

Method I

This shift method is a single axis shift. The direction and axis for the shift is set in AMP by the system installer or can be altered using the milling fixed cycle parameter table (refer to page 25-36).

- The direction of the axis is specified as + or -.
- The feedrate using this shift method is always rapid traverse.
- The Q-word shift amount is always interpreted as a positive value. A negative Q-word is not allowed.

Method II

The direction of the shift using this method is programmed in the boring cycle block. Program a shift amount for axes in the current plane only (determined by G17, G18, or G19) using the following words:

I__ programs a X axis move.
J__ programs a Y axis move.
K__ programs a Z axis move.

Follow the I-, J- and K-words (modal during milling fixed cycles) with incremental values in the block that programs the hole position.

When using Method II, remember:

- If both axes in the current plane are to be shifted, specify both words to move the axes.
- The move generated will be a single linear move and will execute at rapid traverse.

6. The boring tool is then retracted at a rapid feedrate to the initial point level as determined by G98.

7. After reaching the initial point level, the boring tool is shifted back (in a manner previously explained and illustrated) and the spindle is re-started in the counterclockwise direction again.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 8.

(G80): Cancel or End Fixed Cycles

The format for the G80 cancel or end fixed cycles is as follows:

```
G80;
```

Programming a G80 cancels the currently active milling fixed cycle mode. (G00, G01, G02, or G03 will also cancel any active milling fixed cycle.)

If milling fixed cycles are canceled with a G80, program execution returns to the mode which was in effect when the cycles were last turned on, for example, G00 - G03.

(G81): Drilling Cycle, No Dwell/Rapid Out

The format for the G81 cycle is as follows:

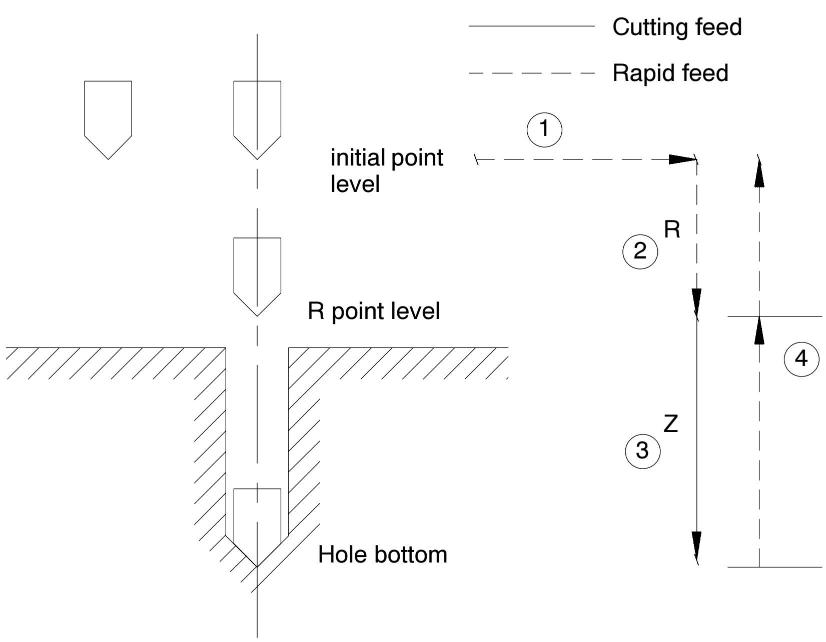
```
G81X__Y__Z__R__F__L__;
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.8
G81: Drilling Cycle without Dwell



In the G81 drilling cycle, the control moves the axes in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the drilling operation.
3. The drilling tool continues to drill at the programmed feedrate until it reaches the depth of the hole as programmed with the Z-word.
4. The control retracts the drilling tool at a rapid feedrate to the initial point level as determined by G98.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 4.

(G82): Drill Cycle Dwell/Rapid Out

The format for the G82 cycle is as follows:

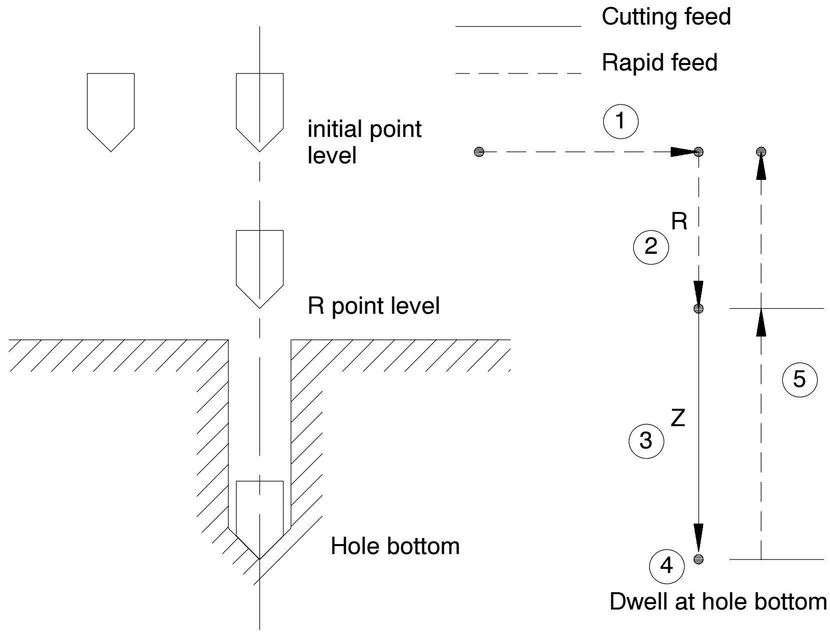
```
G82X__Y__Z__R__P__F__L__;
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.9
G82: Drilling Cycle, Dwell/Rapid Out



In the G82 drilling cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level point above the hole location.

2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the drill operation.

3. The cutting tool drills at the programmed feedrate to the pre-programmed depth of the hole (defined by the Z-word in the boring cycle block).

4. If a value was programmed for the P parameter, the drilling tool will dwell after it reaches the bottom of the hole.

5. After the drilling tool reaches the hole bottom and the dwell is completed, the drilling tool is retracted at a rapid feedrate to the initial point level as determined by G98.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 5.

(G83): Deep Hole Drilling Cycle

The format for the G83 cycle is as follows:

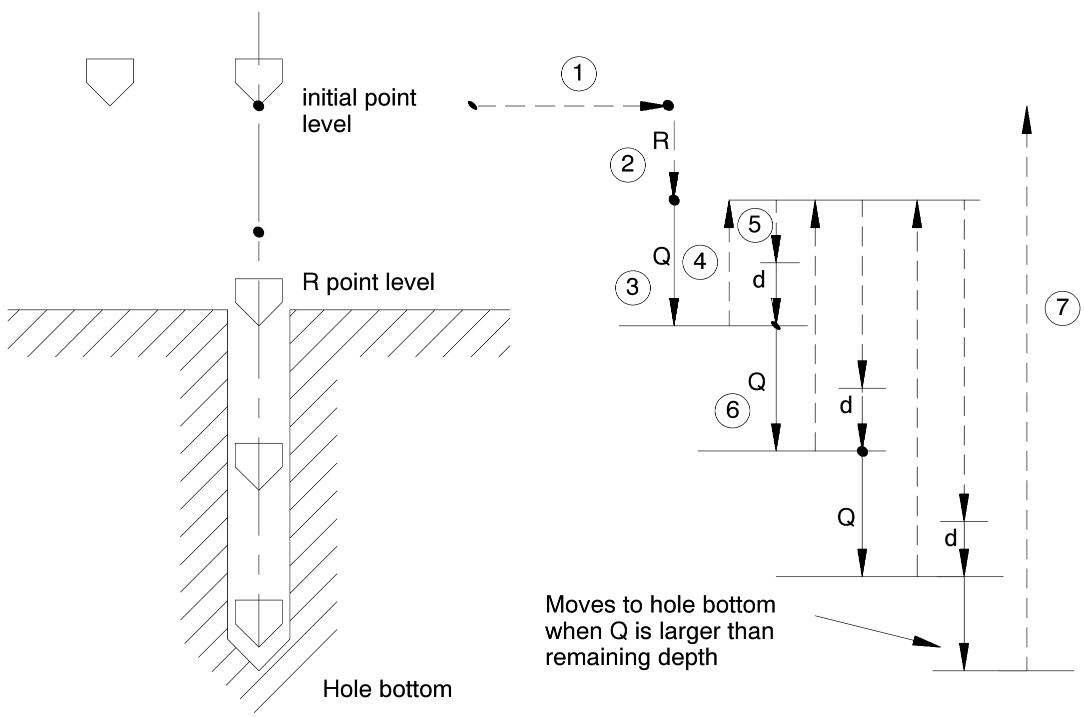
```
G83X__Y__Z__R__Q__F__L__;
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
Q	..defines the infeed amount for each step into the hole.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.10
G83: Deep Hole Drilling Cycle



In the G83 drilling cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level above the hole location.
2. The drilling tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the deep hole drilling operation.
3. During the drilling operation, the control infeeds the drilling tool by an amount Q, as programmed in the G83 block.
4. The drilling tool retracts at a rapid feedrate to the R point level.
5. The control feeds the drilling tool at rapid feedrate to a distance d above the level drilled in the previous infeed. The amount d is specified by the system installer, or can be set by the operator as described on page 25-34. This intermittent feed simplifies chip disposal and permits a very small retraction amount to be set in deep hole drilling.
6. The drilling tool slows to the cutting feedrate again and infeeds an amount Q + d.
7. The cutting tool is then retracted at a rapid feedrate to the initial point level as determined by G98.

When the **single block** function is active, the control stops axis motion after steps 1, 2, and 7.

(G84): Right-hand Tapping Cycle

This cycle is used to cut right-handed threads. The format for the G84 cycle is as follows:

```
G84X__Y__Z__R__P__F__L__;
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the tapping feedrate. This should be programmed as close as possible to the rate in which the tap will be moving into the part (calculated from the tap thread pitch and the active spindle speed). Enter the feedrate in either IPM or IPR modes. No special spindle synchronization occurs with this cycle.
L	defines the number of times the milling fixed cycle is repeated.

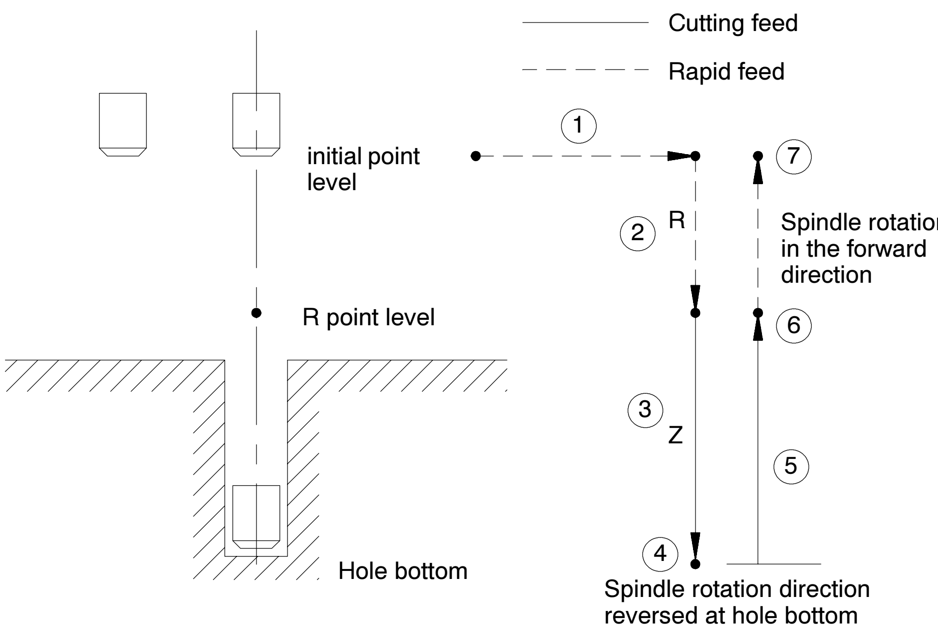
(Refer to page 25-6 for a detailed description of these parameters.)

ATTENTION: The programmer or operator must set the direction of spindle rotation for tap-in. The control forces the proper spindle direction for the tap-out, but uses the programmed spindle direction for the tap-in.

Important: When programming and executing a G84 tapping cycle, consider this:

- The programmer or operator must start spindle rotation.
- **Override** usage - the control ignores the feedrate override switch and clamps override at 100 percent.
- During tapping the feedrate override switch, and the feedhold feature are both disabled. Cycle stop is not acknowledged until the end of the return operation.

Figure 25.11
G84: Right-Hand Tapping Cycle



In the G84 right-hand tapping cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level above the hole location.
2. The threading tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the tapping operation.
3. During the tapping operation, the control infeeds the threading tool to the depth and at the feedrate programmed in the tapping cycle.

4. If a value was programmed for the P parameter, the threading tool will dwell after it reaches the bottom of the hole and after the spindle has been commanded to reverse.

   The spindle reverses to the counterclockwise direction.

5. The threading tool retracts at the cutting feedrate to the R point.
6. If a value was programmed for the P parameter, the threading tool will dwell after it reaches the R point level. (Dwells may be ignored if the system installer has chosen to do so in AMP.)

Then the spindle direction is reversed to clockwise.

7. With G98 active, the cutting tool will then accelerate to the rapid feedrate and retract to the initial point level.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 6.

If the operator activates a **feedhold** during steps 3, 4 or 5, axis motion stops after step 7. Axis motion will also stop during steps 1, 2, and 7. However, if the operator activates a feedhold during step 7, axis motion will stop immediately.

Important: Your system installer can enable a tap retract feature for this cycle through PAL. Tap retract enables you to retract the tapping tool and resume the cycle, or completely abort the tapping operation. Refer to your system installers documentation for details.

(G84.1): Right-hand Solid Tapping Cycle

The format for the G84.1 cycle is:

```
G84.1X__Z__R__F__L__Q__D__S__;
```

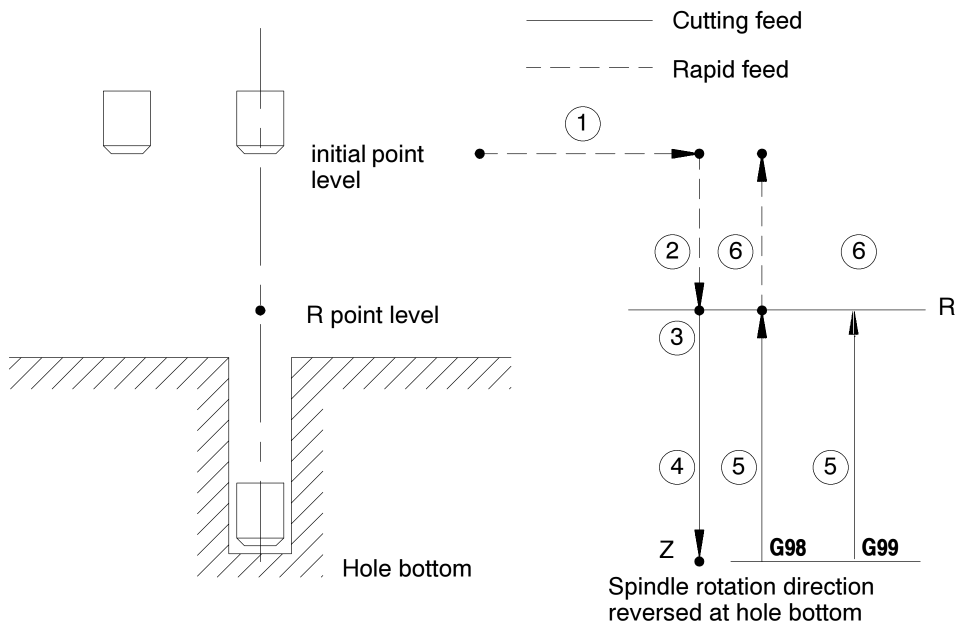
Where :	Is :
X	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	represents the thread lead along the drilling axis (Z in this manual). It is mandatory and modal in any subsequent solid tapping cycle blocks until a new F-word is programmed. The control interprets the F-word as the thread lead in inches per revolution or millimeters per revolution, depending on the inch/metric mode active.
L	defines the number of times the drilling cycle is repeated.
Q	defines the angle at which to orient the spindle before starting the tap. It is modal in any subsequent solid tapping blocks until a new Q-word is programmed or the tapping cycle is cancelled by a G80. To retap a hole, a Q-word must have been programmed when the hole was originally tapped.
D	defines the return spindle speed, but cannot exceed the maximum tapping spindle speed set in AMP. This will adjust your Z-axis feedrate according to the thread lead defined in F.
S	defines spindle speed in rpm.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: When programming and executing a G84.1 solid tapping cycle, remember:

- the feedrate of the tapping phases is derived as `(spindle {RPM} * F-lead {IPR}) = IPM`
- the spindle speed that is active at the start of the cycle determines the effective Z feedrate
- the direction of spindle rotation for tap-in and tap-out phases will be automatically generated by the control
- spindle speed override has no effect on the solid tapping cycle; you can use feedrate override to adjust the tapping operation
- D cannot exceed the maximum tapping spindle speed set in AMP
- you cannot select tools via D-word while in solid tapping mode
- gear changes are locked out
- cycle stop is acknowledged throughout the cycle, but can be disabled by G63
- you can use active reset to abort the cycle after the cycle stop request has been acknowledged
- to retap a hole, a Q-word must have been programmed when the hole was originally tapped
- block retrace is possible during the tap-in portion of the cycle, but not during the tap-out

Figure 25.12
G84.1: Right-hand Solid Tapping Cycle



In the G84.1 right-hand solid-tapping cycle, the control moves the axes in this manner:

1. The tool rapids to the tapping position above the hole location.
2. The threading tool then rapids to the R point.
3. The control either orients or stops the spindle.

If a Q-word was programmed:	the control:
yes	orients the spindle
no	stops the spindle

4. *Tap-in:* The clockwise rotation of the spindle ramps up to the programmed `S` spindle speed and linear motion of the Z axis moves synchronously to reach the Z position.
5. *Tap-out:* The spindle and linear motion reverse to the counterclockwise direction and retract to the R point.

   The tap-out speed is determined by `F * S` unless you programmed `D` (tap-out rpm), in which case tap-out speed is `F * D`.

   At the R point, spindle rotation has ramped to zero.
6. With G98 active, the cutting tool then accelerates to the rapid feedrate and retracts to the initial point level.

   With G99 active, the cutting tool remains at R point; no movement occurs.

In single-block mode, the control stops axis motion after phases 1, 2, 3, and 6 of the cycle (Figure 27.M2).

Important: When it is active, S-Curve Acc/Dec (G47.1) will be applied to the rapid feedrate portions of the solid-tapping cycle.

(G85): Boring Cycle, No Dwell/Feed Out

The format for the G85 cycle is as follows:

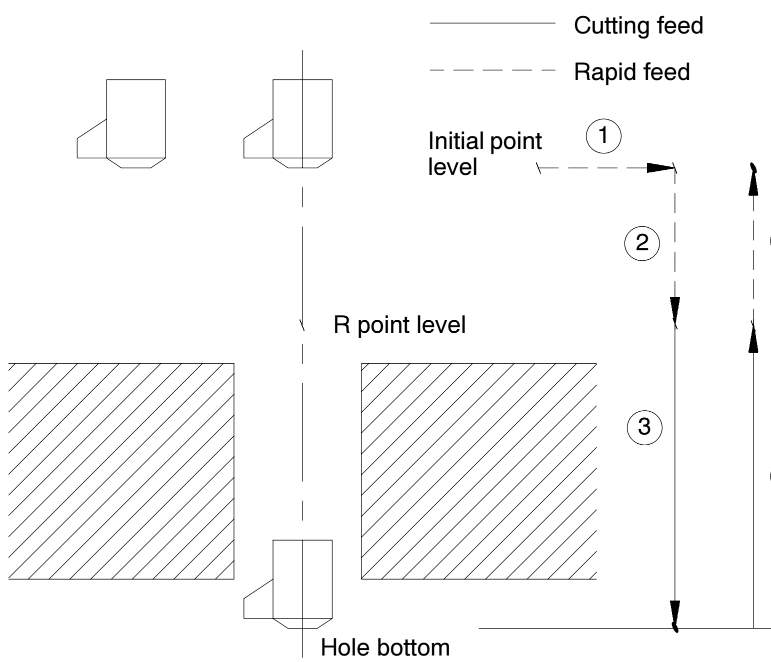
```
G85X__Y__Z__R__F__L__;
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.13
G85: Boring Cycle (Without Dwell, Feed Out)



In the G85 right-hand tapping cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool continues to drill at the programmed feedrate until it reaches the depth of the hole as programmed with the Z-word.
4. The control retracts the boring tool at the **cutting** feedrate to the R point.
5. The control retracts the drilling tool at a rapid feedrate to the initial point level, as determined by G98.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 5.

(G86): Boring Cycle, Spindle Stop/Rapid Out

The format for the G86 cycle is as follows:

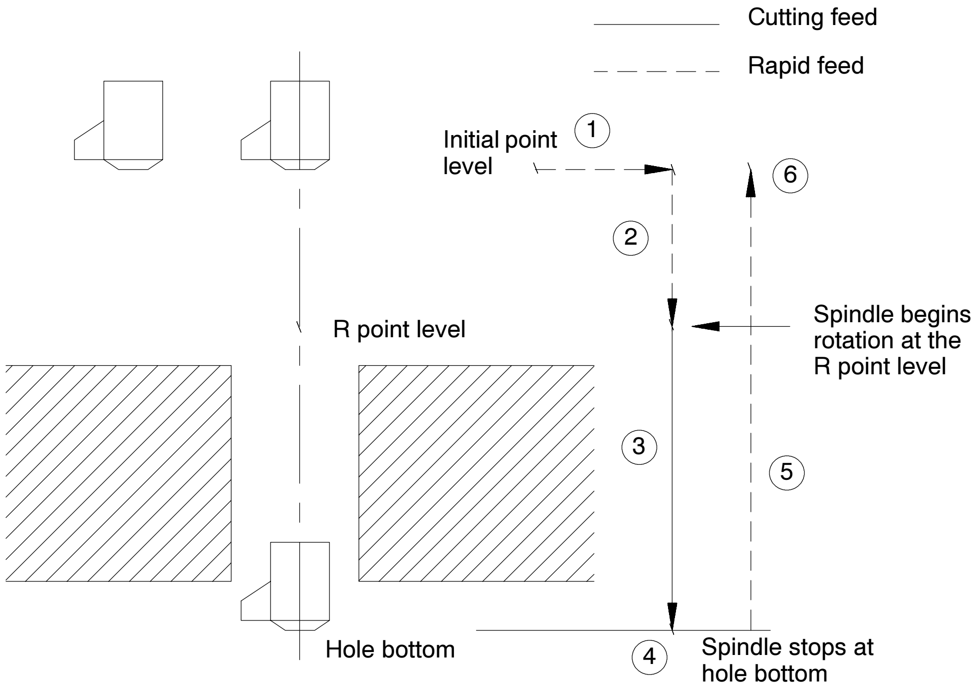
```
G86X__Y__Z__R__P__F__L__;
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.14
G86: Boring Cycle, Spindle Stop/Rapid Out



In the G86 milling fixed cycle, the control moves the axis in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The cutting tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The cutting tool bores at the programmed feedrate until it reaches the depth of the hole as programmed with the Z-word.
4. If the user has entered a value for the P parameter, the cutting tool will dwell after it reaches the bottom of the hole.
5. The spindle stops rotating.
6. The boring tool is then retracted at a rapid feedrate to the initial point level, as determined by G98. Spindle rotation continues forward.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 6.

(G87): Back Boring Cycle

The format for the G87 back boring cycle is:

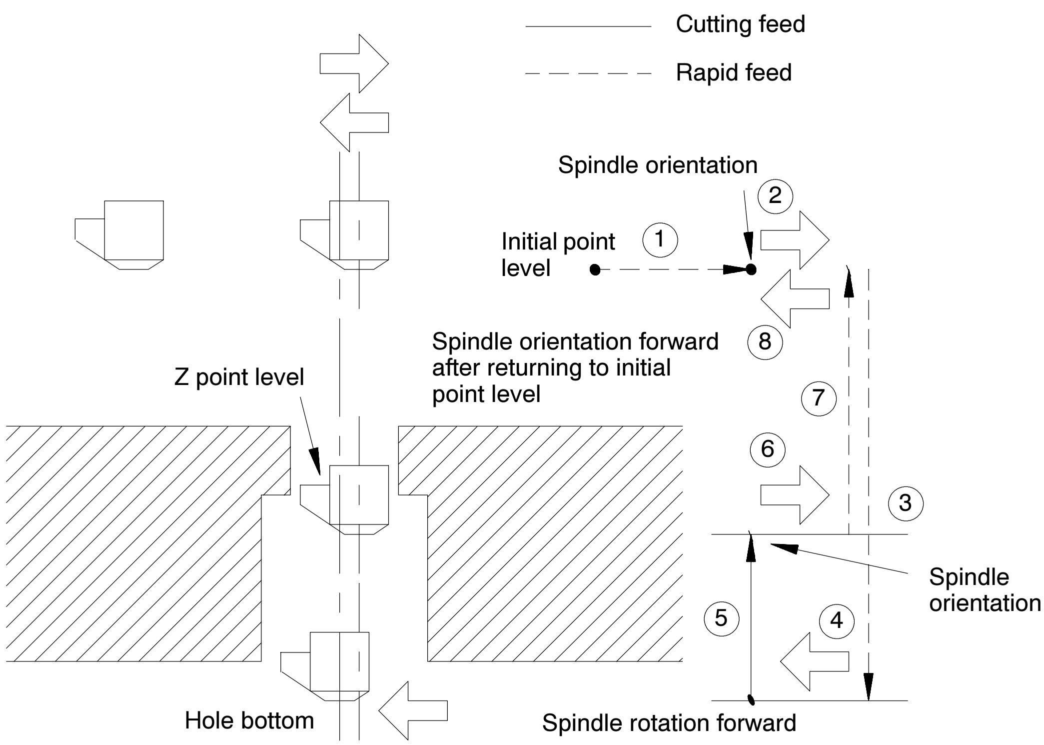
```
G87X__Y__Z__{I__J__K__}R__F__L__;
                Q__
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the Z point level. The Z point level in this case is the **top** of the hole that is being cut by the back boring operation.
Q or I, J, K	defines the tool shift amount and direction.
R	defines the position beyond the hole bottom so the tool can safely shift.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: This cycle requires an existing hole through which the back boring tool can be safely lowered prior to the back boring operation.

Figure 25.15
G87: Back Boring Cycle



In the G87 back boring cycle, the control moves the axes in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. After the back boring tool is positioned, the control orients the tool to a position determined in AMP by the system installer.

   The control shifts the boring tool by one of two possible methods, as discussed below. The shift method is determined in AMP by the system installer. Refer to the documentation supplied by your system installer for additional information.

Method I

This shift method is a single axis shift. The direction and axis for the shift is set in AMP by the system installer or can be altered using the milling fixed cycle parameter table (refer to page 25-34).

- The direction of the axis is specified as + or -.
- The feedrate using this shift method is the programmed feedrate.
- The Q-word shift amount is always interpreted as a positive value. A negative Q-word is not allowed.

Method II

The direction of the shift using this method is programmed in the boring cycle block. Program a shift amount for axes in the current plane only (determined by G17, G18, or G19) using the following words:

I__ programs an X axis move.
J__ programs a Y axis move.
K__ programs a Z axis move.

Follow the I-, J-, and K-words (modal during milling fixed cycles) with incremental values in the block that programs the hole position.

When using Method II, remember:

- If both axes in the current plane are to be shifted, specify both words to move the axes.
- The move generated will be a single linear move and will execute at rapid traverse.

3. The back boring tool moves at a rapid feedrate through the existing hole to the depth designated by the R-word.

4. Once the designated depth is reached, the back boring tool shifts the same distance but in the opposite direction as the previous shift (the shift made in step 2).

   After this shift, the programmer or operator must start spindle rotation. The spindle must rotate in the clockwise direction.

5. The control retracts the back boring tool at the cutting feedrate to a level specified by the Z-word.

6. After reaching the Z depth, the spindle rotation stops so that the control can re-orient the back boring tool to the position specified in AMP.

   The back boring tool is shifted a third time, in the same manner as in step 2, so that it is again "off-center" and can be removed through the existing hole.

7. The back boring tool moves at a rapid feedrate to the initial point level regardless of whether G98 or G99 are active.

8. The back boring tool is shifted a fourth time, in the same manner as in step 2, returning to the initial X, Y coordinates of the hole location.

(G88): Boring Cycle Spindle Stop/Manual Out

The format for the G88 cycle is:

```
G88X__Y__Z__R__P__F__L__;
```

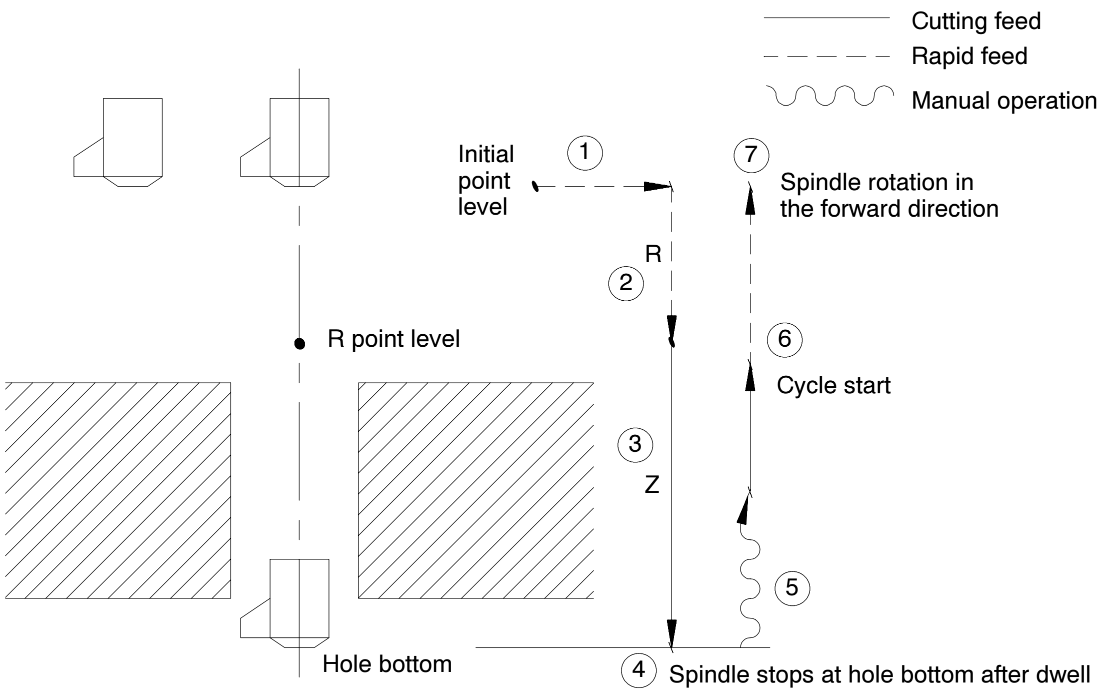
Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at the hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Important: Setting the BW_EOBSTP (End-of-Block Stop Request) flag to true during a G88 fixed cycle causes the cycle to be executed in its entirety without user intervention. When this flag is set to true, the dwell occurs at the bottom of the hole, the spindle stops, and the control automatically retracts the Z axis.

Figure 25.16
G88: Boring Cycle, Spindle Stop/Manually Out



In the G88 boring cycle, the control moves the axis in the following manner:

1. The tool rapids to the initial point level above the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool bores at the programmed feedrate until it reaches the depth specified with the Z-word.
4. If the user has entered a value for the P parameter, the boring tool will dwell after it reaches the bottom of the hole.
5. After the tool reaches the Z depth, the spindle stops revolving. At this point, the operator must perform a manual retraction of the drilling axis as described in chapter 4. (Press <CYCLE START> to return the control to automatic mode.)
6. The boring tool is then retracted at a rapid feedrate to initial point level, as determined by G98.
7. At this point, the rotation of the spindle changes to the clockwise direction.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 5.

(G89): Boring Cycle Dwell/Feed Out

The operations in G89 are identical to as those of the G85 boring cycle with the exception that the control executes a dwell at hole bottom.

The format for the G89 cycle is as follows:

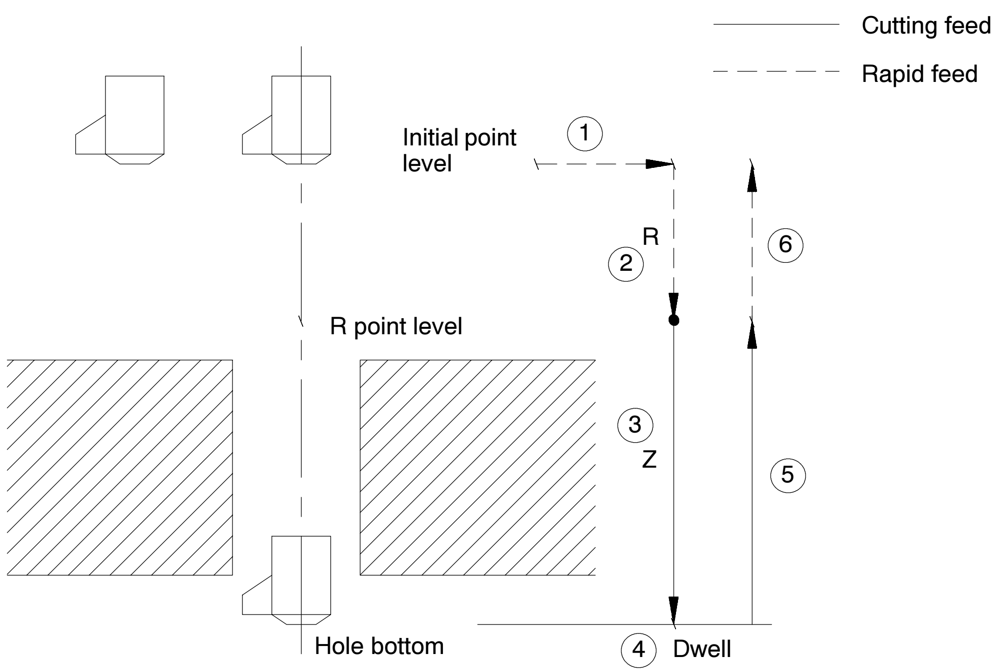
```
G89X__Y__Z__R__P__F__L__;
```

Where :	Is :
X,Y	specifies location of the hole.
Z	defines the hole bottom.
R	defines the R point level.
P	defines the dwell period at hole bottom.
F	defines the cutting feedrate.
L	defines the number of times the milling fixed cycle is repeated.

(Refer to page 25-6 for a detailed description of these parameters.)

Important: The programmer or operator must start spindle rotation.

Figure 25.17
G89: Boring Cycle, Dwell/Feed Out



In the G89 boring cycle, the control moves the axes in the following manner:

1. The tool rapids to initial point level above the hole location.
2. The boring tool then rapids to the R point level, slows to the programmed cutting feedrate and begins the boring operation.
3. The boring tool bores at the programmed feedrate until it reaches the depth of the hole specified by the Z-word.
4. If the user has entered a value for the P parameter, the boring tool will dwell after it reaches the bottom of the hole.
5. The control retracts the boring tool at the cutting feedrate to the R point level.
6. The boring tool accelerates to the rapid feedrate and retracts to the initial point level.

When the **single block** function is active, the control stops axis motion after steps 1, 2 and 5.

Altering Milling Fixed Cycle Operating Parameters

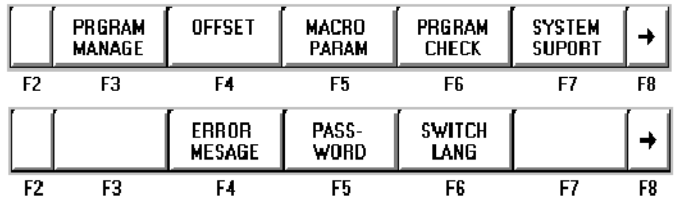
The system installer determines many parameter for the milling fixed cycles in AMP. The following 3 parameters are set in AMP but may be overridden by the operator using the Milling Cycle Parameter screen. When changed through this screen, the new values remain in effect until they are manually changed or AMP is downloaded with new values.

- **G73 Deep Hole Peck Drilling Cycle retract amount -** This parameter determines the value of “d”. “d” for this cycle is the distance above the last infeed step that the control retracts the tool from the part, normally to clear chips. See the section on G73 Deep Hole Peck Drilling cycle for details on this cycle’s operation.
- **Clearance Amount for Cycles** - This parameter also determines the value of “d”. The amount “d” for this cycle is the distance between the end of the tool and the plane of the uncut part. See the section on G83 Deep Hole Drilling for details on this cycles operation.
- **G76 / G87 Fine/Back Boring Cycles Shift Axis -** This parameter determines the axis that the shift amount programmed with “Q” will be on. Note that a shift in either axis, in either direction (positive or negative) for the currently active plane can be selected.This parameter is ignored if the shift direction is programmed in the block using I-, J-, or K-words.

To alter these 3 parameters, follow these steps:

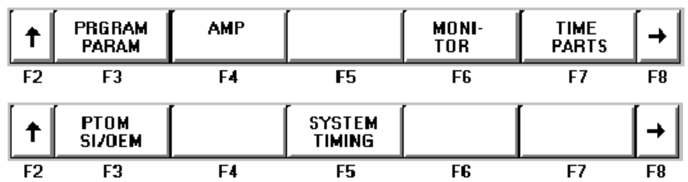
1. Press the **{SYSTEM SUPORT}** softkey.

(softkey level 1)



2. Press the **{PRGRAM PARAM}** softkey.

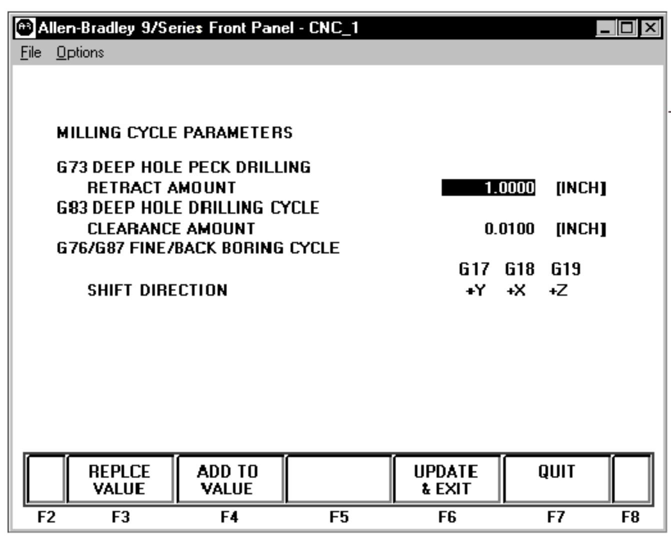
(softkey level 2)



3. Press the **{MILCYC PARAM}** softkey. The Milling Cycle Parameter screen is displayed. Figure 25.18 shows a typical Milling Cycle Parameter screen.

(softkey level 3)

Figure 25.18
Milling Cycle Parameter Screen



4. Use the up and down cursor keys to select the parameter you intend to change. The screen shows the selected parameter in reverse video.

   Once a parameter is selected, you can either replace the parameter value, or add to the parameter value:

- To **replace** the current value of the parameter with a new value, key in the new value on the input line of the screen and press the **{REPLCE VALUE}** softkey. The old value will be replaced with the new value just keyed in.
- To **add** an amount to the current value of the parameter, key in the amount to add to the current parameter value on the input line of the screen and press the **{ADD TO VALUE}** softkey. The value just keyed in is then added to the old value for the selected parameter.

5. Replace the parameter value or add to it.

There are two ways to quit the Milling Cycle Parameter screen:

- To **save** the changes just made to the parameters and leave the Milling Cycle Parameter screen, press the **{UPDATE & EXIT}** softkey.
- To **discard** any changes just made to the parameters and leave the Milling Cycle Parameter screen, press the **{QUIT}** softkey.

(softkey level 3)

6. If you want to quit the Milling Cycle Parameter screen, save or discard any changes made.

Examples of Drilling Cycles

The following are example programs and an illustration of G83, deep hole drilling cycle. Example 25.2 is in incremental mode, Example 25.3 is in absolute. Figure 25.19 illustrates the result for both programs individually.

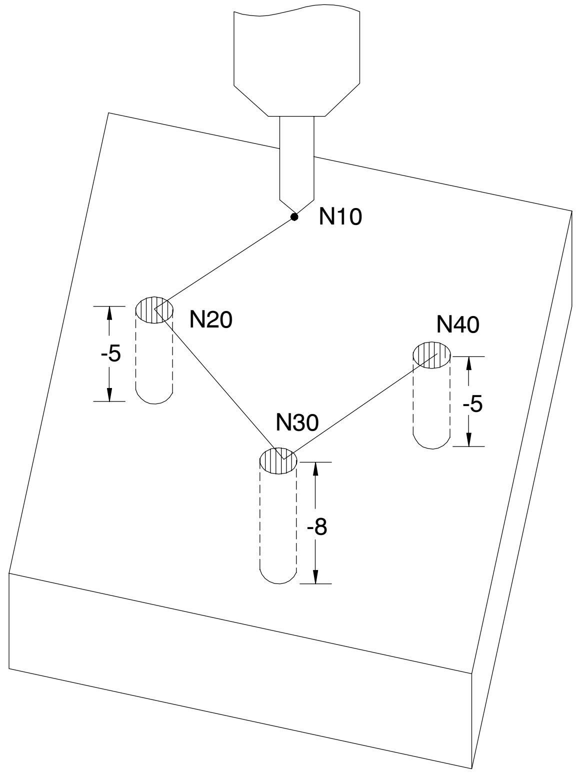
Example 25.2
Programming G83, Deep Hole Drilling Cycle in Incremental Mode

```
N10     G90 G00 X5 Y12;
N20     G91 G83 X-4 Y-2 Z-3 R-2 Q1.5;
N30     X4 Y-5 Z-5;
N40     X4 Y5 Z-3;
N50     M30;
```

Example 25.3
Programming G83, Deep Hole Drilling Cycle in Absolute Mode

```
N10     G90 G00 X5 Y12 Z0 G17 F200;
N20     G83 X1 Y10 Z-5 R-2 Q1.5;
N30     X5 Y5 Z-8;
N40     X9 Y10 Z-5;
N50     M30;
```

Figure 25.19
Result of Example 25.2 and Example 25.3



END OF CHAPTER

Skip and Gauge Probing Cycles

Chapter Overview

This chapter describes the external skip, gauge, and probe functions available on the control. Use this table to find information:

This chapter describes theses major topics in detail.

External Skip, Gauge, and Probe Functions

Use skip, gauge and probe for various automatic measurements, and also for interfacing automatically to external mechanical devices.

These functions are available:

- External Skip (G31-G31.4) -- These functions move an axis on a linear path until the control receives an external signal through the logic program. The block is aborted, the control records the axis coordinate as a paramacro parameter, and program execution skips on to the next block
- Tool Gauging (G37 - G37.4) -- These function identically to the external skip except that the axis coordinates (at the time the external signal is received) can be used to modify the tool offset table.
- Hole Probing (G38) -- Use this probing cycle to directly measure the diameter and center point location of a hole in a part.
- Parallel Probing (G38.1) -- Use this probing cycle to measure the amount that a part is out of parallel with a specified axis.

Important: The G04 dwell feature may also be enabled as an external skip or tool gauging command. For details about G04 refer to chapter 13.

The control provides several means of triggering an external skip, gauge, or probing block:

- Discrete inputs on the I/O ring
- Any one of the four available "High Speed inputs"
- A "Probe" input that directly latches the feedback counters.

These different inputs, each with different degrees of precision, may be used to signal the control to store the current axes positions. Refer to documentation prepared by your system installer for details on your specific machine.

These conditions must be satisfied when an external skip, gauge, or probe bock is executed:

- Cutter compensation must be disabled (G40 mode) when the block is executed.
- The block that contains the external skip G-code (excluding G04 as external skip) must be a linear block.

ATTENTION: We do not recommend using a skip block from any fixed cycle block (such as drilling or pocket milling). If you do choose to execute a skip block in a fixed cycle mode, be aware that the block that is skipped when the trigger occurs can be a cycle generated block. If this is the case the cycle will continue normal execution skipping only the portion of the cycle that was executing when the trigger occurred. If the generated block skipped is a crucial portion of the cycle, damage to the part or machine tool can occur.

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 logic. When the program block is terminated any remaining axis motion generated by the block that has not been performed remains unexecuted (other non-motion commands are still performed). The control continues normal program execution at the beginning of the next block following the skipped block.

The external skip function is controlled by G31, G31.1, G31.2, G31.3, and G31.4. The system installer determines what signal (such as a touch probe, manual switch, etc.) corresponds to each G31 code in logic. The system installer can choose different signals to correspond to G31.2, G31.3, and G31.4. G31 and G31.1 are functionally the same, always using the same external signal and the same AMP defined feedrate. With proper logic programming, a G04 dwell in seconds may also be used as an external skip function.

The format for any G31 external skip blocks is:

```
G31 X__ Y__ Z__ F__;
```

Where :	Is :
G31	Any of the G-codes in the G31 series or G04. Use the one that is configured to respond to the current external skip signal device that is being used.
X, Y, Z	The endpoint of the move if no external skip signal is received. These also determine the direction that the tool 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 produce linear motion regardless of the current mode active at their execution. After their completion the control returns to the operating mode active before the external skip block was read (G00, G01, G02, G03).

Important: The move that immediately follows a G31 series external skip block cannot be a circular move.

The coordinates of the axes when the external skip signal is received are available as the paramacro system parameters #5061-#5066 (work coordinate system) and #5071-#5076 (machine coordinate system). These values will have been adjusted to compensate for the probe tip radius if a radius compensation value was entered.

For example, assume you have entered a probe tip radius of .01. It is triggered as axis 2 approaches in the positive direction at the axis 2 coordinate of 1.1200. The value available for paramacro parameter #5072 would be 1.1300

Probe tip radius is defined by the system installer in AMP. This value may also be changed through the paramacro system parameter #5096.

Refer to chapter 27 for details on paramacro parameters.

Skip Function Application Example

One typical application for these G-codes would be moving the part until it contacts a probe and then proceeding with a machining operation from that point. This would provide part feature consistency by insuring that the machining of all parts began from a fixed reference point (probe trigger point).

Note that for this application, the probe tip radius would not be significant, and should be entered as zero if the skip signal position paramacros are to be used.

Another typical application for these G-codes would be to mount the probe as if it were a tool. When the probe contacts the part and triggers, coordinate data would be available in the paramacros for use in the remainder of the part program.

Note the probe tip radius **is** significant for this application.

Tool Gauging External Skip Functions (G37 codes)

Tool gauging functions are similar to external skip functions. The key difference is that the tool gauging cycles use the actual tool position (when the external skip signal is received) to enter values in the tool offset table for the currently active offset.

Use tool gauging functions to terminate the execution of motion commands in a block and modify offset tables when the control receives a signal through logic. When the program block is terminated any remaining axis motion generated by the block that has not been performed remains unexecuted (other non-motion commands are still performed). The current tool position is stored, and the control continues program execution at the beginning of the next block following the skipped block.

The gauging function is controlled by G37, G37.1, G37.2, G37.3, and G37.4. The system installer determines what signal (such as a touch probe, manual switch, etc.) corresponds to each G37 code in logic. The system installer can choose different signals to correspond to G37, G37.1 G37.2, G37.3, and G37.4. G37 and G37.1 are functionally the same, always using the same external signal and the same AMP-defined feedrate.

The format for any G37 skip blocks is:

```
G37 Z__ F__;
```

Where :	Is :
G37	Corresponds to any of the G-codes in the G37 series. Use the one that is configured to respond to the current skip signal device that is being used.
X, Y, Z	The axis on which the length offset measurement is to be taken is specified here as either X or Z. Only one axis may be specified in a G37 block. The numeric value following the axis name corresponds to the exact coordinate at which the skip signal is expected to occur. This value is a signed value (+ or -) and determines the initial direction of travel.
F	The tool gauging external skip function feedrate. If no value is entered here, the external skip function executes at either the currently active feedrate or at 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: The G37 series G-codes cannot be used to modify the tool tip radius values. Only the tool length offset values can be modified.

The target offset value for these gauging operations is determined by the currently active offset number (active D word for tool diameter offset, active H-word for tool length offset). Note that for length offset measurement, not only must the correct H word be active but the correct offset must be on (G43, or G44). If the tool diameter offset value is to be modified, then the following conditions must be true:

- a tool diameter offset (D word) must be active though diameter compensation (G41, G42) does not need to be on.
- for most tools to get an accurate measurement for tool diameter, the tool must be oriented so that the measured diameter is the outside edge of the tool (not a flute or other geometric feature that would affect the tool diameter).

ATTENTION: If modifying a tool length offset, the offset value generated with this gauging operation is immediately loaded into the offset table. Since this offset must be the currently active offset, it becomes effective either immediately when the next block is executed or delayed until the next block that contains motion on the tool length axis is executed (when an offset is activated is determined in AMP by the system installer).

The G37 series of G-codes always produce linear motion regardless of the current mode active at their execution. After their completion, the control returns to the operating mode active before the skip block was read (G00, G01, G02, G03).

The system installer determines (in AMP) a position tolerance for the G37 functions. This tolerance defines a legal range before and after the coordinate position programmed with the axis word in the G37 block.

If the skip signal is received before the tool enters **or** after the tool exits the position tolerance range, a PROBE ERROR occurs. This error appears on the screen as a warning but does not place the control in E-Stop. Instead the G37 block is aborted and program execution proceeds to the next block. No modification of the tool offset table is performed.

Important: The move that immediately follows a G37 series skip block cannot be a circular move.

The system installer determines in AMP if the new value is added to or replaces the old value in the table. The system installer also determines in AMP what gauge cycles alter which tool offset tables, geometry, or wear.

The control automatically compensates for probe radius and length when calculating tool 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 will have been adjusted to compensate for the probe tip radius and the probe length if radius and length compensation values were entered.

For example, assume you have entered a probe tip radius of .01. It is triggered as axis 2 approaches in the positive direction at the axis 2 coordinate of 1.1200. The value available for paramacro parameter #5072 would be 1.1300

Probe tip radius and probe length are defined by the system installer in AMP. These values may also be changed through the paramacro system parameters #5096 (for radius) and #5095 (for length).

Refer to chapter 27 for details on paramacro parameters.

Tool Gauging Application Example

A typical application for these G-codes in determining tool length offsets executes as follows:

1. When the control executes the G37 block, the tool is moved towards the triggering device using the axis specified in the block.
2. When the control receives the appropriate skip signal through logic, axis motion stops.
3. The control records the position when the skip signal is received. It determines the difference by subtracting the position specified with the axis word in the G37 block from this position. The difference is then added to or replaces the value in the appropriate geometry or wear table for the currently active tool offset number.

Figure 26.1 Typical Tool Gauging Configurations

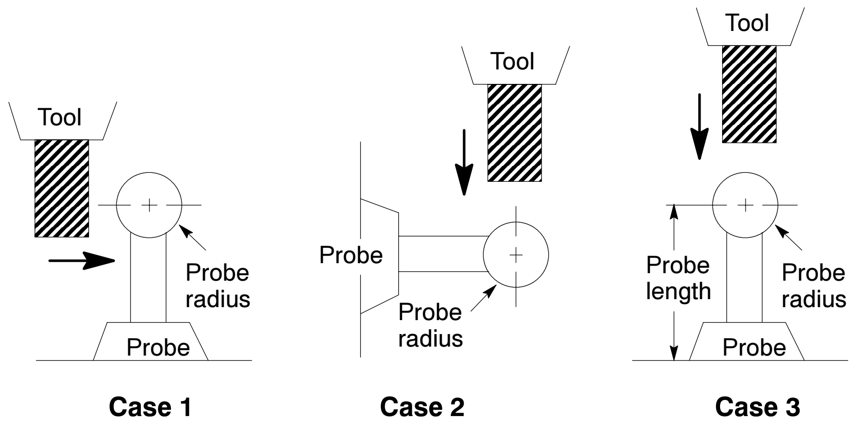


Figure 26.1 illustrates 3 typical tool gauging configurations. All 3 cases assume that the probe is at a known, fixed point on the machine.

In Case 1, the Z axis tool offset length is being gauged, while in Case 2, the tool length is being gauged. In both of these cases, only the probe tip radius is significant to the control in calculating the offset adjustment.

In Case 3, the X axis tool offset length 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 tool gauging configuration Case 3 depicted in Figure 26.1 due to the risk of probe damage.

Hole Probing (G38)

The purpose of this cycle is to provide a means to measure the actual radius and/or locate the center of a hole in a part or gauge using a touch probe.

To use the G38 cycle, the currently active plane when the G38 is programmed must be the same plane that the hole to be measured is in (refer to chapter 12 for information on plane selection). For example, to measure a hole that is cut in the XY plane, the G38 code must be programmed with the XY plane active.

Format for the G38 code is as follows:

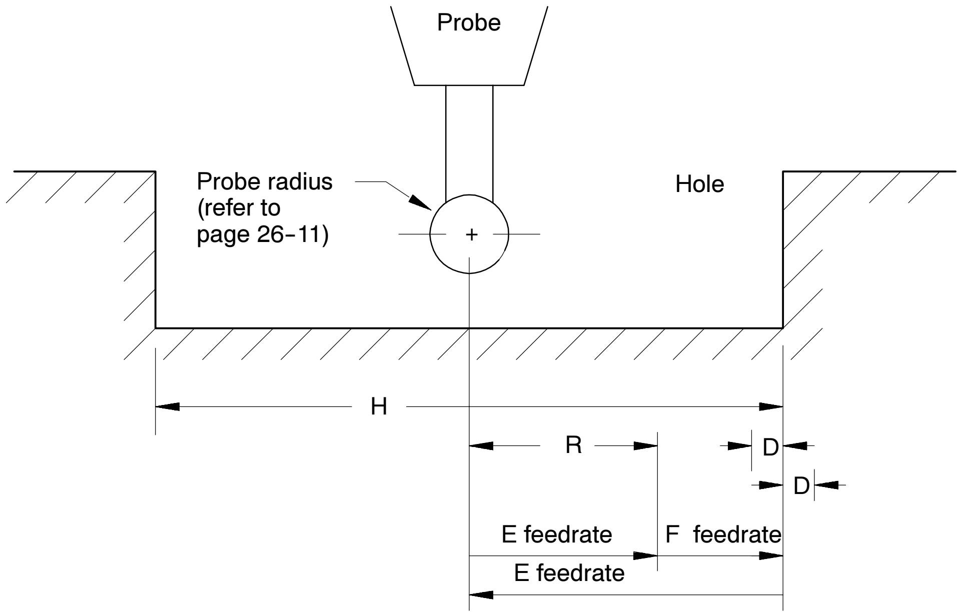
```
G38 H__ R__ D__ E__ F__;
```

Where :	Is :
H	The estimated or expected diameter of the hole. This parameter is optional. If H is not programmed, the control will generate an H value that is equal to 2(R+D), where R is the approach distance and D is the tolerance band. Note that if R+D is greater than one half H, the control will ignore the programmed H value and use 2(R+D) as the new expected diameter.
R	The incremental unsigned approach distance. Enter the distance from the start-point of the probing cycle to a point that it is desirable for the feedrate to be slowed. At this point, the feedrate will slow from the approach feedrate (E) to the probing feedrate (F). This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed on page 26-14.
D	The tolerance band distance. The value entered for D defines a band on each side of the expected diameter entered with the H parameter. Enter a value defining a tolerance distance on either side of the expected probe triggering point. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed on page 26-14.
E	The approach feedrate. Enter a value for this parameter that defines the feedrate at which the probe is to approach the position specified by the R parameter. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed on page 26-14.
F	The probe feedrate. Enter a value for this parameter that defines the feedrate at which the probe is to move after passing the point defined by the R parameter. The probe continues on at this feedrate until contact has been made with the diameter of the hole or until the tolerance band is exceeded. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed on page 26-14.

Parameters R, D, E, and F can be entered in three ways:

- The system installer may have entered them in AMP, in which case they will always be available and need not be programmed in the G38 block. Refer to the documentation provided by your system installer.
- They may be entered or changed through the probing parameters table described on page 26-14. If entered in the table, they need not be programmed in the G38 block. The table value will supersede any values entered in AMP.
- They may be programmed directly in the G38 block. Values entered for these parameters in the G38 block supercede both AMP values and probe parameter table values.

Figure 26.2 Parameters for G38 Hole Probing Cycle



When the G38 code is executed, the probe must have previously been positioned with the probe in the hole so that it will contact the sides of the hole when axis motion takes place in the currently active plane. Ideally (to maximize accuracy) the probe should be positioned near the center of the hole.

The control executes the G38 cycle in the following manner:

1. When the G38 block is read during program execution, the control starts axis motion in the positive direction of the second axis defining the plane at the approach feedrate (E). If the plane is the XY plane, then this would be the Y axis.

2. The axis continues on at the approach feedrate (E) until the approach distance (R) is reached. Then the feedrate is reduced to the probe feedrate (F).

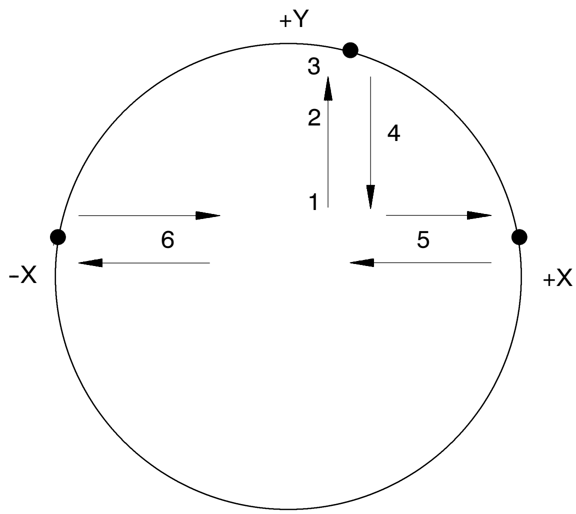
3. The axis continues towards the estimated diameter (H) until the probe signals that contact has been made. If the probe triggers before reaching the negative tolerance band (D), or does not trigger after passing through the positive tolerance band (D), a PROBE ERROR will occur. This error appears on the screen as a warning but does not place the control in E-Stop. Instead the G38 block is aborted, and program execution proceeds to the next block.

4. If the probe triggers within the tolerance band, the position is recorded, and the axis returns to the start-point of the probing operation at the approach feedrate (E).

5. The control repeats the preceding steps on the first axis in the current plane. If the plane is the XY plane, then this would be the X axis.

6. After successfully probing in the positive direction of the first axis in the current plane, the control will then probe in the negative direction of the first axis. In all, three points are measured for determining the circle's radius.

Figure 26.3Typical Probe Path During G38 Hole Probing



The control calculates the actual radius and center position of the hole from the three data points just measured.

After the probing moves are completed and the hole center location has been calculated, the axes are positioned at the approach feedrate (E) to the exact hole center location.

Important: To accurately measure a hole radius and determine its center, the exact probe tip radius must be available to the control. This value is entered either through AMP, through paramacro system parameter #5096, or through the probe parameter table discussed on page 26-14.

Table 26.A shows the paramacro system parameters used to relay information from the probing operation to the programmer. Refer to the paramacro chapter for details on these system parameters.

Table 26.A Paramacro System Parameters for G38 Hole Probing

System Parameter	Value this parameter holds
#5092	G38 hole center coordinate on first axis of active plane
#5093	G38 hole center coordinate on second axis of active plane
#5094	G38 radius of hole

An easy way to view these parameters is to copy them to a common parameter that may be accessed on a paramacro screen; for example, the program block:

```
#500=#5094;
```

would take the value of the cycles calculated hole radius and copy it to common parameter 500 that may be viewed in the macro table **{COM 2A PARAM}**. Note that the **{COM 2A PARAM}** parameters also allow a parameter description to be entered in the table, such as "HOLE RADIUS VALUE".

Parallel Probing Cycle (G38.1)

The purpose of this cycle is to provide a means to measure the amount that a part is out of parallel (or rotated) with a selected axis through the use of a touch probe. Note that the currently active plane (G17, G18, or G19) must be the same plane in which probe motion is to occur in and must be active before the probing cycle block is executed.

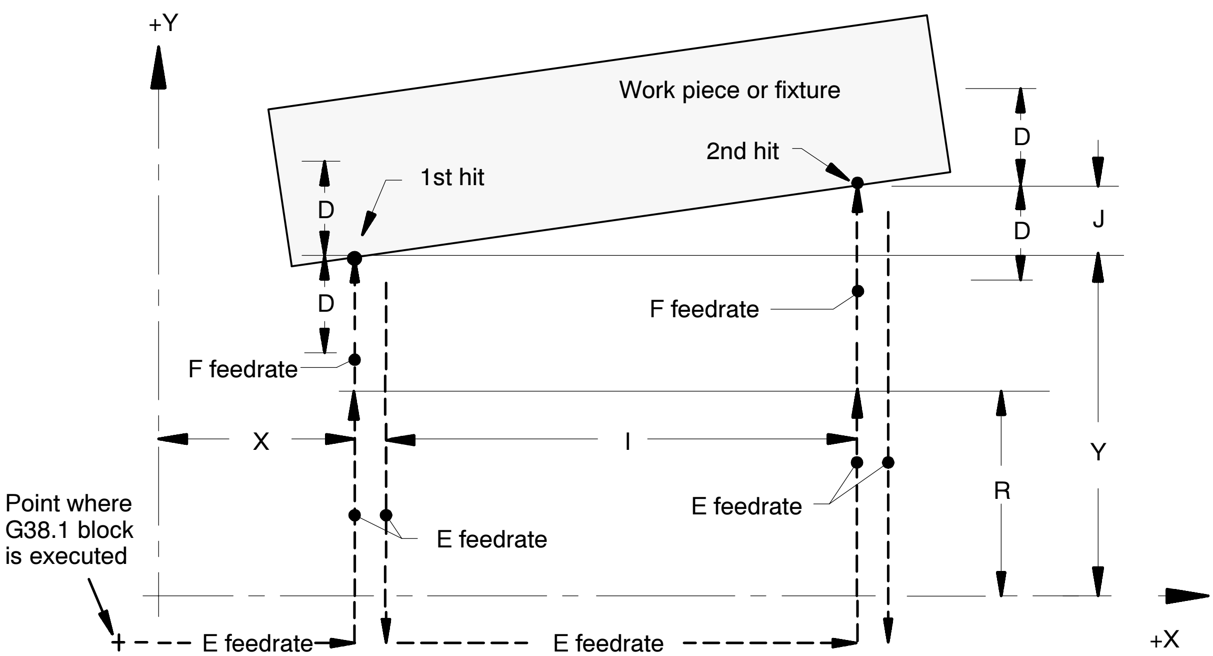
Format for the G38.1 code is as follows:

```
G38.1 X__ Y__ I__ J__ R__ D__ E__ F__;
```

Where :	Is :
X	Any valid axis name followed by the coordinate position of the first measuring point on that axis. May be an absolute or incremental, signed value. Being the first axis word in the G38.1 block indicates that this axis is the one from which measurements are to be taken. The G38.1 probing cycle will determine how much the part is out of parallel with this axis
Y	The name of any axis that is perpendicular to the first axis in the G38.1 block. May be an absolute or incremental, signed value. Parallelism will be measured by moving the probe along this axis to the edge of the part. The value entered with this parameter defines the expected position on this axis where the probe will hit the edge of the part.
I (integrand of first axis in G38.1 block)	The incremental signed distance between the first and second probe hits. This incremental distance is measured along the first axis programmed in the G38.1 block. In this manual I is the integrand word for the X axis. The integrand word for an axis is determined in AMP.
J (integrand of second axis in G38.1 block)	The estimated amount the part is out of parallel. J is an incremental, signed distance. J is added to the coordinate value entered with the second axis in the G38.1 block for the second probe hit only. The net result is to shift the tolerance band (programmed with the D word) by the amount J.

R	The incremental unsigned approach distance. This parameter determines the distance the second axis in the G38.1 block travels at the E feedrate when probing towards the part. After this distance is reached the probe slows to the F feedrate. This parameter is optional. If not entered, the control will default to the value entered in the probing table discussed on page 26-14..
D	The tolerance band distance. The value entered for D defines a band on both sides of the expected endpoint entered with the Y parameter. Enter a value for this parameter defining a tolerance distance on either side of the expected probe triggering point (Y above). This parameter is optional, but must have a positive value if programmed. If not programmed, the control will default to the value entered in the probing cycle parameter table discussed on page 26-14.
E	The approach feedrate. Enter a value for this parameter that defines the feedrate used to reach the approach distance (R). This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed on page 26-14.
F	The probe feedrate. Enter a value for this parameter that defines the feedrate at which the probe is to move after passing the point defined by the R parameter. The probe continues on at this feedrate until contact has been made with the edge of the part or the tolerance band is exceeded. This parameter is optional. If not entered, the control will default to the value entered in the probing cycle parameter table discussed on page 26-14.

Figure 26.4 Parameters and Motion Paths for G38.1 Probing Cycle



Parameters R, D, E, and F can be entered in 3 ways:

- The system installer may have entered them in AMP, in which case they will always be available and need not be programmed in the G38.1 block. Refer to the documentation provided by your system installer.

- They may be entered or changed through the probing parameters table described on page 26-14. If entered in the table, they need not be programmed in the G38.1 block. The table value will supersede any values entered in AMP.
- They may be programmed directly in the G38.1 block. Values entered for these parameters in the G38.1 block supercede both AMP values and probe parameter table values.

The control executes the G38.1 cycle in this manner:

1. When the G38.1 block is executed, the control initially moves only the first axis in the G38.1 block to the coordinate position entered with it. The approach feedrate (E) is used for this move.
2. The second axis in the G38.1 block is then moved to the coordinate defined by the approach distance parameter, R. Again, the approach feedrate (E) is used.
3. The feedrate is then reduced to the probe feedrate (F). The second axis continues to move into the tolerance band (D) until the probe triggers, signaling that contact has been made. If the probe triggers before reaching the negative tolerance band (D) or does not trigger after passing through the positive tolerance band (D), a PROBE ERROR will occur. This error appears on the screen as a warning but does not place the control in E-Stop. Instead the G38.1 block is aborted, and program execution proceeds to the next block.
4. If the probe triggers within the tolerance band, the position is recorded. The axis then reverses direction and retracts the distance traveled in steps 2 and 3 above at the approach feedrate (E).
5. The first axis in the G38.1 block moves the incremental distance entered with the I parameter. The approach feedrate (E) is used for this move.
6. Steps 2, 3, and 4 above are repeated. Note that if the J parameter is programmed, the value of J is added to the Y parameter for the second probe hit. This will shift the location of the tolerance band programmed with the D-word.

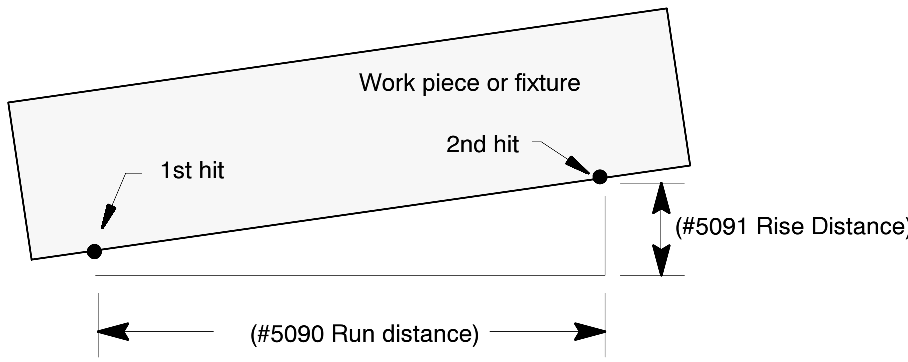
At this point, the control calculates the amount that the workpiece or fixture is out of parallel (with the first axis in the G38.1 block) using the two recorded positions. Note that the probe tip radius is not a factor in G38.1 parallel probing.

26.B and 26.5 show the values generated by this cycle and stored by the control as paramacro system parameters. These values may be viewed using the paramacro tables as discussed in chapter 27.

Table 26.B Paramacro System Parameters for G38.1 Parallel Probing.

System Parameter	Value this parameter holds
#5090	G38.1 run measurement (always equal to I)
#5091	G38.1 rise measurement

Figure 26.5 G38.1 Parallel Probing Cycle Paramacro Parameter Values



Probing Parameters Table

Use this feature to access the Probe Parameters table and alter probe parameters affecting the operation of the G31, G37, and G38 codes . For details on the parameters available here, refer to the appropriate section of this chapter.

Access to this table may be restricted. Refer to chapter 2 regarding access control.

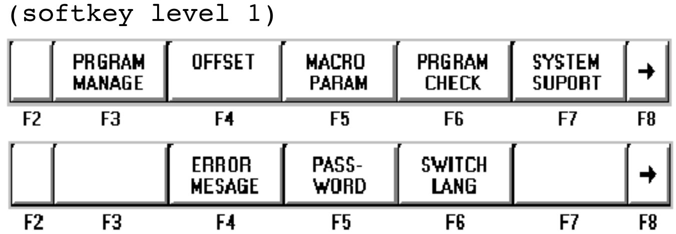
The parameters available for modification through this table may have already been defined in AMP. Refer to the documentation provided by your system installer.

Values entered here will supersede the corresponding AMP value. They will remain active even if the control is powered down.

To display or alter the values in the probing parameters table follow the steps below:

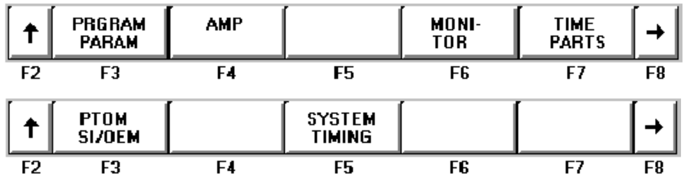
1. Press the **{SYSTEM SUPORT}** softkey.

(softkey level 1)



2. Press the **{PROGRAM PARAM}** softkey.

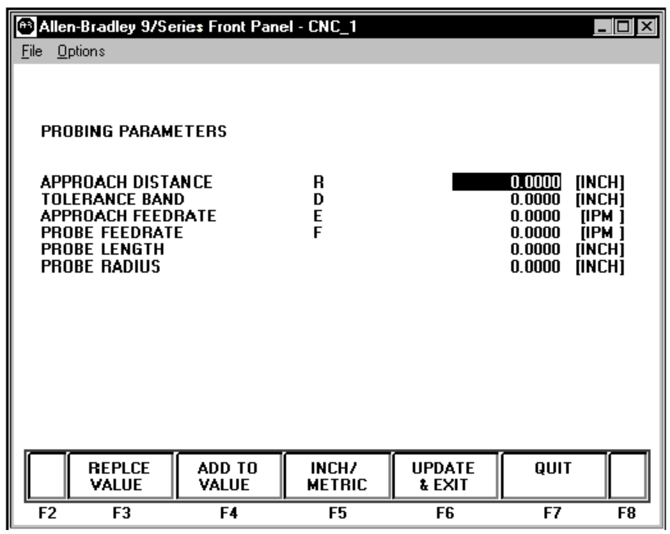
(softkey level 2)



3. Press the **{PROBE PARAM}** softkey to display the probing cycle parameter table.

(softkey level 3)

Figure 26.6 Probing Parameters Screen

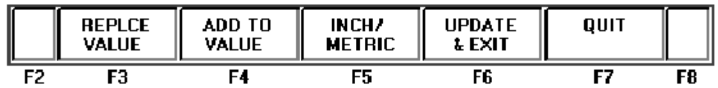


4. Use the up, or down cursor keys to move the block cursor to the probing parameter to be changed. The selected parameter will be shown in reverse video.

5. You can change parameter values two ways:

   - Press the **{REPLCE VALUE}** softkey then type in a new value for the selected parameter by using the keys on the operator panel. When you press the **[ENTER]** key, the value typed in will replace the old value for that parameter.
   - Press the **{ADD TO VALUE}** softkey then type in a value to add to the old value for the selected parameter by using the keys on the operator panel. When you press the **[ENTER]** key, the value typed in will be added to the old value for that parameter.

(softkey level 4)



6. Exit the probing parameters screen in one of two ways:

If you want to:	Press this softkey:
save your changes and exit	{UPDATE & EXIT}
lose your changes and exit	{QUIT}

END OF CHAPTER

Paramacros

Chapter Overview

The Paramacro™ feature is similar to a subprogram with many added features. Special features available with a paramacro include:

- computable variables
- computable word address fields in any block type
- variable to and from logic
- access to certain modal system parameters for computations
- arithmetic operators and expressions for computations
- conditional branching, subroutines, and subprogram calls based upon logical function results
- parameter programs, subroutines, and subprograms
- parameter autocycles
- user-definable prompts to aid in program generation and execution
- prompting of parameters for main program execution
- calculator function under prompt edit

All of these features are valid in any block within a main program, subprogram, or paramacro program. Most are permitted in an MDI program unless otherwise stated; the only restriction being that no other program commands, except other paramacro commands, may exist in a block that contains paramacro commands. Macro and nonmacro commands may not exist in the same program block.

This chapter includes these major topics:

Parametric Expressions

It may be necessary for mathematical expressions to be evaluated in a complex paramacro. This requires that some form of mathematical equation be written in a paramacro block. This section describes the operators and function commands available for use on the 9/PC control. These operators and function commands are valid in any block within a program, subprogram, paramacro, or MDI program.

Basic Mathematical Operators

This subsection lists the basic mathematical operators that are available on the 9/PC controller. Use these operators to accomplish mathematical operations that are necessary to evaluate the basic mathematical equation such as addition, multiplication, etc. Table 27.A lists the basic operators and their meanings.

Table 27.A
Mathematical Operators

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
[ ]	Brackets
OR	Logical OR
XOR	Logical Exclusive OR
AND	Logical AND
MOD	Modulus

The control executes a mathematical operation in this order:

1. Any part of the expression that is between the brackets [] is evaluated first.
2. Multiplication, division, and MOD are evaluated second.
3. All other operations are evaluated third.

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

Example 27.1
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.

If B is negative in the above format, 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 27.2
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

Mathematical Function Commands

This subsection lists the basic mathematical functions that are available on the 9/PC controller and their use. Use these functions to accomplish mathematical operations that are necessary to evaluate the trigonometric and other complex mathematical equation such as rounding off, square roots, logarithms, exponent, etc. Table 27.B lists the basic functions that are available and their meanings.

Table 27.B
Mathematical Functions

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

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

The functions in Table 27.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 27.3
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 27.4
Mathematical Function Examples

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

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

Example 27.5
Precaution for Order of Operation

`N1#1=1.6;`	Parameter #1 is set at 1.6
`N2#2=2.8;`	Parameter #2 is set at 2.8
`N3#3=ROUND[#1+#2];`	Parameter #3 is set at 4.0

The values composing parameter #3 are added together first and then rounded, not rounded and then added together.

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

Transfer of Control Commands

Use transfer of control commands to alter the normal flow of program execution. Normally the 9/PC control executes program blocks sequentially. By using control commands, the programmer can alter this normal flow of execution and transfer execution to a specific block or begin looping (executing the same set of blocks repetitively).

Important: Transfer of control commands call a block by its N number. If more than one N number exists in a block, the control uses only the left-most N number in that block. If the same N number is used for more than one block, the control uses the first block it encounters with the correct N number (the control searches in the forward direction first, then starts at the top of the program).

Two types of transfer of control commands are available:

- Conditional -- The execution of a jump or loop is dependant on whether a mathematical condition is true.
- Nonconditional -- The execution of a jump or loop is always performed when that block is executed.

Conditional Operators

This section describes conditional operators that are available for paramacro programming. A conditional operator causes a comparison between two values and yields a result of true or false. Use conditional operators in "IF" or "WHILE" commands as described on pages 27-8 and 27-9.

Use the true or false condition to determine whether the "IF" or "WHILE" blocks are executed. Table 27.C lists the conditional operators available for paramacro programming:

Table 27.C
Conditional Operators

Operator	Condition Tested
EQ	Equal
NE	Not Equal
GT	Greater Than
LT	Less Than
GE	Greater Than or Equal
LE	Less Than or Equal

Program a condition between the [and] brackets in this format:

[A EQ B]

where A and B represent some numerical value. The values for A and B can be in the form of some mathematical equation or in the form of a paramacro parameter.

Example 27.6
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 27-8 on "IF" statements and 27-9 on "WHILE" statements. For details on the use of paramacro parameters, see page 27-11.

GOTO and IF-GOTO Commands

Unconditional GOTO

Any time the control executes a GOTO block, the unconditional GOTO command automatically transfers control.

Use this format for the GOTO command:

```
GOTO n;
```

Where:	Is:
n	Execution is transferred to the block with the sequence number specified as n any time that the GOTO block is executed.

Example 27.7
Unconditional GOTO

```
N1...;
N2...;
N3GOTO5;
N4...;
N5...;
N6...;
/N7GOTO1;
```

In Example 27.7, execution continues sequentially until block N3 is read; then execution transfers to block N5 and again resumes sequential execution to block N6. If optional block skip 1 is off, block N7 transfers execution back to block N1.

Conditional IF-GOTO

The conditional IF-GOTO command is dependant on whether a mathematical condition is true. If this condition is true, execution transfers to the block specified.

Use this format for the IF-GOTO command:

```
IF [(condition)] GOTO n;
```

Where:	Is:
(condition)	some mathematical condition (refer to page 27-7). This condition is tested by the control to determine if it is true or false.
n	if the condition is tested as true, execution is transferred to the block specified as n

If the condition is tested as false, execution falls through the block and the GOTO is not executed. Program execution continues in a normal fashion.

Example 27.8
Conditional IF

```
N1...;
N2IF[#3EQ-1.5]GOTO5;
N3...;
N4...;
N5...;
N6IF[#4LT3]GOTO1;
N7...;
```

When block N2 is read, parameter #3 is compared to the value -1.5. If the comparison is true, then blocks N3 and N4 are skipped, and execution continues on from block N5. If the comparison is false, then execution continues to block N3. When block N6 is read, parameter #4 is compared to the value 3. If the comparison is true, then execution is transferred to block N1; if it is false, execution continues to block N7.

DO-END and WHILE-DO-END Commands

Unconditional DO-END

The unconditional DO-END command is rarely used. The lack of a condition here causes the control to loop indefinitely until reset or **<CYCLE STOP>** is pressed, or until some other transfer of control command forces execution out of the loop.

The format for the unconditional DO-END command is as follows:

```
DO m;
:
:
:
END m;
```

Where:	Is:
m	a loop identifier used to relate a DO block with an END block. The value of m must be the same for the DO as it is for the corresponding END. This value can be either 1, 2, or 3.

All blocks between the DO and the END command are executed indefinitely or until execution is stopped by some external operation such as by pressing **<E-STOP>** or **<CYCLE STOP>**, or when a block delete is performed if programmed.

Conditional WHILE-DO-END

The conditional WHILE-DO-END command is dependant on whether a mathematical condition is true. If this condition is false, execution transfers to the block immediately following the END statement block.

The following options are not available with OCI:

- software front panel
- standard 9/PC editor
- online search monitor utility
- tool path graphics
- encryption mode
- encryption setup

Use this format for the WHILE-DO-END command:

```
WHILE [ (condition) ] DO m;
;
;
;
END m;
```

Where:	Is:
`(condition)`	some mathematical condition (refer to page 27-7). This condition is tested by the control to determine if it is true or false.
`m`	an identifier used by the control to relate a DO block with an END block. The value of m must be the same for the DO as it is for the corresponding END. This value can be either 1, 2, or 3.

All blocks between the DO and the END command are executed until the condition is tested as false. This set of blocks is referred to as a WHILE-DO-END program segment.

When the condition for the WHILE-DO block is tested as false, execution is then transferred to the block immediately following the END statement block.

Example 27.9
WHILE-DO-END Program Segment

```
N1 #1=1;
N2WHILE[#1LT10]DO1
N3#1=[#1+1];
N4...;
N5...;
N6END1;
N7...;
```

In Example 27.9, blocks N2 through N6 are executed nine times. At that time, the condition in block N2 becomes false, and program execution is transferred to block N7.

Nesting is possible with a WHILE-DO-END command. Nesting is defined here as one WHILE-DO-END program segment executing within another WHILE-DO-END program segment. WHILE-DO-END nesting is limited to three independent segments at one time.

Example 27.10
Nested WHILE DO Commands

```
N1#1=1;
N2WHILE[#1LT10]DO1;
N3#1=[#1+1];
N4WHILE[#1EQ2]DO2;
N5...;
N6END2;
N7END1;
N8...;
```

In Example 27.10, blocks N2 through N7 are repeated until the condition in block N2 becomes false. Within DO loop 1, DO loop 2 repeats until the condition in block N4 becomes false.

Parameter Assignments

These subsections describe assigning paramacro parameter values and how these parameters are used in a paramacro. Use parameters for paramacros to replace a numeric value. They are used as a variable.

Five types of parameters can be called for use in a paramacro:

- local - independent set of variables assigned to each nested macro
- common - variables available to all programs
- system - variables that indicate specific system condition
- logic - provide variables shared between part programs and logic

The following sections describe these parameters independently. This in no way means that they are not interchangeable in the same macro program. Mixing the different types of parameters in the same paramacro is acceptable.

Local Parameter Assignments

Local parameters are #1 - #33. There are five 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 (four levels of nesting maximum).

Assigned parameter values are specific to the individual macro nesting levels. Local parameters are assigned as described on page 27-28.

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, uses 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 initialized when the control executes a end of program block (M99, M02, or M30).

Considerations for Local Parameters

When assigning values to local parameters, remember:

- All local variable assignments are reset to zero any time the control reads an M02, M99, 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 27.H (B) on page 27-29 for the parameter assignment.

Example 27.11
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 27.12
Assigning the Same Parameter Twice

```
G65P1001R3.1A2R-0.5
```

The above block sets the following parameters:

parameter #1 = 2.0 As set by the A-word

parameter #18 = -0.5 As set by the last R-word.

The 1st value of 3.1, assigned to parameter #18 by the R-word, is replaced by the 2nd value set by the second R-word.

Example 27.13
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 1st value of 5, assigned to parameter #7 by the D-word, is replaced by the 2nd value set by the second I-word.

Common Parameters

The common parameters refer to parameter numbers 100 to 199 and 500 to 999. The common parameters are assigned through the use of a common parameter table as described on page 27-28.

Common parameters are global in nature. This means that the same set of parameters can be called by any program, macro, subprogram, or MDI program.

Common parameters are divided in to two types: saved or unsaved.

- Saved common parameters refers to the common parameters that retain their value even after power to the control is lost. Saved common parameters are parameter numbers 500 - 999.
- Unsaved common parameters refers to the common parameters that do not retain their value after power to the control is lost. When power to the control is turned back on, these parameters reset their value to zero. Unsaved common parameters are numbers 100 - 199.

The logic 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 BW_PRMQTY and BW_PRMERR in your 9/PC Logic Reference Manual (8520-9.2).

System Parameters

System parameters may be used by any part program, including paramacros and subprograms. All of these parameters may be used as data or may be changed by assignment (read and write) unless indicated differently in Table 27.D.

These system parameters are generated by the control and can be modified by operation or programming. They correspond to different control conditions such as current operating modes, offsets, etc.

Table 27.D lists the system parameters available on the 9/PC control.

Table 27.D
System Parameters

Parameter #	System Parameter	Page
2001 to 2999	Tool Offset Tables	27-16
3000	[2] Program Stop With Message (logic)	27-16
3001	System Timer (logic)	27-17
3002	System Clock	27-17
3003	[2] Block Execution Control 1	27-17
3004	[2] Block Execution Control 2	27-18
3006	[2] Program Stop With Message	27-18
3007	[1] Mirror Image	27-19
4001 to 4120	[1] Modal Information	27-19
5001 to 5008	[1] Coordinates of End Point	27-20
5021 to 5028	[1] Coordinates of Commanded Position	27-20
5041 to 5048	[1] Machine Coordinate Position	27-20
5061 to 5068	[1] Skip Signal Position (Work Coordinate)	27-21
5071 to 5078	[1] Skip Signal Position (Machine Coordinates)	27-21
5081 to 5088	[1] Active Tool Length Offsets	27-21
5095 to 5096	Probe stylus length and radius	27-22
5101 to 5108	[1] Current Following Error	27-22
5201 to 5208	External Offset amount	27-22
5221 to 5226	G54 Work Coordinate Table Value	27-23
5241 to 5246	G55 Work Coordinate Table Value	
5261 to 5266	G56 Work Coordinate Table Value	
5281 to 5286	G57 Work Coordinate Table Value	
5301 to 5306	G58 Work Coordinate Table Value	
5321 to 5326	G59 Work Coordinate Table Value	
5341 to 5346	G59.1 Work Coordinate Table Value	
5361 to 5366	G59.2 Work Coordinate Table Value	
5381 to 5386	G59.3 Work Coordinate Table Value	
5630	[1] S-Curve Time per Block	27-24
5661 to 5638	[1] Acceleration Ramps for Linear Acc/Dec Mode	27-24
5651 to 5658	[1] Deceleration Ramps for Linear Acc/Dec Mode	27-24
5671 to 5678	[1] Acceleration Ramps for S-Curve Acc/Dec Mode	27-25
5691 to 5798	[1] Deceleration Ramps for S-Curve Acc/Dec Mode	27-25
5711 to 5718	[1] Jerk	27-25

[1] These parameters may only have their value received (read-only)
[2] These parameters may only have their value changed (write only)

#2001 to 8801
Tool Offset Tables

These parameters may be changed or simply read through programming. The values for these parameters are received or entered into the tool offset tables for geometry and wear (described in chapter 3). Table 27.E gives the parameter numbers associated with each table value.

Table 27.E
Tool Offset Table Parameters

	Offset Number	Parameter # for Geometry Table	Parameter # for Wear Table
Tool Length (Axis 1)	1 to 99	#2701 to 2799	#2001 to 2099
Tool Length (Axis 2)	1 to 99	#2801 to 2899	#2101 to 2199
Tool Length (Axis 3)	1 to 99	#8501 to 8599	#8101 to 8199
Tool Length (Axis 4)	1 to 99	#8601 to 8699	#8201 to 8299
Tool Length (Axis 5)	1 to 99	#8701 to 8799	#8301 to 8399
Tool Length (Axis 6)	1 to 99	#8801 to 8899	#8401 to 8499
Tool Radius	1 to 99	#2901 to 2999	#2201 to 2299
Tool Orientation	1 to 99	#2301 to 2399	N/A

#3000
Program Stop With Message (Logic)

Use this parameter to cause a cycle stop operation and display a message on line 1 of the CRT. Any block that assigns any nonzero value to parameter 3000 results in a cycle stop. The actual value assigned to parameter 3000 is not used. Parameter 3000 is a write only parameter.

When the control executes this block, a cycle stop is performed and the message “SEE PART PROGRAM FOR MACRO STOP MESSAGE” is displayed on line 1 of the CRT. This is intended to point out to the operator an important comment in the program block that assigns a value to parameter 3000 (refer to chapter 9 on comment blocks).

For example, programming

```
#3000=.1 (TOOL NUMBER 6 IS WORN);
```

causes program execution to stop at the beginning of this block and displays a message telling the operator to read the comment in the block. A block reset must be performed before a cycle start resumes normal program execution.

When this block is executed, it also sets the paramacro alarm logic flag (BR_MCALRM) true. Refer to the system installer's documentation for details on the effect of this logic flag.

#3001
System Timer (Logic)

This parameter is referred to as the timer parameter. It is a read-write parameter. Every 20ms a value of 20 is added to the value of parameter 3001. The value of this parameter is also stored by a logic flag ($PM20MS) and may be modified or set by the system installers logic program. Refer to the system installer's documentation for details on the use of this timer. The maximum value of this parameter is 32768ms. Any value greater than 32768 causes this parameter to "roll over" to zero and restart counting again. The value of this parameter is reset to zero every time power is lost.

#3002
System Clock

This parameter is referred to as a clock parameter and references an hour counter. It is a read-write parameter with negative value assignments being illegal. The maximum value for this parameter is 1 year (8760 hours). The parameter value is maintained when power is lost. It is incremented by .000005556 every 20 ms.

#3003
Block Execution Control 1

Use this parameter to control whether the control ignores single-block mode and to control when M-codes are executed in a block. The value of this parameter ranges from 0 to 3, and it is a write only parameter.

These results occur when parameter 3003 is set to the corresponding values:

Value:	Single-block mode:	M-codes are executed:
0	can be activated	at the beginning of the program blocks execution
1	requests are ignored	

Value:	Single-block mode:	M-codes are executed:
2	can be activated	after the complete execution of the other commands in the block
3	requests are ignored	

#3004
Block Execution Control 2

This parameter determines whether a cycle stop request 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 27.F shows the results of the different values for parameter number 3004. If they are ignored, the control does not let you use the feature. If they are recognized, you can activate the feature in the normal manner.

Table 27.F
Parameter 3004 Values

Value of Parameter	Cycle Stop	Feedrate Override	Exact Stop Mode
0	Recognized	Recognized	Recognized
1	Ignored	Recognized	Recognized
2	Recognized	Ignored	Recognized
3	Ignored	Ignored	Recognized
4	Recognized	Recognized	Ignored
5	Ignored	Recognized	Ignored
6	Recognized	Ignored	Ignored
7	Ignored	Ignored	Ignored

#3006
Program Stop With Message

Use this parameter to cause a cycle stop operation and display a message on line 1 of the CRT. Any block that assigns a new value to the parameter 3006 results in a cycle stop. Any decimal value may be assigned to this parameter the value of which is not used.

When the control executes this block, a cycle stop is performed and the message “SEE (MESSAGE) IN PART PROGRAM BLOCK” is displayed on line 1 of the CRT. This is intended to point out to the operator an important comment in a program block (refer to chapter 9 on comment blocks). This parameter is a write only.

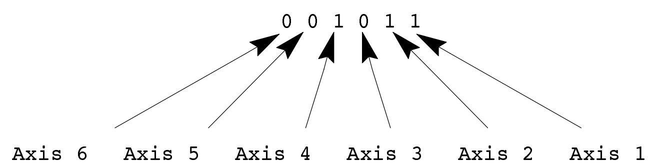
For example, programming:

```
#3006=.1 (Install Tool Number 6);
```

causes program execution to stop at the beginning of this block and displays a message telling the operator to read the comment in the block.

#3007
Mirror Image

This parameter is a read-only. It generates an integer that represents, in binary, what axes are mirrored. For example, if the value of this parameter was 3, the binary equivalent for this is 000011. The first digit of this binary equivalent (reading right to left) corresponds to axis 1, the second is axis 2, the third is axis 3, etc., up to the number of axes configured in your system. When a zero is in the binary location for an axis, it indicates that the axis is not mirrored. If a 1 is in that axis location, that axis is mirrored. For example, if the parameter #3007 is the integer 11 (binary 001011), it indicates axes 1, 2, and 4 are mirrored and axes 3, 5, and 6 are not mirrored.



Where:

0 indicates axis is not mirrored

1 indicates axis is mirrored

This parameter reflects both the programmed and front-panel (external mirror) status of mirroring on the axes.

#4001 to 4120
Modal Information

These are read-only parameters. They indicate the value of a modal program word. Table 27.G shows the modal program word that applies to the given parameter number.

Table 27.G
Modal Data Parameters

Parameter Number	Modal Data Value
4001 to 4025	These correspond to the different G-code Groups 1-25 (refer to chapter 9) and show what G-code from group is currently active.
4108	Current E-word value
4109	Current F-word value
4113	Most recently programmed M-code
4114	Most recently programmed N-word
4115	Current program number O-word
4119	Current S-word value
4120	Current T-word value

For example, if currently programming in G02 mode at a feedrate of 100, the parameters would be as follows:

G02 is a group 1 G-code, so its value of 02 is set to parameter number 4001.

The feedrate programmed with an F-word gives parameter number 4109 a value of 100.

#5001 to 5008
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	5005	Axis 5 coordinate position
5002	Axis 2 coordinate position	5006	Axis 6 coordinate position
5003	Axis 3 coordinate position	5007	Axis 7 coordinate position
5004	Axis 4 coordinate position	5008	Axis 8 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5021 to 5028
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	5025	Axis 5 coordinate position
5022	Axis 2 coordinate position	5026	Axis 6 coordinate position
5023	Axis 3 coordinate position	5027	Axis 7 coordinate position
5024	Axis 4 coordinate position	5028	Axis 8 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5041 to 5048
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	5045	Axis 5 coordinate position
5042	Axis 2 coordinate position	5046	Axis 6 coordinate position
5043	Axis 3 coordinate position	5047	Axis 7 coordinate position
5044	Axis 4 coordinate position	5048	Axis 8 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5061 to 5068
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 logic from a probe or other device such as a switch. These are the coordinates in the work coordinate system.

5061	Axis 1 coordinate position	5065	Axis 5 coordinate position
5062	Axis 2 coordinate position	5066	Axis 6 coordinate position
5063	Axis 3 coordinate position	5067	Axis 7 coordinate position
5064	Axis 4 coordinate position	5068	Axis 8 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5071 to 5078
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 logic 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	5075	Axis 5 coordinate position
5072	Axis 2 coordinate position	5076	Axis 6 coordinate position
5073	Axis 3 coordinate position	5077	Axis 7 coordinate position
5074	Axis 4 coordinate position	5078	Axis 8 coordinate position

The system installer determines in AMP the name (or word) that is used to define the axis.

#5081 to 5088
Active Tool Length Offsets

These are read-only parameters. They correspond to the currently active tool length offsets (refer to chapter 19).

5081	Current axis 1 tool length offset.	5085	Current axis 5 tool length offset.
5082	Current axis 2 tool length offset.	5086	Current axis 6 tool length offset.
5083	Current axis 3 tool length offset.	5087	Current axis 7 tool length offset.
5084	Current axis 4 tool length offset.	5088	Current axis 8 tool length offset.

#5095 to 5096
Probe stylus Length and Radius

These parameters correspond to the values set in the probing cycle parameter table discussed in chapter 26. When values are assigned to these parameters, the current values in the probe table is replaced.

5095	Probe stylus Length
5096	Probe stylus Radius

For details on probe radius and length parameters, refer to chapter 26 on tool gauging.

#5101 to 5108
Current Following Error

These parameters are read-only. They correspond to the current following error for an axis.

5101	Axis 1 following error	5105	Axis 5 following error
5102	Axis 2 following error	5106	Axis 6 following error
5103	Axis 3 following error	5107	Axis 7 following error
5104	Axis 4 following error	5108	Axis 8 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 5208
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 (refer to 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	5205	Axis 5 offset amount
5202	Axis 2 offset amount	5206	Axis 6 offset amount
5203	Axis 3 offset amount	5207	Axis 7 offset amount
5204	Axis 4 offset amount	5208	Axis 8 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 rehomed, or power to the control has been cycled.

#5221 to 5386
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 (refer to chapter 3). This lets you read data from the tables and also set data into the table by assigning values to the parameters.

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

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

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

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

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

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

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

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

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

The system installer determines in AMP the name (or word) that is used to define the axis.

#5630
S-Curve Time per Block

This parameter is read only. The value represents the amount of time (seconds converted to system scans) for a part program block's S-Curve filter where S-Curve Acc/Dec is applied during G47.1 mode. When it is multiplied by the scan time, the product equals the amount of time required by the acceleration.

This parameter is only calculated for blocks that have programmed motion with S-Curve Acc/Dec.

#5631 to 5638
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 17 in this manual.

5631	Axis 1 acceleration ramp	5635	Axis 5 acceleration ramp
5632	Axis 2 acceleration ramp	5636	Axis 6 acceleration ramp
5633	Axis 3 acceleration ramp	5637	Axis 7 acceleration ramp
5634	Axis 4 acceleration ramp	5638	Axis 8 acceleration ramp

#5651 to 5658
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 17 in this manual.

5651	Axis 1 deceleration ramp	5655	Axis 5 deceleration ramp
5652	Axis 2 deceleration ramp	5656	Axis 6 deceleration ramp
5653	Axis 3 deceleration ramp	5657	Axis 7 deceleration ramp
5654	Axis 4 deceleration ramp	5658	Axis 8 deceleration ramp

#5671 to 5678
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 17 in this manual.

5671	Axis 1 acceleration ramp	5675	Axis 5 acceleration ramp
5672	Axis 2 acceleration ramp	5676	Axis 6 acceleration ramp
5673	Axis 3 acceleration ramp	5677	Axis 7 acceleration ramp
5674	Axis 4 acceleration ramp	5678	Axis 8 acceleration ramp

#5691 to 5698
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 17 in this manual.

5691	Axis 1 deceleration ramp	5695	Axis 5 deceleration ramp
5692	Axis 2 deceleration ramp	5696	Axis 6 deceleration ramp
5693	Axis 3 deceleration ramp	5697	Axis 7 deceleration ramp
5694	Axis 4 deceleration ramp	5698	Axis 8 deceleration ramp

#5711 to 5718
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 17 in this manual.

5711	Axis 1 jerk	5715	Axis 5 jerk
5712	Axis 2 jerk	5716	Axis 6 jerk
5713	Axis 3 jerk	5717	Axis 7 jerk
5714	Axis 4 jerk	5718	Axis 8 jerk

Logic Parameters

Paramacro parameters are provided on the 9/PC control to allow a means of communicating values between the logic program and the part program. This is done by assigning values to specific paramacro parameters or logic flags. They are:

- Input parameters

Use these parameters to transfer data from logic to the part program

- Output parameters

Use these parameters to transfer data from the part program to logic. Some applications may, however, use the output flags transfer data from logic to the part program as needed.

Input Flags:

The paramacro input parameters available to the part programmer are:

These are 4-integer or 3-integer and 32-bit pattern input parameters available. The part program may only read the values assigned to these parameters; it may not write values to them. The paramacro input parameters available to the part programmer are:

- #1000 – #1031 and #1040 – #1071

  These paramacro logic parameters are used to display the binary equivalent of the integer assigned to #1032. #1000 is the first bit, #1001 is the second bit, #1002 is the third bit, and so forth up to parameter #1031 (which is the 32nd bit).

  The second set of parameters, #1040 – #1071, functions the same way.

- #1032 – #1035 and #1072 – #1075

  The control always interprets parameter #1032, #1033, #1034, and #1035 as integer values regardless of how they are assigned in logic (as an integer or on a per bit basis). #1032 is the only parameter that may also be interpreted by the control on a per-bit basis using parameters #1000 - #1031. Logic may always interpret these values on either a per-bit basis or as integer values.

  The second set of parameters, #1072 – #1075, functions the same way.

See the system installer's documentation for a detailed description of the use and operation of these input flags.

Output Flags:

Output flags function almost identically to input flag with one key difference. Where input flags may only be read by the part program, output flags may be both read and written to by the part program. Typically these are used only to output information to the logic program from the part program; however, if the available number of input flags is not sufficient for a given application, the Output flags may also be used to send information to the part program from logic.

Output flags should not be used as Input flags unless absolutely necessary. This is because the operator/programmer has the ability to inadvertently write data to the Output flags, whereas the Input flags cannot be written to from the control.

Output flags are broken into four 32-bit words. The part programmer can only assign or read the values of to these flags as integers with the exception of parameter #1032 which may be assigned as an integer or as a bit pattern. The paramacro output input parameters available to the part programmer are:

- #1100 – #1131 and #1140 – #1171

  When the values of these parameters are assigned in the part program, they should be assigned values of 1 or 0 (as bit patterns). If any integer value other than zero is assigned to these parameters, logic interprets it as a 1. These paramacro logic parameters are used to pass the binary equivalent of the integer assigned to #1132. #1100 is the first bit, #1101 is the second bit, #1102 is the third bit, and so forth up to parameter #1131 (which is the 32nd bit). When a value is assigned to #1132, the values assigned to #1100 - #1131 are overwritten with the binary equivalent of #1132.

  The second set of parameters, #1140 – #1171, functions the same way.

- #1132 – #1135 and #1172 – #1175

  The control always interprets these parameters as **integer** values. #1132 is the only parameter that may also be interpreted by the part program on a per-bit basis using parameters #1100 – #1131.

  The second set of parameters, #1172 – #1175, functions the same way.

See the system installer's documentation for a detailed description of the use and operation of these input flags.

Assigning Parameter Values

There are three methods for assigning parameters. They can be assigned by:

- using arguments (only available for local parameters)
- direct assignments
- using tables (view or set common parameters, view local parameters)

Assigning Parameters Using Arguments

Arguments may be used only to assign local parameter values. System, Common, and logic variables cannot be assigned by using arguments. Usually parameters assigned by 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 9/PC control provides five sets of local parameters. The first set of local parameters (those that apply to the main program and any subprogram call) may not be assigned using arguments. The second through fifth sets may be assigned by their association to given words in an argument statement located in a paramacro calling block. Table 27.H gives a listing of arguments and their corresponding parameter numbers.

These arguments assign values to the local parameters associated with the paramacro called in the same block.

Table 27.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 27.H (B) for the parameter assignment.

To enter a value for a parameter # using an argument, enter the word corresponding to the desired parameter number in a block that calls a paramacro (for legal argument locations, see specific formats for calling the macro) followed by the value to assign that parameter. For example:

```
G65P1001A1.1 B19;
```

assigns the value of:

1.1 to local parameter #1 in paramacro 1001

19 to local parameter #2 in paramacro 1001

You can specify arguments as any valid parametric expression. For example:

```
G246A#100B[#500+10.0]C[SIN[#101]];
```

Direct Assignment Through Programming

This assignment method applies to Local, Common, System, and logic parameters. You can perform direct assignment in Main, Macro, or MDI programs. Direct assignment is done by setting the parameter equal to some value in an equation using the " = " operator. For example, to assign a value of 2 to parameter number 100, simply enter the following program block:

```
#100=2;
```

The value to the left of the equals sign must contain the # sign followed by a legal parameter number. This parameter number may also take on the form of:

```
#parameter expression = parameter expression
```

Example 27.14
Calling Parameter Numbers

```
#6=1;
#144=1;
#[SIN[#6]]=1;
#[148/2]=1;
#[#6]=1;
```

All of the above can be used as legal parameter numbers. Any time that a different parameter is used between the [] symbols, the current value of that parameter is used for evaluation. For example:

```
#1=4;
#1=#1+2;
```

The net result of the above two blocks would be the assignment of a value of 6 to parameter #1.

Example 27.15
Assigning Parameters:

```
#100=1+1;
#100=5-3;
#100=#3;
#100=#7+1;
#100=#100+1;
```

You can also assign multiple paramacro parameters in a single block. In a multiple assignment block, each assignment is separated by a comma. For example:

```
#1=10,#100=ROUND[#2+#3],#500=10.0*5;
```

If you use multiple assignments in the same block, remember:

- You can enter as many assignments as can be typed into one block (127 characters maximum).
- For local and common parameters, block execution is from left to right. For example:

  ```
  #1 = 10,#2=#1+2;
  ```

  When executed, #1 is 10 and #2 is 12
- Once the first paramacro parameter assignment is made in a block, only assignment syntax is allowed in that block. You cannot program other information in that block, including programming a G-code. For example:

  ```
  #1 = 19.0,G1X10;
  ```

  gets the error message, "PARAMETER ASSIGNMENT SYNTAX ERROR"
- Only assign the same parameter a value once in each block. For example:

  ```
  #1=5,#2=4,#1=6;
  ```

  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.

Access the paramacro tables using the following steps:

1. Press the **{MACRO PARAM}** softkey.

(softkey level 1)

	PRGRAM MANAGE	OFFSET	MACRO PARAM	PRGRAM CHECK	SYSTEM SUPORT	→
F2	F3	F4	F5	F6	F7	F8

		ERROR MESAGE	PASS-WORD	SWITCH LANG		→
F2	F3	F4	F5	F6	F7	F8

2. Press the appropriate softkey for the table to be viewed. The macro parameters are separated in to 4 tables:

   - **{LOCAL PARAM}** softkey – Use this softkey to view the local parameters of the currently active program (unless the block look ahead has scanned an upcoming paramacro call). This table includes parameters numbered 1 - 33. Only one of the five available sets can be viewed on this screen at any one time. The local parameters reset to zero at the end of program command (M02 or M30).
   - **{COM-1 PARAM}** softkey – Use this softkey to view or edit the common parameters numbered 100 - 199 (available to any program, subprogram, or paramacro program). These common parameters lose their value and are reset to zero when power to the control is turned off.
   - **{COM-2A PARAM}** softkey – Use this softkey to view or edit the common parameters (available to any program, subprogram, or paramacro program) numbered 500 - 519. Their value is retained even when power to the control is cycled off. In addition to being 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 may not be used to call a common parameter in a program.
   - **{COM-2B PARAM}** softkey – Use this softkey to view or edit the common parameters. Their value is retained even when power to the control is cycled off. They differ from common 2A parameters in that they do not allow an additional name to be stored in the table with their values.

If viewing the local parameter table, do not continue to step 3. If editing one of the common tables, move on to step 3.

(softkey level 2)

↑	LOCAL PARAM	COM-1 PARAM	COM-2A PARAM	COM-2B PARAM		→
F2	F3	F4	F5	F6	F7	F8

3. Select a parameter to change by moving the cursor to the desired parameter number. The selected parameter is shown in reverse video. Move the cursor by an entire page by pressing the up or down cursor key while holding down the **[SHIFT]** key.

   Pressing the **{SEARCH NUMBER}** softkey allows a rapid search for the desired parameter number. To use this feature to position the cursor, press the **{SEARCH NUMBER}** softkey. Key in the desired parameter number and press the **[ENTER]** key. The entered parameter number is shown in reverse video.

4. Select and complete the appropriate step to alter the common parameter values. The three options include:

   - **To replace the current value of the selected parameter**, press the **{REPLCE VALUE}** softkey. Key in the new value for the parameter and press the **[ENTER]** key. The old value is replaced with the value just keyed in.
   - **To zero the current of a selected parameter**, press the **{ZERO VALUE}** softkey. The message "SELECT VALUE TO ZERO AND PRESS ENTER" appears. Select the parameter which is to be set to zero, and press the **[ENTER]** 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 control displays the prompt "OK TO ZERO ALL VALUES? (Y/N):" Press the Y character followed by the **[ENTER]** key to zero all parameter values in the table. Press the N character followed by the **[ENTER]** key to abort the operation.
     - While viewing one of the parameter screens during program execution, any changes to a parameter value on that screen that are made by the program are updated and displayed. By pressing the **{REFRSH SCREEN}** softkey, any parameters that have been changed by the program are updated to their current values.

5. If you pressed the **{COM-2A PARAM}** softkey (in step 2), additional softkeys are available to alter the parameter name. Select and complete the appropriate step to alter the common parameter names. The three options include:

   - **To edit an existing parameter name or enter a parameter name for the first time for a local parameter**, press the **{REPLCE NAME}** softkey. Key in a parameter name for the parameter. A name may be up to 8 characters long and include any alphanumeric character with the exception of a few of the special symbols. After the name is keyed in, press the **[ENTER]** key. The new parameter name is displayed next to the value of that parameter.

- **To clear a parameter name so that no name is displayed next to the parameter on the screen**, press the **{CLEAR NAME}** softkey. The message "SELECT NAME TO CLEAR AND PRESS ENTER" appears. Select the name to clear and press the **[ENTER]** key. The currently selected parameter name is deleted.
- **To clear all of the parameter names that are found on the {COMMON 2A} screen for all of the parameters**, press the **{CLEAR ALL NM}** softkey. The prompt "OK TO CLEAR ALL NAMES? (Y/N): " appears. Press Y followed by the **[ENTER]** key if it is okay to delete all parameter names. Press N followed by the **[ENTER]** key if you want to abort the delete-all-name operation.
  - The parameter name is used only for display purposes. It has no real function other than to permanently label a parameter value. The parameter name is retained as is the parameter value for these parameters even after power is turned off. The softkeys used to edit the parameter name operate this way:

(softkey level 3)

Addressing Assigned Parameters

Once you assign a parameter you can address it in a program:

Example 27.16
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

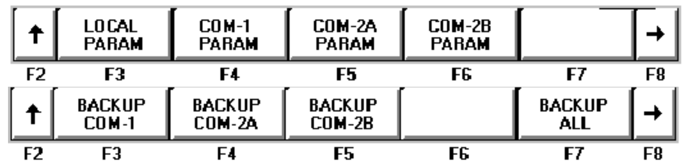
Example 27.17
Indirectly Addressing Parameters

```
#100=101
#101=2.345
G01 X#[#100];  X axis moves to the
               contents  of #100 which
               is #101. #101 has the
               value of 2.345.
```

Backing Up Parameter Values

You can back up the contents of COM1, COM2A, or COM2B individually, or all of these simultaneously, by using the BACKUP softkeys.

(softkey level 2)



To back up parameters:	Press this softkey:
#100 - 199	**{BACKUP COM1}**
#500 - 519	**{BACKUP COM2A}**
#520 - 999	**{BACKUP COM2B}**
all of the above	**{BACKUP ALL}**

1. Press the appropriate BACKUP softkey.

The system prompts you for a file name.

2. Enter a name for the backup file and press [**ENTER**].

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.

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.

Macro Call Commands

When a paramacro is called, execution of the currently active part program is halted, and execution is transferred to the macro program. Call paramacros in the following ways:

- Programming G65 in a part program
- Programming G66 or G66.1 in a part program
- Setting the proper AMP data can call a paramacro with the programming of specific G-, T-, S-, M-, and B-codes

You can use a paramacro call to call any program that has a program name of up to five numeric digits following the letter O (refer to chapter 9 on program names). This program must also contain an M99 end of subprogram or macro code somewhere in the program before an M02 or M30 is read. This M99 code causes control to return to the main program or restarts the paramacro if it is to be executed more than one time.

Important: The M99 code may be programmed anywhere in a paramacro program block provided no axis words are programmed to the left of the M99. Any information (other than axis words) programmed to the left of M99 is executed as part of the paramacro. Any information (including axis words) programmed in the block to the right of the M99 command is ignored.

M99X10;	X10 is ignored
X10M99;	Error is generated
M03M99;	M03 is executed

After the control has executed the macro the specified number of times (as specified by the L-word), execution is returned to the block following the paramacro call in the calling program.

ATTENTION: Any edits made to a subprogram, or to a paramacro program (as described in chapter 5) that has already been called for automatic execution, are ignored until the calling program is disabled and reactivated. Subprograms and paramacros are called for automatic execution the instant that the calling program is selected as active (refer to chapter 7).

Nonmodal Paramacro Call (G65)

Use this format for calling a paramacro using the G65 command:

```
G65 P_ L_ A_ B_;
```

Where:	Is:
P	Indicates the program number of the called macro. P ranges from 1 - 99999.
L	Programs the number of times the macro is executed. L ranges from 1 - 9999, and may be expressed as any valid parametric expression. If not specified, the control uses a default value of 1.
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 27.H). Arguments may be specified as any valid parametric expression.

The G65 command is nonmodal. The macro is executed only at the time the control executes the G65 block. The control calls the macro specified by the G65 block as programmed by the P-word.

The control executes this macro until the control reaches an M99 macro return code. The macro then returns to the next unexecuted sequential block in the calling program unless the macro has not been repeated the number of times as determined by the L-word. If this is the case, the macro re-executes.

You can define the L-word or any optional argument statements in a G65 block by using any valid parametric expression. For example:

```
G65 P1002 L[#1+1] A[12*6] B[SIN[#101]];
```

Modal Paramacro Call (G66)

Use this format for calling a paramacro using the G66 command:

```
G66 P_ L_ A_ B_;
```

Where:	Is:
P	Indicates the program number of the called macro. P ranges from 1 - 99999.
L	Programs the number of times the macro is executed after each motion block that follows the G66. L ranges from 1 - 9999, and may be expressed as any valid parametric expression. If not specified, the control uses a default value of 1.
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 27.H). Arguments may be specified as any valid parametric expression.

The G66 command is modal and remains in effect until canceled with a G67 block. The macro programmed by the P-word in the G66 block is not executed when the G66 block is read. The control delays macro execution to any block following the G66 command that contains a motion command.

When the control encounters a motion block (even if this block is contained in a different macro) following the G66 block, it executes the motions called for by that block first. After that block has been executed, the control then calls the macro specified by the G66 block.

The control executes this macro until the control reaches an M99 macro return code. The macro then returns to the next unexecuted sequential block in the calling program unless the macro has not been repeated the number of times as determined by the L-word. If this is the case, the macro re-executes.

Each time that a specific macro is called by a motion command, it is executed the number of times programmed with the L-word. All local variables remain at their current value throughout the program unless replaced, the control is reset, E-STOP is reset, or the control encounters an M02 or an M30 code in a program.

An L-word programmed with a G66 macro call cannot be replaced without 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 by using any valid parametric expression. For example:

```
G66 P1002 L[#1+1] A[12*6] B[SIN[#101]];
```

Unlike nonmodal macro calls, the G66 macro call repeats automatically after any axis move until cancelled by a G67 block. This also applies to nested macros. When the control begins execution of the nested macro 1002 in the program below, each axis move in the nested macro also calls for the execution of the macro 1001.

Example 27.18
Modal Macro Call

```
N0100G66P1001;
N0200G65P1002;
```

In Example 27.18, after the complete execution of the macro 1002, the macro 1001 is called. Any motion blocks in macro 1002 cause macro 1001 to be executed.

Example 27.19
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.

Modal Paramacro Call (G66.1)

Use this format for calling a paramacro using the G66.1 command:

```
G66.1 P_ L_ A_ B_;
```

Where:	Is:
P	Indicates the program number of the called macro. P ranges from 1 - 99999.
L	Programs the number of times the macro is executed. L ranges from 1 - 9999, and may be expressed as any valid parametric expression. If not specified, the control uses a default value of 1.
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 27.H). Arguments may be specified as any valid parametric expression.

The G66.1 command is modal and is executed in the same manner as the G66 with these exceptions:

- The macro programmed by the P-word in the G66 block is not executed when the G66 block is read, whereas the macro programmed by the G66.1 is executed when G66.1 is read.
- The macro is executed in any and all blocks following the G66.1, not just after motion blocks, except for paramacro command blocks such as assignment, goto, etc.
- Axis motion cannot be generated by normal program blocks. Axis motion can be generated only in the program called by G66.1.
- The following words, when programmed after the G66.1 block, are used as argument assignments:

N: when programmed after a word other than N or O, is used as assignment #14.

G: The last G-code programmed in a block 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 27.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 27.20
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.

Example 27.21
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]];
```

AMP-defined G-code Macro Call

Use this format for calling an AMP-defined macro:

```
G_ A_ B_;
```

Where:	Is:
G_	Programs an AMP-defined G-code command (from G1 to G255.9).
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 27.H). Arguments may be specified as any valid parametric expression.

An AMP-defined G-code macro is a G-code that is specified in AMP by the system installer. When one of these AMP-defined G-codes is executed in a part program, execution is transferred to the macro with the program number associated to that G-code.

G-code values for paramacro calls may range from 1 to 255.9. The system installer may define a maximum of 25 AMP-defined G-codes to call specific paramacro programs. The paramacro program name called with the AMP-defined G-code is a program number from 1 to 8999 or 9010 to 9019. Refer to the system installer's documentation for details.

Important: The system installer may disable the use of AMP-defined G and M-code macro calls when in MDI mode. Refer to the system installer's documentation to determine if this feature is functional in MDI.

AMP-defined G-code macros can be executed as either modal or nonmodal macros as selected in AMP. If selected as modal, they can be execute using either G66 modality (see page 27-37) or G66.1 modality (see page 27-39). This modality type for AMP defined G-codes is also determined by the system installer in AMP.

Any optional argument statements following an AMP-defined G-code may contain any valid parametric expression. For example:

```
G255A[12*6]B[SIN[#101]];
```

In a part program, if more than one digit is entered after the decimal point, the value 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 any other AMP-defined macro call. For details, see page 27-43.

AMP-defined M-code Macro Call

Use this format for calling an AMP-defined M-code macro:

```
M255 A_B_
```

Where:	Is:
M255	Programs an AMP-defined M-code command.
A-Z	Optional argument statements. May be programmed using any letter from A to Z excluding G, L, N, O, or P. Used to assign numeric values to parameters in the paramacro (see Table 27.H). Arguments may be specified as any valid parametric expression.

These macros are executed only as nonmodal macro.

The term AMP-defined M-code macro comes from the fact that the M-code that calls a specific macro program is specified in AMP by the system installer. The system installer may define M-codes that calls paramacro programs with program names ranging from 9001 to 9009. Refer to the system installer's documentation to determine what M-codes are used to call what paramacro program name.

When one of these AMP assigned M-codes is specified in a part program, execution is transferred to the macro associated to that specific M-code.

M-code values for paramacro calls may range from -1 to 999. The system installer may define a maximum of 9 AMP-defined M-codes to call specific paramacro programs.

Important: The system installer may optionally disable the use of AMP-defined G and M-code macro calls when in MDI mode. See the system installer's documentation to determine if this feature is functional in MDI.

AMP-defined T-, S-, and B-code Macro Call

Use this format for calling an AMP-defined T-, S-, or B-code macro:

```
T t ;
S s ;

or

B b ;
```

Where:	Is equal to the value assigned to parameter:
t	#149
s	#147
b	#146

Important: Programming arguments are not allowed with the AMP-defined T-, S-, or B-code macro calls.

These macros are executed only as nonmodal macro.

The execution of the T-, S-, or B-code macro calls is the same as M-code macro calls with the following exceptions:

- the parameter # referenced when called
- the macro program called
  T calls macro 9000
  S calls macro 9029
  B calls macro 9028

In order for the T-, S-, or B-words to call up a macro program, these prerequisites must be met:

1. The value following the word must be equal to the value stored for the specified parameter #.

   For example:

```
T14;
```

   The value of 14 must have been previously stored as the value for the parameter #149.

2. An AMP flag for that specific word must be turned on by the system installer to allow that word to call a macro.

3. The value for an AMP-defined T-, S-, or B-code command has the same format and range as an ordinary T-, S-, or B-code.

Nesting Macros

Nesting occurs when one program calls another program. A subprogram called by a main program is an example of nesting. (The "nested" program is the program called.)

Nesting applies to macros as well. When the main program calls a macro, the macro is said to be on nesting level 1. If this macro in turn calls another macro, this second macro is said to be in nesting level 2. Macros may be nested up to a maximum of 4 levels. However, if the maximum number of nested paramacros (4) is combined with up to 4 subprograms that end with M98, a maximum of 8 levels of nesting can be programmed.

What is **not** counted as an additional nested level? When a lower nested macro with a modal feature forces a higher nested macro to call it, the number of nested levels does not increase. Nor does it increase when a subprogram is called using M98.

Precautions must be taken when attempting to nest AMP assigned macro calls since many combinations of these calls may not be valid. The system installer determines in AMP the functionality of the AMP-defined macro call when nested. These two options are available (see the system installer's documentation to determine which applies to your system):

- Works as a macro call – When "works as a macro call" is selected,
  G-, M-, T-, S-, or B-code macro calls that are nested and called by other G-, M-, T-, S-, or B-code macro calls allow nesting as shown in Table 27.I.

Table 27.I
Works as a Macro Call

CALLING PROGRAM	TYPE OF MACRO NESTED [1]			
	G65, G66,or G66.1	AMP-G	AMP-M	AMP-T S or B
G65, G66 or G66.1	Yes	Yes	Yes	Yes
AMP G-code	Yes	No	Yes	Yes
AMP M-code	Yes	Yes	No	No
AMP T-, S- or B-code	Yes	yes	No	No

[1] What Yes/No means:
Yes -- the macro type across the top row may be called from the macro type down the left column.

No -- the macro type across the top row may **not** be called from the macro type down the left column. When this nesting is attempted, the control executes any other operation that would normally be performed by that G-, M-, T-, S-, or B-code (as defined by the 9/PC system as a standard code, logic, 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 27.J.

Table 27.J
Works as the System-defined Code

CALLING PROGRAM	TYPE OF MACRO NESTED [1]			
	G65, G66,or G66.1	AMP-G	AMP-M	AMP-T S or B
G65, G66 or G66.1	Yes	Yes	Yes	Yes
AMP G-code	Yes	No	No	No
AMP M-code	Yes	No	No	No
AMP T-, S- or B-code	Yes	No	No	No

[1] What Yes/No means:
Yes -- the macro type across the top row may be called from the macro type down the left column.

No -- the macro type across the top row may **not** be called from the macro type down the left column. When this nesting is attempted, the control executes any other operation that would normally be performed by that G-, M-, T-, S-, or B-code (as defined by the 9/PC system as a standard code, logic, 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 9/PC system as a standard code, logic, 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 27.J is that an AMP-assigned macro may **not** call an AMP-assigned macro.

For example, if the calling program is an AMP-assigned M-code macro, then G65, G66 and G66.1 macro calls are allowed; but no other types of macro calls are allowed, including an M-code macro.

END OF CHAPTER

Program Interrupt

Chapter Overview

This chapter describes the program interrupt feature. This feature lets you execute a subprogram or paramacro program while some other program is executing. This subprogram or paramacro is executed when logic receives an interrupt signal (usually through the use of some switch triggered by the operator or one of the axes). The interrupt program can be executed even mid-block during a programs execution.

Use this table to find the information in this chapter:

Enabling and Disabling Interrupts (M96/M97)

Interrupts are enabled or disabled on the control by two modal M-codes. These M-codes are determined in AMP by the system installer. This manual assumes the following values for these M-codes (these are the default values in AMP):

M96	Enables Program Interrupts
M97	Disables Program Interrupts

When program interrupts are enabled (M96) the correct interrupt signal to logic will cause an interrupt program to be executed. When program interrupts are disabled (M97) an interrupt signal is ignored.

The format for these M-codes is:

```
M96L__P__;
M97L__;
```

Where :	Selects:
L	the type of interrupt and the signal that will call the interrupt. L ranges from 0 - 3.
P	the interrupt program. P is followed by a 5 digit non-decimal program name.

An error is generated if anything other than an N word, a P or L word, a block delete /, or a comment character is programmed in the M96 or M97 block.

An interrupt M–code M96 or M97 may also be programmed within a interrupt program. If this is the case the interrupt does not become enabled/disabled until the interrupt currently being executed is completed and execution is returned to the main program.

Selecting the Type of Interrupt

Two types of interrupt programs may be enabled or disabled with these M-codes. These two types are described on page 28–5. You can use up to four signals to logic (switches) to call interrupt programs. The system installer determines what switch corresponds to which type of interrupt in logic. Refer to documentation prepared by the system installer for details on the types available and switches used to control them for a specific machine application.

The M–code determines which type and which switch signal is enabled or disabled by programming an L-word with the M–code. There are four L-words:

L0	Interrupt type 1	Switch 0 triggers this.
L1	Interrupt type 2	Switch 1 triggers this.
L2	Interrupt type 2	Switch 2 triggers this.
L3	Interrupt type 2	Switch 3 triggers this.

Program these L-words in each M96 or M97 block. Not programming the L-word causes the control to assume an L-word of L0 has been programmed. Refer to documentation prepared by the system installer for the location and operation of the 4 switches.

If more than one L-word is programmed in a block, the right most L-word is the one that is used for that block. Other L-words in that block are ignored.

Selecting an Interrupt Program

Any legal subprogram or paramacro may be selected as a interrupt program (refer to the section in chapter 9 on subprograms or chapter 27 for paramacros). For a program to be used as an interrupt program it must have a program name of 5 numeric characters following an O address (see the section on program names in chapter 9). This interrupt program must contain an M99 block as the last block in the interrupt program. This M99 block has all the same restrictions as an M99 block for a subprogram as discussed in chapter 9.

The subprogram or paramacro program is assigned to a particular type of interrupt by programming a P-word in the M block that enables the interrupt (M96 in this manual). When selecting a program with a P-word note that only the numeric value of the program name is entered, the letter O is omitted. For example, programming:

```
M96L0P11111;
```

would enable the program O11111 as a type 1 interrupt and allow it to be executed when switch 0 sends a signal to logic. If the program called with the P-word does not exist, the control will generate an error when the switch that activates the program is activated. Note the P-word is not programmed in the disable M-code block (M97 in this manual).

Example 28.1
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. Note that this is the same program as selected for type 0 interrupts.
N4M97L3;	Disables any interrupt program that is called by switch 3. Any signal to execute an interrupt from switch 3 is ignored after this block is executed unless reactivated with a M96L3 block.
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.

Interrupt Request Considerations

When using system interrupts, take into consideration:

- The system installer can determine in AMP if a signal to execute an interrupt program is delayed until the end of a currently executing block, or if the interrupt is executed immediately when the signal is received.

- The system installer can determine in AMP whether an interrupt program request is recognized when an interrupt switch is turned on, or only when the switch makes the transition from off to on. This is to help prevent the accidental execution of an interrupt program if a switch is inadvertently left on when a program begins execution.
- Interrupt programs should normally be disabled during thread cutting. The execution of an interrupt program during a threading pass may cause undesired results.
- The system installer has the option of writing logic to allow the use of 4 interrupt signals (4 switches). One of these signals may call a type 1 interrupt; the other three call a type 2 interrupt.
- Before an interrupt signal is recognized by the control it must first have been enabled by programming a M96 followed by the correct L-word for that signal.
- Interrupt programs may only be executed when the control is in the automatic mode. Interrupt requests that occur during MDI or manual modes are ignored.
- Cutter compensation is temporarily cancelled during the execution of an interrupt program. It is reactivated upon completion of the interrupt program. No entry or exit move from compensation is performed.
- Interrupts that are requested when the control is in E-Stop are ignored regardless of whether the interrupt is enabled or not.
- Interrupts can only be executed when the control is in the **<CYCLE START>** state. If a request for an interrupt is made when the control is in **<CYCLE STOP>** or cycle suspend the interrupt request is still recognized. The interrupt program will be executed when a **<CYCLE START>** state becomes active again.
- If an interrupt occurs during a block retrace, the interrupt will be performed. The block retrace however will be aborted at that point and no further retrace will be allowed. Block retrace will, however, still be able to return any moves that have already been retraced before the interrupt occurred.
- During the execution of a milling cycle, if the interrupt is a delay type (executed at the completion of the currently executing block), the control will execute the interrupt after all motions generated by that block are completed. If the interrupt type is immediate (executed as soon as the interrupt signal is received), the control interrupts the currently executing path.

Interrupt Types

There are two types of interrupts, type 1 and type 2. These are selected by the L-word in the M96 block. L0 activates type 1 interrupts and L1, L2, and L3 activate type 2 interrupts. Type 1 and type 2 interrupts are shown in Figure 28.1 and Figure 28.2.

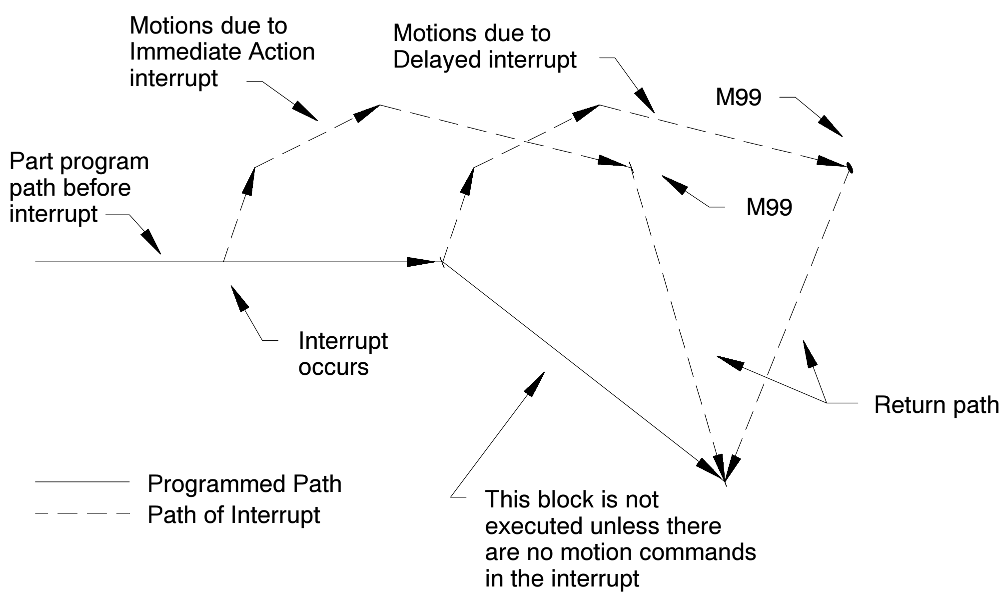
The key difference between a type 1 and a type 2 interrupt is the tool path that is taken when the return from interrupt is made as programmed with an M99 in the interrupt program.

Type 1 Interrupts

If no axis motion is generated by the interrupt program then the control executes the interrupt program and then continues executing the part program as normal regardless of the location that the interrupt program was executed.

If axis motion is generated by the interrupt program then the control returns the tool to the endpoint of the next fully unexecuted block and continues executing the part program from this point.

Figure 28.1
Type 1 Interrupt

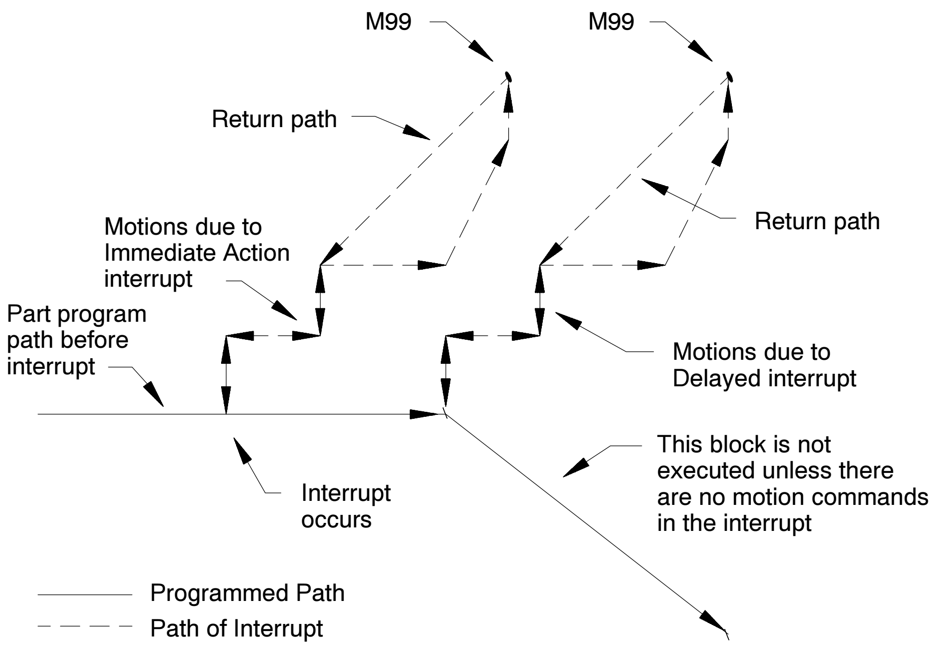


ATTENTION: If the interrupt is executed in the middle of a blocks execution, and there is axis motion in the interrupt program than the control will never reach the end point of the interrupted block, execution will transfer to the end point of the following block. This applies to type 1 interrupts only.

Type 2 Interrupts

The control returns the tool to the point in the program where it was when the interrupt was performed by using type 2 interrupts. Normally the first four linear moves (G00 or G01) in the interrupt program are remembered (this may be altered by programming a specific M-code as discussed later). If a non-linear (G02 or G03) circular move is performed as one of the first four blocks of the interrupt, the control only remembers the moves up to that block. If that block is the first block in the interrupt then the start point of that block is all that is remembered. These remembered blocks are retraced after the M99 code is read by the control in the interrupt program. This returns the cutting tool to the location in the program where program execution was interrupted.

Figure 28.2
Type 2 Interrupt



It is possible to alter the number of blocks that the control re-executes in reverse when returning to the start position of the interrupt. The number of return blocks is normally four; however, it may be altered by programming one of these M-codes:

M900 - zero blocks retraced

M901 - one block retraced

M902 - two blocks retraced

M903 - three blocks retraced

M904 - four blocks retraced

These M-codes may be programmed in any block in the main program that is before the interrupt program is executed. They may also be programmed within an interrupt though the M-code will not go into effect until execution is returned to the main program. If not programmed, the default is four blocks. The default is automatically reset at the end of program commands (M02 or M30).

The number of retrace blocks as set with this M-code is the same for all active or inactive interrupts. If an interrupt is enabled after this M-code is programmed it will take on the number of retrace blocks as programmed with this M-code.

When the return from interrupt is executed (M99 in the interrupt) the control will generate a linear move to the end point of the last remembered move for retrace. Then the moves are retraced returning the tool back to the start point of the interrupt. Note this may not be the same location in the main program if a different tool offset has been activated.

The Interrupt Program

When you intend to use a program as an interrupt program, remember:

- Any modal data (G-codes, feedrates, spindle speeds, coordinate system offsets, etc.) contained in the main program are carried into the interrupt program. Any changes made to this modal data within the interrupt will be effective only in the interrupt program; changes are not carried back into the main program when the interrupt is completed. This does not include tool or tool offset data that is changed in the interrupt. Any tool or tool offset changes will be carried back into the suspended main part program.
- The system installer can determine if an interrupt program is to be called as a paramacro when executed, or a subprogram when executed.
- If it is to be called as a paramacro, remember that this assigns a new set of local parameters for the interrupt. If it is to be called as a subprogram, the same set of local parameters that applied to the interrupted program apply to the subprogram.
- If an interrupt is chosen as a macro program, it may not be a macro that requires the assignment of local variables in the calling block (can not require an argument).
- Macro type interrupts are always called as the G65 non-modal type. G66 and G66.1 modal types may not be called. Refer to the chapter on paramacros for details on the G65 type macros.

- The interrupt program must contain an M99 block. Any axis motion commands that are to the left of the M99 code in the block will result as an error. Other programming commands to the left of the M99 code in the block will be executed. Any characters to the right of the M99 code in the block are ignored.
- If using a type 2 interrupt (L1, L2, or L3), remember that the control remembers up to the first 4 blocks in the program and uses these to retrace its moves back to the starting point of the interrupt program. The control remembers up to four of the first moves or until a circular block is executed. For details, refer to page 28-5 for more information about interrupt types.
- The interrupt program may contain a milling cycle in the interrupt.
- These G-codes are illegal in an interrupt program:

  cutter compensation G-codes G40, G41, and G42

  coordinate system offsets G52, G92, and G92.1

- Any inherent modality from the main program (such as a milling cycle, or an active modal paramacro) will be temporarily canceled during the execution of a interrupt program.
- Only one interrupt may be executing at any given time. All four may be active at once, but only one may be executing. This means that an interrupt may not be executed during the execution of another interrupt.

END OF CHAPTER

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 screen. The function of each softkey is displayed on the screen 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.

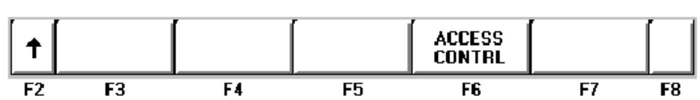
Specific softkeys for all levels change depending on the previous softkey pressed, with the exception of softkey level 1, which always remains the same. Softkey levels are all referenced from softkey level 1.

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 :



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)

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

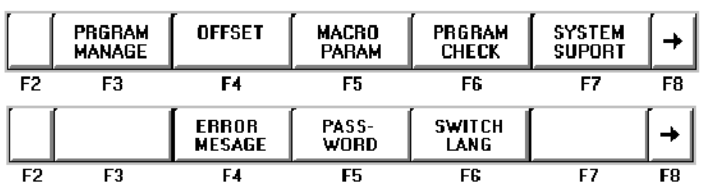
The softkey functions for level 1 softkeys are explained in this appendix. Softkey functions for level 2 or higher are explained in the sections that apply to specific operations. A "tree" of softkeys listing all the softkeys and their levels is included in the back of this appendix.

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

The following section describes Level 1 softkeys and their prospective functions.

(softkey level 1)

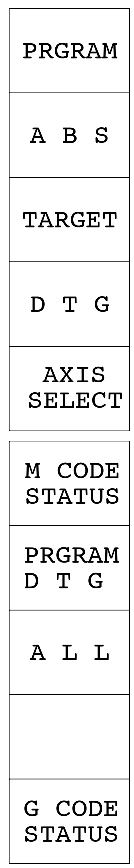


If you want to:	Press:
edit, activate, or copy a program from control memory	**{PRGRAM MANAGE}**
display or enter tool offset data, the work coordinate system offset data, etc.	**{OFFSET}**
view and modify the local and global parameter assignments for paramacros	**{MACRO PARAM}**
check the part program, QuickCheck, and active program without actually moving an axis	**{PRGRAM CHECK}**
enter and display inhibit zone limits, canned cycle parameter data, AMP, etc.	**{SYSTEM SUPORT}**
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(after selecting {F12})



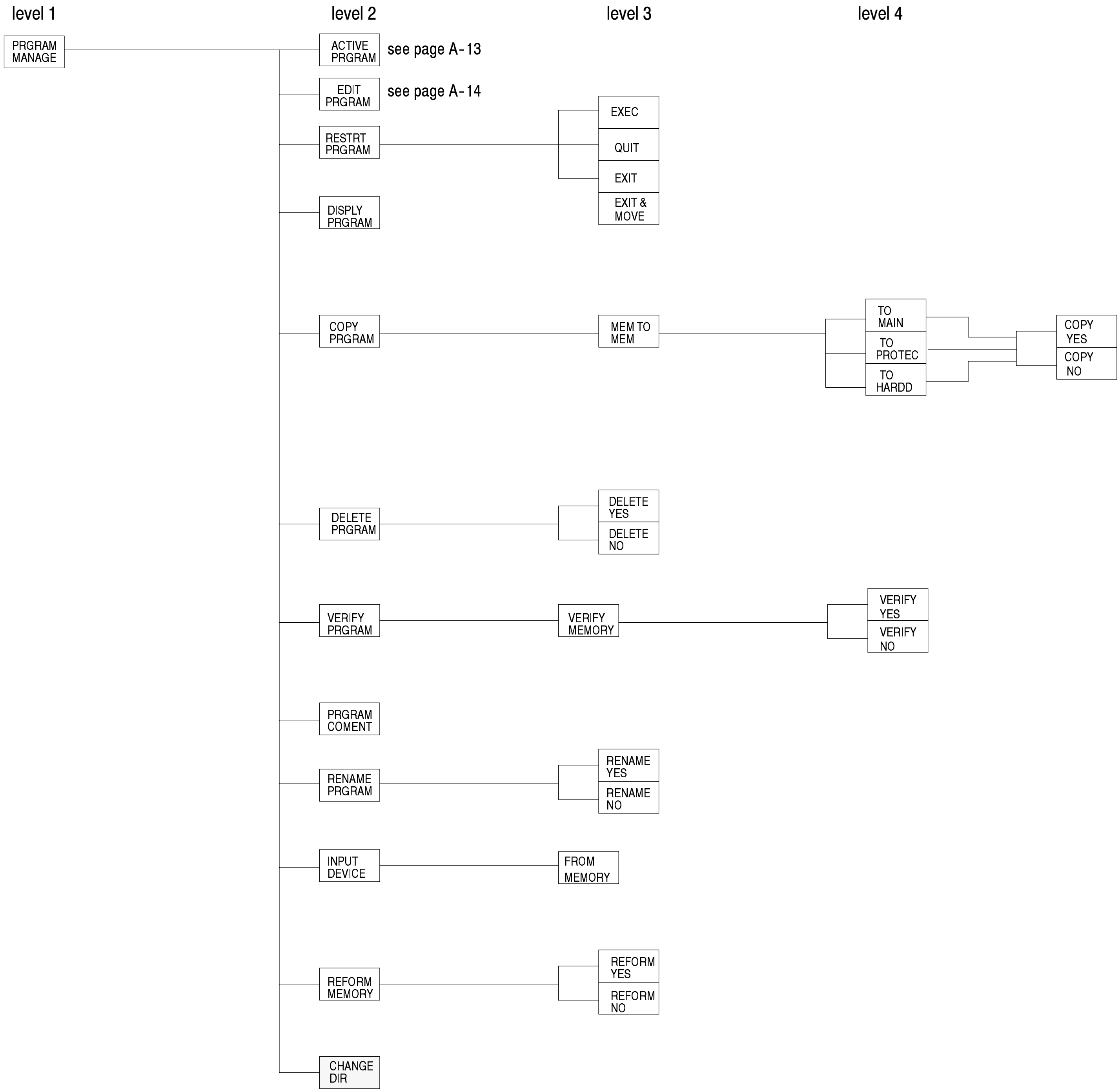
NOTE: The first four softkeys (from {PRGRM} to {DTG}) toggle between a small and large screen display.

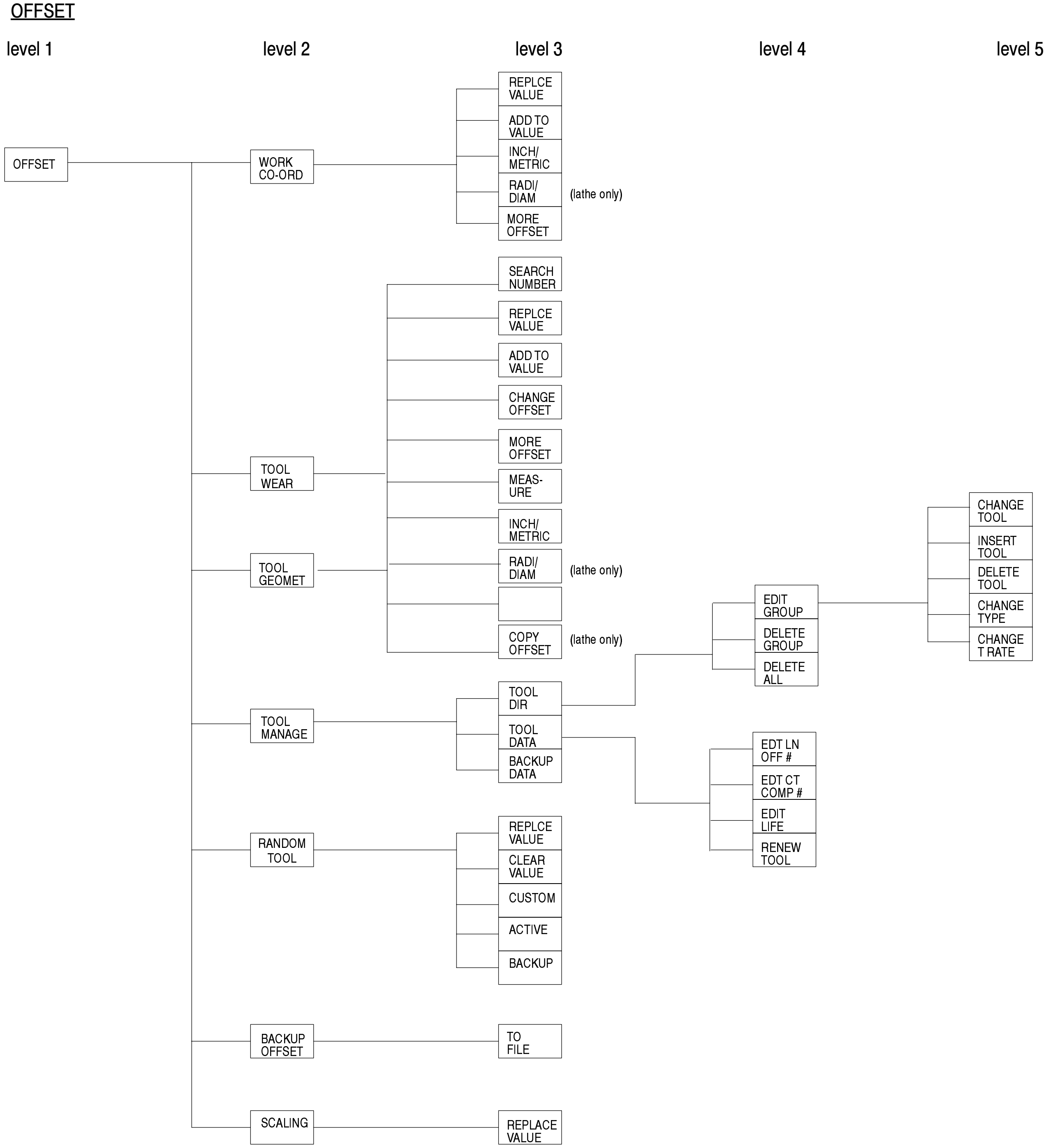
THE FUNCTION SELECT SOFTKEYS LEVEL 1

WITH POWER UP (AXIS POSITION) DISPLAY SCREEN

PRGRAM MANAGE	refer to page A-5
OFFSET	refer to page A-6
MACRO PARAM	refer to page A-7
PRGRAM CHECK	refer to page A-8
SYSTEM SUPORT	refer to page A-9
ERROR MESAGE	refer to page A-10
PASS-WORD	refer to page A-11
SWITCH LANG	refer to page A-12

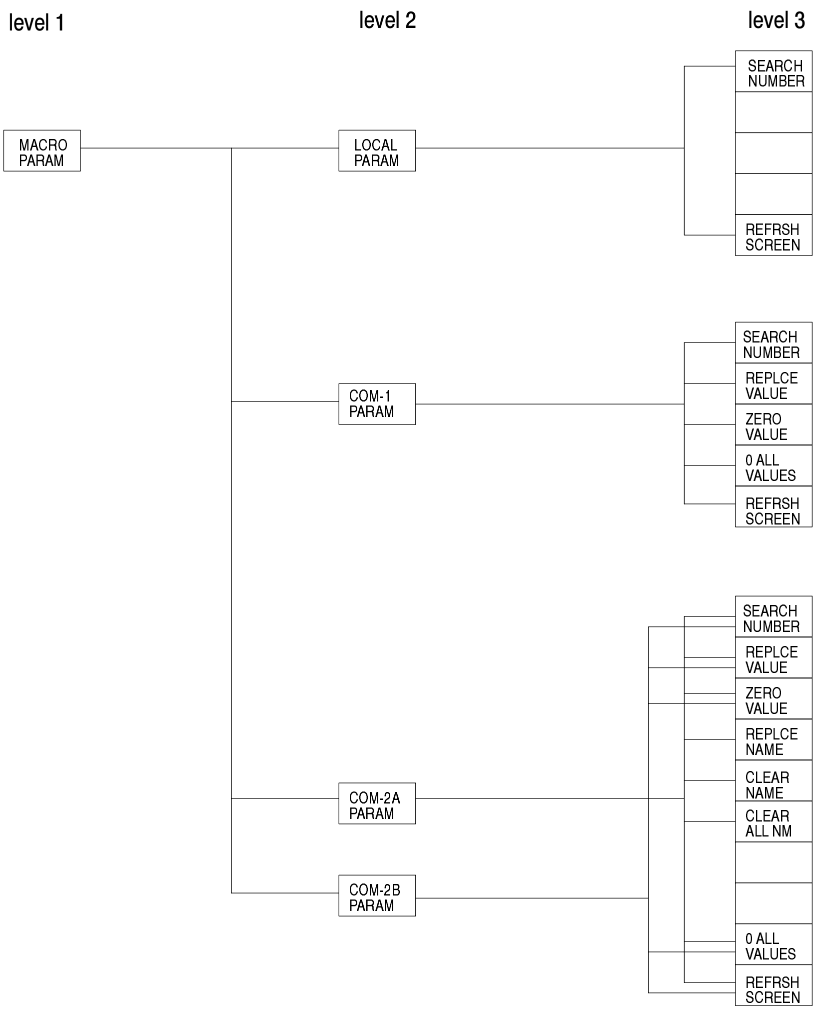
PRGRAM MANAGE



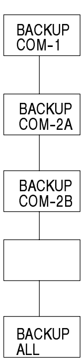


OFFSET
level 1
level 2
level 3
level 4
level 5
OFFSET
WORK CO-ORD
REPLCE VALUE
ADD TO VALUE
INCH/ METRIC
RADI/ DIAM
(lathe only)
MORE OFFSET
TOOL WEAR
TOOL GEOMET
SEARCH NUMBER
REPLCE VALUE
ADD TO VALUE
CHANGE OFFSET
MORE OFFSET
MEAS- URE
INCH/ METRIC
RADI/ DIAM
(lathe only)
COPY OFFSET
(lathe only)
TOOL MANAGE
TOOL DIR
TOOL DATA
BACKUP DATA
EDIT GROUP
DELETE GROUP
DELETE ALL
CHANGE TOOL
INSERT TOOL
DELETE TOOL
CHANGE TYPE
CHANGE T RATE
EDT LN OFF #
EDT CT COMP #
EDIT LIFE
RENEW TOOL
RANDOM TOOL
REPLCE VALUE
CLEAR VALUE
CUSTOM
ACTIVE
BACKUP
BACKUP OFFSET
TO FILE
SCALING
REPLACE VALUE

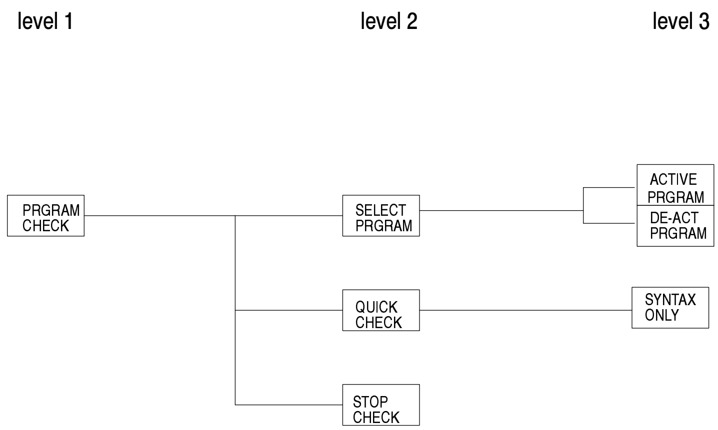
MACRO PARAM



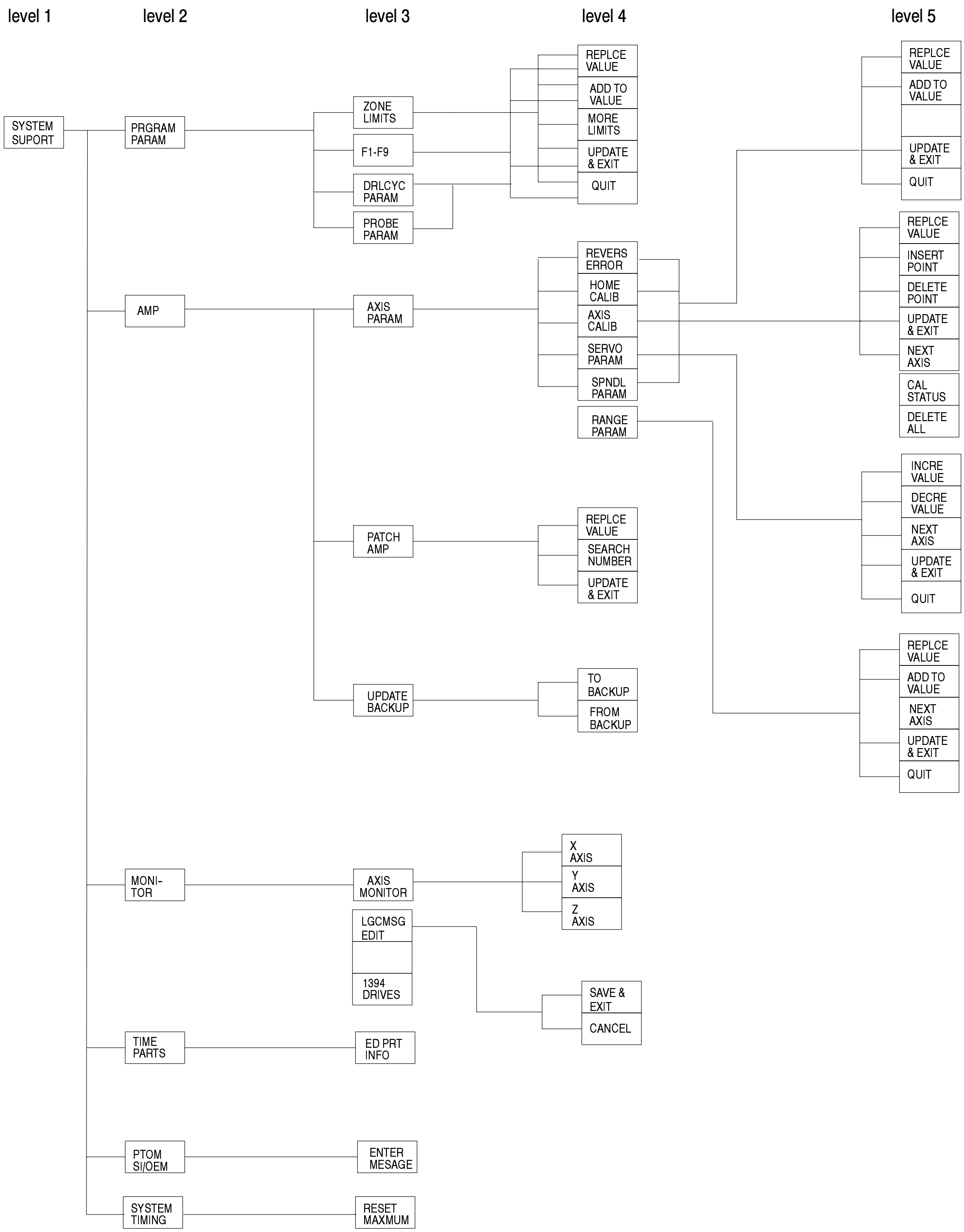
BACKUP PARAM



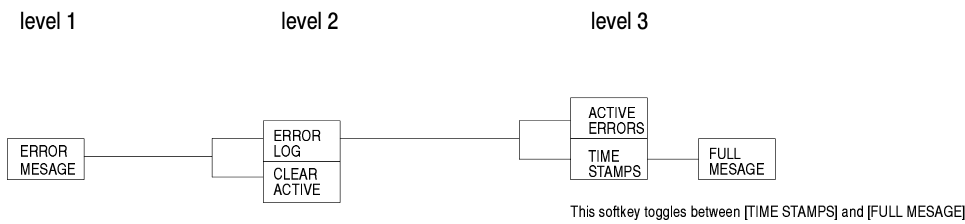
PRGRAM CHECK



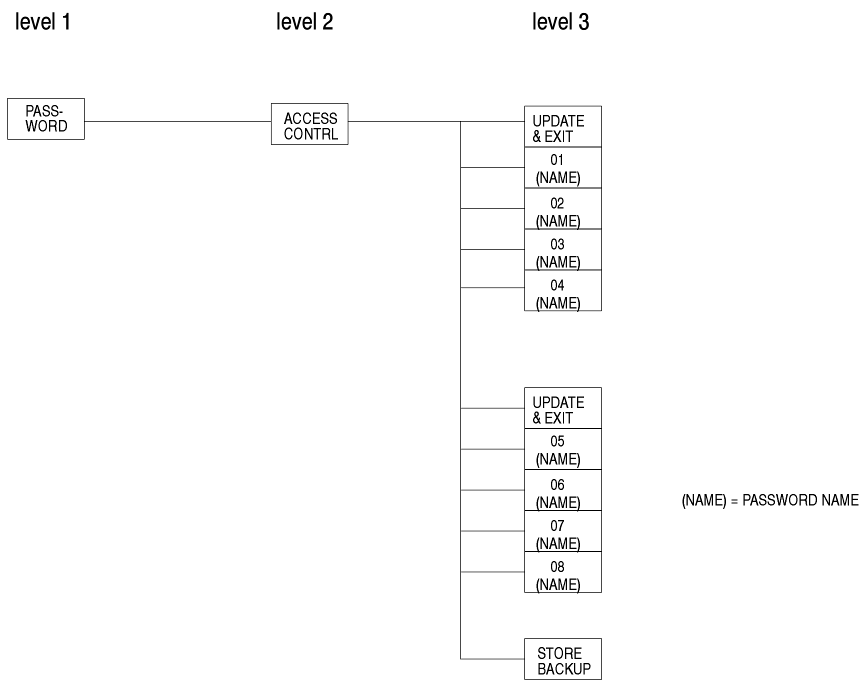
SYSTEM SUPPORT



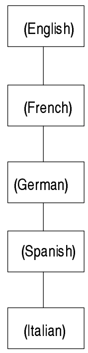
ERROR MESAGE



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

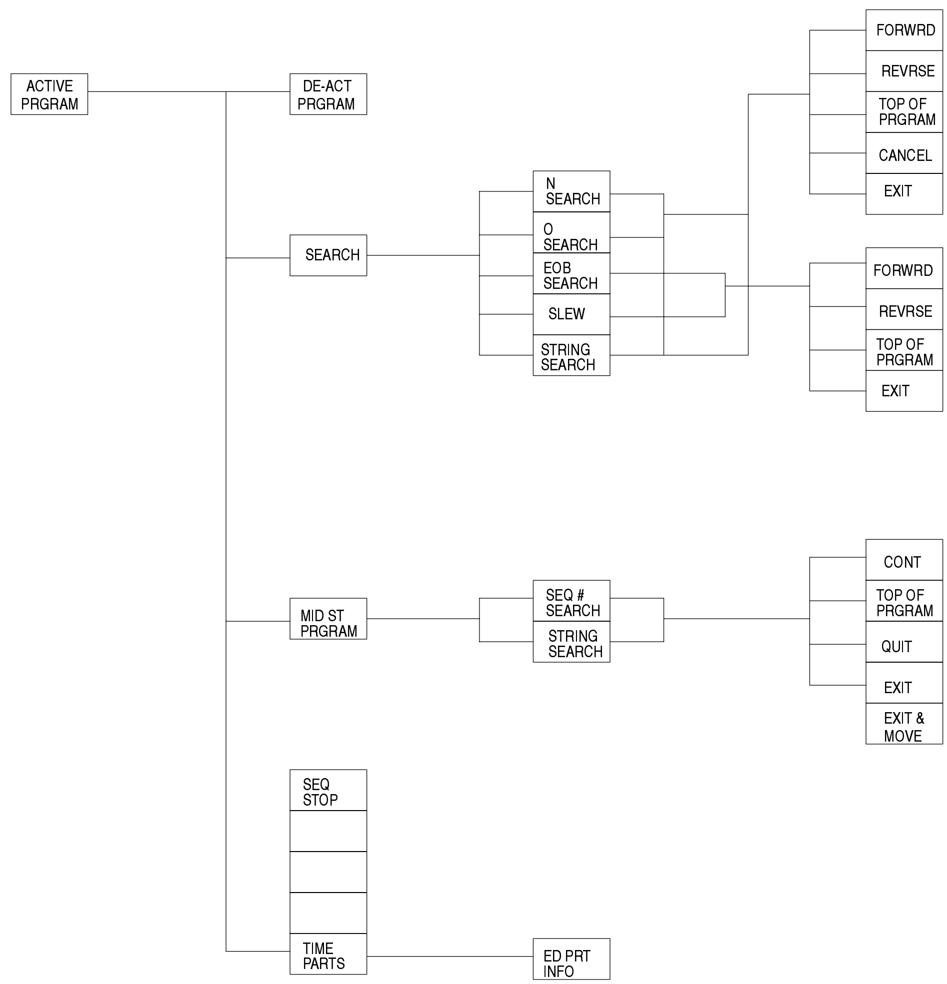
SWITCH LANG



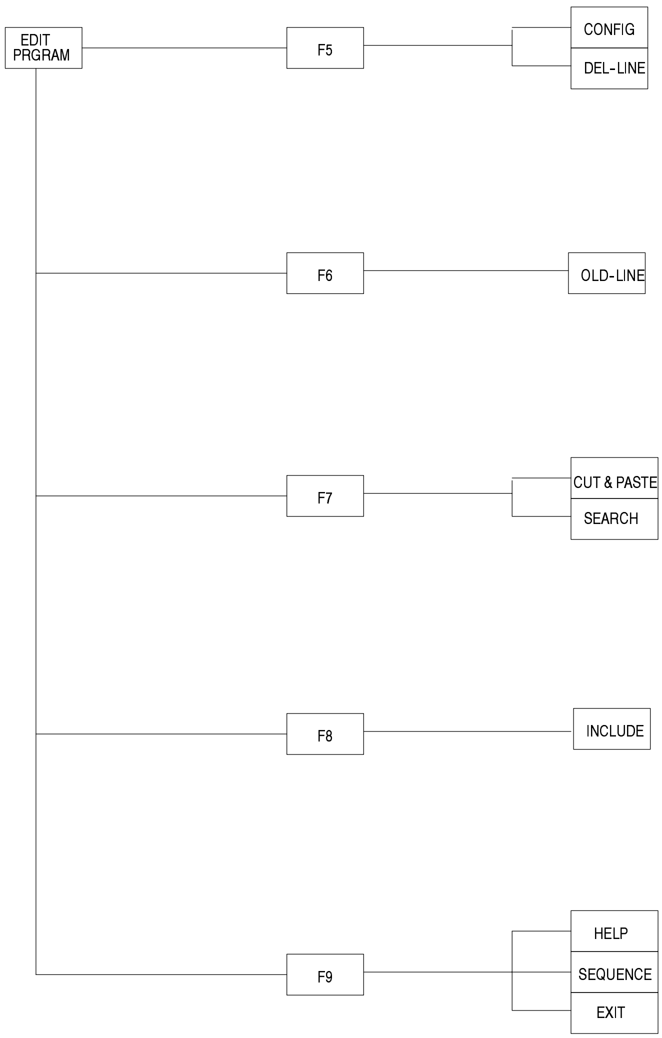
This chart shows the order that the languages are changed.

ACTIVE PRGRAM

level 2 level 3 level 4 level 5



EDIT PRGRAM



NOTE: These keys are the layered softkeys available through the part program editor.

END OF APPENDIX

Error and System Messages

Overview

This appendix serves as a guide to error and system messages that can occur during programming and operation of your 9/PC 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, refer to chapter 2.

Important: This appendix covers only error and system messages. Logic-generated operator messages generally appear on lines 21 and 22 of the BDS screen and should be described in documentation prepared by the machine tool builder.

Message	Description
Symbols	
(+) 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. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
15V SUPPLY FAILURE	15V dc is out of range on the main processor board. The System Ready Contacts open and the system goes into E-Stop. Check the wiring on the analog output circuits to make sure there is not a short.
1746 RACK CARDS MISMATCH	The I/O configuration for the 1746 I/O rack that was downloaded from ODS, or resides in the PAL PROMs, contradicts what is actually in the rack (devices must match slot-for-slot).
1771 RACK CARDS MISMATCH	The I/O configuration for the 1771 I/O rack that was downloaded from ODS, or resides in the PAL PROMs, contradicts what is actually in the rack (devices must match slot-for-slot).

Message	Description
2	
2MB RAM IS BAD/MISSING	The control has discovered the RAM SIMMs for the two megabyte extended storage option are either damaged or missing. The RAM SIMMs must be installed or replaced. Contact your Allen Bradley sales representative for assistance.
9	
9/SERIES LATHE - CANNOT USE MILL AMP	The control was powered up with a **lathe** software option chip installed, when the AMP file that was downloaded was configured for a mill.
9/SERIES MILL - CANNOT USE LATHE AMP	The control was powered up with a **mill** software option chip installed, when the AMP file that was downloaded was configured for a lathe.
7300	
7300 NAMES TABLE IS CORRUPTED	7300 program name doesn't match corresponding name in cross-reference table.
7300 PATTERN NAME TOO LONG	More than 5 digits have been used in the pattern name.
A	
A RETRACE BUFFER WAS DELETED	The control required one (or more) of the block retrace buffers to perform a necessary block look-ahead operation (refer to block look-ahead in the user's manual). When this occurs, less block retrace operations can be performed than AMP is configured to allow. If this error occurs, to improve control efficiency, it is recommended that the number of allowable block retrace blocks set in AMP be lowered or add additional RAM to you system.
ABS POSITION NOT INITIALIZED	This message indicates that axes with absolute encoders have not been homed. These axes require an initial homing operation to establish the absolute position.
ABSOLUTE FEEDBACK FAILURE	The control has detected a loss of feedback from the absolute encoder. The most likely cause of this error would be a broken or disconnected wire. Axis homing may be required after the error condition is corrected.
ACC/DEC CONFIGURATION ERROR	An axis configuration error was detected by the control when manual acc/dec was requested in a program block.
ACCUM. AND EXPECTED LIFE ARE 0	No tool life data was entered for the current tool selected by the tool life management feature. Tool life management will be disabled for this tool.
ACTIVE GROUP CANNOT BE DELETED	An attempt was made to delete a tool group in the tool life management feature that contains an active tool currently in the tool holder.
ACTIVE OFFSET CANNOT CHANGE	An attempt was made to alter a tool offset value of a tool offset that is currently the active tool offset on the control. The active tool offset is indicated with an * on the tool offset table.
ACTIVE TOOL CANNOT BE CHANGED	An attempt was made to edit tool data for the currently active tool. Deactivate the tool before editing.
ACTIVE TOOL CANNOT BE DELETED	An attempt was made to delete tool data for the currently active tool. Deactivate the tool before editing.
ADAPTIVE FEED MAX LIMIT	The actual torque is less than the desired programmed torque and the adaptive feed axis has reached the programmed maximum feed limit. Either raise the programmed maximum feed limit or lower the programmed desired torque.
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 CHECKSUM ERROR - CNC STOPPING	The saved checksum that protects the AMP area of the executive does not match the computed checksum of the AMP data that the executive is operating with. AMP is corrupt and was not saved into the flash.
AMP FILE SIZE ERROR	The size of the AMP file being downloaded is incorrect. The file cannot be downloaded.

Message	Description
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.
AMP WAS MODIFIED BY PATCH AMP UTILITY	This message always appears after changes have been made to AMP using the patch AMP utility. Its purpose is to remind the user that the current AMP has not been verified by a cross-reference check normally performed by ODS. It is meant as a safety warning.
AMPED HOLDING OR DETECT TRQ OUT OF RANGE	This message is displayed when you have entered a value in AMP for either the holding torque or the detection torque, for the feed to hard stop feature, that is higher than the value entered for the servos available peak torque. You must change your AMP values.
ANALOG SERVO VOLTAGE FAILURE	A $\pm$ 15V to the servo cards has failed.
ANGLE WORD NOT ALLOWED	An angle word was programmed in a QPP block where it is not allowed, for example, programming an angle word in a circular QPP block.
ANGLED WHEEL AXES, JOG ONE AT A TIME	While in the angled wheel grinding mode you can not jog more than one axis in the angled wheel plane at any one time.
ANGLED WHEEL CONFIG ERROR	The angled-wheel grinder AMP downloaded to the control is not configured correctly. Make sure all necessary angled-wheel parameters are configured correctly and re-download AMP to the control.
ANGLED WHEEL NOT CONFIGURED	The user 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 CNC.
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. • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
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 nonscalable.
AXIS ASSIGNED TO LOGIC AXIS MOVER	The user attempted to move the axis configured as the logic axis mover axis by some means other than logic.

Message	Description
AXIS ASSIGNED TO PAL AXIS MOVER	The user attempted to move the axis configured as the logic 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.
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. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
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 the Acc/Dec command from the CNC 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. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
AXIS MODULE BUS 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. To clear this fault, you should: • determine the cause of the fault • resolve the situation • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
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 temperature sensors should be wired directly into the E-Stop string. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
AXIS MOVER CONFLICT WITH G16.3/G16.4	You requested a PAL or logic axis mover function on an angled wheel grinder. You can not use the PAL or logic 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.

Message	Description
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.
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.
BAUD RATE MUST BE 4MHZ FOR 1394 DRIVES	The baud rate for 1394 SERCOS drives must be AMPed at 4MHz. For third-party drives (with no 1394 SERCOS drives in use), the baud rate must be AMPed at 2 MHz.
BBU SAVE FAILED, USING LAST VALID DATA	There was an improper shutdown of your 9/PC. This message appears when the save of BBU data (i.e., tool offsets, paramacros, axis calibration, work coordinates, and AMP data) failed to occur after the 9/PC executive was started.
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 CODE CRC ERROR - CNC STOPPING	A hardware failure occurred while the boot executive was performing a CRC test of the boot flash. During the CRC test, an error was detected in the boot flash CRC. This is basically a self-test to confirm that the boot flash that loads and runs the CNC executive is OK.

Message	Description
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.
BOOT RAM ACCESS ERROR – CNC STOPPING	A hardware failure occurred while the boot executive performed a RAM test of the memory where the executive will be loaded. An error reading/writing into the designated RAM area occurred.
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 exit 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 .
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.

Message	Description
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 exiting 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.
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 displays when a nonprogrammed 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.

Message	Description
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 logic 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.
CHARACTERS MUST FOLLOW WILDCARD	You have used incorrect search string syntax in the PAL search monitor utility.
CHECKSUM ERROR IN FILE	The file (AMP, logic) 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/PC 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.

Message	Description
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 ERROR	The master or slave processor detected an error in the configuration of your 9/PC system.
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 reconfigure 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).
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.

Message	Description
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).
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.

Message	Description
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 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".
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- or logic-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.
DMA TRANSFER ERROR - CNC STOPPING	An error condition occurred as a result of a DMA transfer while the executive was running. The integrity of the data (logic PIT data or communication packets to the data server) is questionable. The CNC executive cannot continue operating.
(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.

Message	Description
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 or logic. 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 reconfigure 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.
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 or logic only allows one axis at a time to be homed. PAL or logic can be changed to allow homing of multiple axes in a dual group.

Message	Description
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.
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 INITIALIZING DMA - CNC STOPPING	When the CNC executive starts, it sets up the module DMA controller for transferring data to and from the PC. This error indicates that the DMA initialization sequence failed. Attempt to restart your 9/PC executive to recover from the error and to properly initialize the DMA.
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.

Message	Description
ESTOP STRING FAILURE	Software detected an apparent problem with the E-Stop string. Although the servo is attempting to open it, the Drive_OK contact is being held closed.
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 bootstrapper failed to respond within the specified time for any code segment. Consult Allen-Bradley customer support services.
EXECUTIVE CHECKSUM ERROR - CNC STOPPING	The saved checksum that protects the CNC executive code does not match the computed checksum of the executive. The executive is corrupt and AMP will not be saved into the flash.
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.
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 logic flags. You should configure the keyboard or HPG with the I/O assigner utility (refer to your *9/Series PAL Reference Manual* for details).
F	
FATAL FAULT OCCURRED - CNC STOPPING	A fatal fault occurred that did not report a specific error condition (e.g., an i960 internal NMI error). Contact your Allen-Bradley sales representative.
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.

Message	Description
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. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive (for Stegmann devices only) • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
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.
FEEDBACK OPTION NOT INSTALLED	A PTO check determines the legal number of axes.
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.
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 Installation and Integration 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 the 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.

Message	Description
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 G26 section of your operation and programming manual 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.
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.

Message	Description
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.
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 combination 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. This message may also appear if there was an attempt to use switchless homing on an axis with distance coded marker (DCM) feedback.
I	
I/O RING COMMUNICATIONS ERROR	A communication error 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.

Message	Description
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.
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 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.

Message	Description
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.
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 PART PROGRAM	There is not enough available memory for the current program to be stored. Any attempt to store the program in memory will be aborted by the control.
INTEGRANDS FOR DUALS MUST BE THE SAME	This is a power turn-on message. When using dual axes, all integrands of the dual group must use the same letter.
INTEGRANDS FOR NON-MASTER MUST BE NONE	An axis integrand name was configured in AMP that corresponds to an axis in a dual axis group that is not the master axis of that group. Only the master axis in a dual axis group can have a corresponding axis integrand name.
INTEGRANDS NOT AMPED PROPERLY	The axis integrand names were not configured properly in AMP. Refer to your AMP manual for additional details on axis integrand names.
INTERF CHECKING ZONE TABLE CORRUPTED	The zone tables used by interference checking have an invalid checksum and were cleared.
INTERNAL COMMUNICATIONS ERROR	Communication failed. Contact Allen-Bradley customer support services.
INTERRUPT NOT RECOGNIZED	An interrupt macro was not acted on for some reason. An example would be if an interrupt occurred in the middle of another interrupt.
INVAL LOOP BASE	An attempt was made to configure ports TB2 and TB3 as position/velocity loop or digital or digital spindle.
INVALID AMP-DEFINED G CODE	An attempt was made to assign the same G-code to different macro calls. This message appears after AMP is downloaded and the control does secondary calculations.
INVALID AMP LETTER FORMAT	The programmed word or parameter has an invalid letter format defined in AMP. Since ODS AMP detects and prohibits invalid formats, this error usually indicates that an invalid format was entered through patch AMP. Refer to your AMP reference manual for details.
INVALID ARC-COSINE ARGUMENT	An attempt was made to calculate or execute a paramacro block that calculates the arc cosine of an invalid or improperly entered number.
INVALID ARC-SINE ARGUMENT	An attempt was made to calculate or execute a paramacro block that calculates the arc sine of an invalid or improperly entered number. Change cosine to sine.
INVALID ARGUMENT ASSIGNMENT	An invalid argument assignment was programmed.
INVALID AXIS	The axis programmed in the adaptive depth (G26) or adaptive feedrate (G25) block is invalid. Valid axis names for programming these features are defined in AMP.
INVALID AXIS FOR CSS	The CSS axis (the axis that is perpendicular to the center-line of the rotating part) is not a valid axis on the control. This usually occurs when the CSS axis is changed from the default axis by programming a P-word in the G96 block that selects some other axis.

Message	Description
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.
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.

Message	Description
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.
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.
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.

Message	Description
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 LOGIC	The logic offset change feature specified an invalid tool number to the control.
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.
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.

Message	Description
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.
LOGIC ANALOG PORT ILLEGAL CONFIGURATION	This is a power turn-on error that occurs when an AMP configuration error exists in the logic analog port configuration.
LOGIC ANALOG PORT/SERVO F-W INCOMPATIBLE	Logic-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.
LOGIC AXIS STATUS CANNOT CHANGE	You attempted to change the status of a logic axis (either to logic 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...
LOGIC INITIATED MOTION POSSIBLE	While in QuickCheck mode it is possible for logic to physically move axes. This includes any motion generated by logic including the logic 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.
LOGIC NOT RESPONDING - CNC STOPPING	The logic engine failed to send a logic completed reply to the PC within the allotted amount of time (300 msec).
LOGIC OVERLAP ERROR	During the first or last iteration of a block the control is forced into E-Stop because: • the logic processor did not finish executing the logic in the allotted AMPed time during synchronous mode or • the logic processor did not finish executing the logic in the allotted AMPed time executing a nonmotion block in autosynchronous mode. E-Stop is forced.
LOGIC OVERWRITING G54 → G59.3	Logic is overwriting the current G54 - G59.3 offset through logic offsets.
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.

Message	Description
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.
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 reconfigure 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, logic error message, E-Stop condition, or PAL display page was 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.

Message	Description
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.
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.
MISMATCHED SERCOS/1394 DRIVE REVISIONS	The 9/PC SERCOS master is incompatible with the 1394 SERCOS slave. You must either update your 9/PC software or your 1394 software. For assistance with this problem, contact your local Rockwell Automation support person.
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.

Message	Description
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.
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 subprogram 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 Integration 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.

Message	Description
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.
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.

Message	Description
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.
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 Installation and Integration Manual* for details on installing additional SIMMS.
NEED SPINDLE FEEDBACK	You attempted to use the solid-tapping 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 control's 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.

Message	Description
NO OPTIONAL FB PORT ON ANALOG SERVO	The system was incorrectly AMPed with optional feedback module on an analog servo module.
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 your system. 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/PC, 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 1394 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.
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.

Message	Description
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.
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.

Message	Description
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 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.
PC COMMUNICATION LOST - CNC STOPPING	No communications between the CNC module and the PC occurred because the "heartbeat" signal was not established between the PC and the CNC module within the allotted time.
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.
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.

Message	Description
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 rearrange 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.
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.

Message	Description
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 ANGLE WRD SAME AS SECONDARY AUX. WRD	AMP downloaded an angle word for QuickPath Plus that is the same as the secondary auxiliary word. AMP must be reconfigured; the angle word cannot be the same as the secondary auxiliary word.
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.
RAMP/JERK OUT OF RANGE	There was an attempt to set an acceleration limit in a G48.n block that is outside the valid range. Check AMP for the allowable programmed range for the acceleration limit.
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.
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.

Message	Description
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).
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.

Message	Description
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 the 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.
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.
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.

Message	Description
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.
SERCOS COMMUNICATIONS LOST	Communications between the 9/PC SERCOS master and all devices in the fiber optic ring is lost. The SERCOS cable may be damaged, communications between one or more of the drives may be lost, or power to one or more of your drives may be off. The system remains in E-Stop until this condition is corrected.
SERCOS CYCLE TIME/FG SCAN TIME MISMATCH	An attempt was made to AMP a Fine Foreground Scan Time parameter that was not an integer multiple of the SERCOS Cycle Time parameter. Only integer multiples are permitted. To correct this error, reconfigure AMP so that the previously mentioned prerequisite is met.
SERCOS MASTER-9PC HANDSHAKE FAULT	The handshake between the 9/PC SERCOS master and the 9/PC main that occurs each coarse foreground scan did not complete in the time allotted. The system remains in E-Stop until a power cycle of the control is conducted. In the event that this error occurs, contact your Rockwell Automation support person for assistance.
SERCOS MEMORY FAILURE	A memory failure was detected in either the 9/PC SERCOS master or one of the 1394 serial drives connected on the fiber optic ring. This condition may occur if the size of the AMP image received by the 1394 CNC Serial Drive is larger than permitted. This condition may be the result of a mismatch between your ODS AMP version and the 1394 SERCOS slave software. In the event that this error occurs, contact your Rockwell Automation support person for assistance.
SERCOS NETWORK OVERLAP	The 9/PC SERCOS master did not complete processing the data for one communications cycle while in the cyclic phase (phase 4) of the SERCOS protocol before another communications cycle started. Until the error is cleared by the SERCOS master software, the 9/PC will remain in E-Stop. Check your system to make sure that there is not a misconfiguration of AMP or you do not have an incorrect version of the 9/PC SERCOS master software installed on your system. In the event that this problem persists, contact your local Rockwell Automation support person for assistance.
SERCOS PHASE 1 NOT COMPLETED	An error occurred in the SERCOS protocol during phase 1 of the PTO run-up sequence. This error appears when communications between the 9/PC SERCOS master and all drives connected on the fiber optic ring failed to reach phase 1 of the protocol within 5s. This message may appear if: • power to the drive is lost or no power was applied • a power up or power down sequence interrupted phase 1 communications • there is a mismatch between the SERCOS node addresses configured in AMP and the actual 1394 CNC Serial Drive node addresses If this error occurs, the 9/PC SERCOS master initializes communications and attempts to complete phase 1 of the SERCOS protocol. Verify that each AMPed device has a SERCOS address that matches the SERCOS address of the physical device located on the fiber optice ring. In the event that this problem persists, contact your local Rockwell Automation support person.

Message	Description
SERCOS PHASE 2 NOT COMPLETED	An error occurred in the SERCOS protocol during phase 2 of the PTO run-up sequence. This message may appear if: • power to the drive is lost • a power up or power down sequence interrupted phase 2 communications • there is a synchronization problem between the 9/PC SERCOS master and a drive on the fiber optic ring If this error occurs, the 9/PC SERCOS master will attempt to restart communications beginning in phase 1 of the protocol. In order to clear this error, power cycle your drive. In the event that this problem persists, contact your local Rockwell Automation support person.
SERCOS PHASE 3 NOT COMPLETED	An error occurred in the SERCOS protocol during phase 3 of the PTO run-up sequence. This message may appear if: • power to the drive is lost • a power up or power down sequence interrupted phase 3 communications • there is a synchronization problem between the 9/PC SERCOS master and a drive on the fiber optic ring If this error occurs, the 9/PC SERCOS master will attempt to restart communications beginning in phase 1 of the protocol. In order to clear this error, power cycle your drive. In the event that this problem persists, contact your local Rockwell Automation support person.
SERCOS PHASE 4 NOT COMPLETED	An error occurred in SERCOS protocol during phase 4 of the PTO run-up sequence. This message may appear if: • power to the drive is lost • a power up or power down sequence interrupted cyclic (phase 4) communications • the servo processor is indicating a fault • the SSRN is indicating a fault If any of these conditions occur, the 9/PC will log a second message providing additional information in conjunction with the original message. In order to clear this error, power cycle your drive. In the event that this problem persists, contact your local Rockwell Automation support person.
SERCOS RING LOST DATA	The 9/PC SERCOS master detected data loss during communications in the cyclic phase (phase 4) of the SERCOS protocol. During cyclic communications, the SERCOS protocol requires the transaction of data between the master and slave to complete at specific, predefined intervals. If the master or slave detects an incomplete communications cycle, the 9/PC goes into E-Stop and the 9/PC attempts to restart SERCOS communications beginning in phase 1. In the event that this problem persists, contact your local Rockwell Automation support person.
SERCOS SERVO POWERUP FAULT	An error occurred during 1394 DSP processor power-up. Possible causes are: • the servo DSP processor did not initialize correctly • unable to set up AMP or flash SIMMS on the 1394 servo DSP loop processor This error indicates a problem with the servo hardware or a compatibility problem between the 1394 SERCOS slave software and the servo software. In the event that this fault occurs, contact your local Rockwell Automation support person.
SERCOS SLAVE POWER FAIL	This condition occurred as a result of one of the following: • power supply in the 1394 CNC Serial Drive failed • the 1394 CNC Serial Drive detected an interruption in power (+24V) • the +24V power supply fell below acceptable levels
SERCOS SLAVE POWERUP FAULT	An error occurred during initialization of the slave 1394 DSP processor. This error indicates a problem with the 1394 SERCOS slave hardware or a compatibility problem between the 9/PC SERCOS master software and the 1394 SERCOS slave software. In the event that this problem persists, contact your local Rockwell Automation support person.
SERCOS MS SYNCH LOST ON 1394	Clock synchronization between the 9/PC SERCOS master and 1394 SERCOS slave was lost. Data was not received or it was received at the wrong time. In order to clear this error, power cycle your drive. Excessive electrical noise or a hardware fault may be the cause of this problem. To correct this problem power down your PC and all devices in the fiber optic ring. Verify the fiber cables are securely fastened and if possible, eliminate any source of unnecessary noise that might trigger the problem. Without applying power to the machine tool, restart the 9/PC and apply power to each of the drives connected on the fiber optic ring. If the error is no longer apparent, restart your system. If this problem persists, contact your local Rockwell Automation support person.

Message	Description
SERCOS SS SYNCH LOST ON 1394	Clock synchronization between the 1394 SERCOS slave and the servo loop processor was lost. Data was not received or it was received at the wrong time. Excessive electrical noise or a hardware fault may be the cause of this problem. To correct this problem power down your PC and all devices in the fiber optic ring. Verify the fiber cables are securely fastened and if possible, eliminate any source of unnecessary noise that might trigger the problem. Without applying power to the machine tool, restart the 9/PC and apply power to each of the drives connected on the fiber optic ring. If the error is no longer apparent, restart your system. If this problem persists, contact your local Rockwell Automation support person.
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.
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. Make sure to check the error log for additional messages that may appear in conjunction with this message.
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.

Message	Description
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.
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 logic 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.
SETUP BUFFER ALLOC ERROR - CNC STOPPING	The 9/PC executive was unable to allocate the setup buffers used for processing part program blocks.
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.

Message	Description
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 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.
SPINDLE IS CLAMPED	An attempt was made to program a block containing a spindle code other than an M05 while the PAL or logic 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.
SSRN COMMUNICATIONS ERROR	A 1394 SERCOS drive reported an error with the system serial communications network (SSRN) that monitors hardware status. Excessive electrical noise could cause the SSRN in the drive to fault. This message may also appear if the terminator connector on the 1394 CNC Serial Drive is loosened or removed while the system is operating. To recover from this condition, remove power to the drive, verify that the terminating connector is attached, and power up the 1394 CNC Serial Drive. If the problem persists, contact your local Rockwell Automation support person.

Message	Description
SSRN NOT INITIALIZED (X)	An SSRN on a 1394 rack was not initialized properly. One possible cause for this message to appear during power-up is if the terminator connector on one of the 1394 CNC Serial drives (X = 0 or 1) is missing. The system remains in E-Stop until initizialization is complete. To recover from the situation, plug the terminator and power cycle the drive.
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.
SWITCHLESS HOME NOT ALLOWED WITH DCM	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. This message may also appear if there was an attempt to use switchless homing on an axis with distance coded marker (DCM) feedback.
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.
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. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
SYSTEM MODULE OVER TEMP	The 1394 contains a thermal sensor which senses the internal ambient temperature. Causes could be: that the cabinet ambient temperature is above rating. The machine duty cycle requires an RMS current exceeding the continuous rating of the controller. The airflow access to the 1394 is limited or blocked. This does not necessarily indicate a motor over temperature. Motor over temperature sensors should be wired directly into the E-Stop string. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.

Message	Description
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 (810 Vdc), a fault is sensed and the power supply is disabled. This can be caused by an under sized shunt requirement, shunt regulator fuse has blown, the shunt regulator transistor has malfunctioned, the power driver board is malfunctioning and incorrectly sensing the bus voltage, the CNC acc/dec rate is incorrectly set, the input line voltage is excessive, the system inertia is too high causing excessive energy to be returned to the power supply bus, or a vertical axis with insufficient counterbalancing is over driving the servomotor and causing excessive energy to be returned to the power supply bus. To clear this fault, you should: • determine the cause of the fault • resolve the situation • power cycle to your 1394 drive • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
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. To clear this fault, you should: • determine the cause of the fault • resolve the situation • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
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 275V dc or less. It will clear at 300V dc. Typical causes include low voltage on the three phase input. • determine the cause of the fault • resolve the situation • turn your system back on For more information regarding this fault, refer to the Troubleshooting chapter of the *1394 Digital AC Multi-Axis Motion Control System User Manual*.
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 1394 DRIVES AMPED	An attempt was made to AMP more than two 1394 SERCOS drives.
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 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.
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.

Message	Description
TOO MANY QPP NONMOTION BLOCKS	Too many nonmotion blocks separate the first and second tool paths with unknown intersections in QuickPath Plus. A maximum number of nonmotion blocks is set in AMP by the system installer. A nonmotion block is defined as any block that does not generate axis motion in the current plane.
TOO MANY SERCOS AXES FOR SCAN TIME	An attempt was made to AMP more SERCOS axes than was allowed by the SERCOS Scan Time. The AMPed foreground scan time must be increased for the number of AMPed 1394 axes.
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 subprograms 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 AMPed 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.
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.

Message	Description
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.
UNKNOWN ERROR - CNC STOPPING	An unknown error condition caused a fatal fault to occur. Contact Allen-Bradley Support Services.
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 control hardware and your system will not report watchdog errors. Call Allen-Bradley field service.
WATCHDOG TIMEOUT - CNC STOPPING	The watchdog timed out due to a possible failure in the 9/PC executive. Contact local Allen-Bradley Support Service.
WATCHLIST ALLOC ERROR - CNC STOPPING	The executive was unable to allocate sufficient memory for watchlists to run the CNC executive.
WHEEL AXIS MOTION INVALID IN G16.3/G16.4	While in the angled wheel grinding mode you have attempted to program the wheel axis directly. Only the virtual axis and the axial axis can be programmed in angled wheel mode.
WILDCARD MUST BE AT START/END OF SYMBOL	You have used incorrect search string syntax in the PAL search monitor utility.
WORK CO-ORD CHANGES NOT ALLOWED	You have attempted to make a change to the work coordinate system at an invalid time. Changes to the work coordinate system can not be performed when some features are active. Disable the offending feature before attempting to change coordinate systems.

Message	Description
Z	
Z-WORD CANNOT BE GREATER THAN R-WORD	The depth (Z-word) of a pocket formed using a G88.5 and G88.6 hemispherical pocket cycle cannot be greater than the radius (R-word) of that pocket.
ZONE 2 PROGRAM ERROR	The next block in the program or MDI entry would cause the specified axis to enter the restricted area of programmable zone 2.
ZONE 2 PROGRAM ERROR:	The current block in the program or MDI entry caused the specified axis to enter the restricted area of programmable zone 2.
ZONE 3 PROGRAM ERROR	The next block in the program or MDI entry would cause the specified axis to enter or exit the area defined as programmable zone 3.
ZONE 3 PROGRAM ERROR:	The current block in the program or MDI entry caused the specified axis to enter the restricted area of programmable zone 3.

END OF APPENDIX

G-code Tables

Appendix Overview

This appendix lists the G-codes for your CNC. They are listed numerically along with a brief description of their use. These G-codes are discussed in detail in the sections within this manual that refer to their specific usage.

G-code Tables

The group numbers given in the table refer to modality. Group 00 are not modal and are independent of other G-codes. The remaining G-code groups are modal with other G-codes with the same group number. This means programming a G-code in group 1 replaces any other active group 1 G-code but does not affect any G-codes in the other group numbers.

A	Modal Group	Function	Type
G00	01	Rapid Positioning	Modal
G01		Linear Interpolation	
G02		Circular/Helical Interpolation (Clockwise)	
G03		Circular/Helical Interpolation (Counterclockwise)	
G04	00	Dwell	Nonmodal
G09		Exact Stop	
G10L2		Setup Work Coordinate Offset Tables	
G10L3		Setup Tool Management Table	
G10L10		Setup Tool Length Values Geometry Table	
G10L11		Setup Tool Length Values Wear Table	
G10L12		Setup Tool Radius Values Geometry Table	
G10L13		Setup Tool Radius Wear Values Wear Table	
G10.1		Setup Random Tool Table	
G11		Setup Tool Management Table (Cancel)	
G12.1	21	Primary Spindle Controlling	Modal
G12.2		Auxiliary Spindle 2 Controlling	
G14	19	Scaling (Disable)	Modal
G14.1		Scaling (Enable)	
G15	15	Polar Coordinate Programming (Cancel)	Modal
G16		Polar Coordinate Programming	
G16.1		Cylindrical Interpolation	
G17	02	Plane Selection	Modal
G18		Plane Selection	
G19		Plane Selection	
G20	06	Inch System Selection	Modal
G21		Metric System Selection	
G22	04	Programmable Zone 2 and 3, ON	Modal
G22.1		Programmable Zone 3, ON	
G23		Programmable Zone 2 and 3, OFF	
G23.1		Programmable Zone 3, OFF	

A	Modal Group	Function	Type
G24	00	Feed to Hard Stop	Nonmodal
G25		Adaptive Feedrate (torque mode)	
G26		Adaptive Depth	
G27		Machine Home Return Check	Nonmodal
G28		Automatic Machine Home	
G29		Automatic Return From Machine Home	
G30		Return to Secondary Home	
G31		External Skip Function 1	
G31.1		External Skip Function 1	
G31.2		External Skip Function 2	
G31.3		External Skip Function 3	
G31.4		External Skip Function 4	
G36	22	Short Block Acc/Dec Check (Enable)	Modal
G36.1		Short Block Acc/Dec Check (Disable)	
G37	00	Tool Gauging Skip, Function 1	Nonmodal
G37.1		Tool Gauging Skip, Function 1	
G37.2		Tool Gauging Skip, Function 2	
G37.3		Tool Gauging Skip, Function 3	
G37.4		Tool Gauging Skip, Function 4	
G38		Circle Diameter and Center Measurement	
G38.1		Parallel Probing Cycle	
G39	20	Cutter Diameter Comp (Linear Generated Block)	Modal
G39.1		Cutter Diameter Comp (Circular Generated Block)	
G40	07	Cutter Diameter Compensation (Cancel)	
G41		Cutter Diameter Compensation (Left)	
G42		Cutter Diameter Compensation (Right)	
G43	08	Tool Length Offset (Plus)	Modal
G43.1		Tool Length Offset Selection (Plus)	
G44		Tool Length Offset (Minus)	
G44.1		Tool Length Offset Selection (Minus)	
G45	23	Disable Spindle Synchronization	Modal
G46		Set Spindle Positional Synchronization	
G46.1		Set Active Spindle Speed Synchronization	
G47	24	Linear Acc/Dec in All Modes	Modal
G47.1		S-Curve Acc/Dec for Positioning and Exact Stop Mode	
G47.9		Infinite Acc/Dec (No Acc/Dec) (AMP-selectable only)	
G48	00	Reset Acc/Dec to Default AMPed Values	Nonmodal
G48.1		Acceleration Ramp for Linear Acc/Dec Mode	
G48.2		Deceleration Ramp for Linear Acc/Dec Mode	
G48.3		Acceleration Ramp for S-Curve Acc/Dec Mode	
G48.4		Deceleration Ramp for S-Curve Acc/Dec Mode	
G48.5		Programmable Jerk Value	
G49	08	Tool Length Offset Cancel)	Modal

A	Modal Group	Function	Type
G50.1	11	Programmable Mirror Image (Cancel)	Modal
G51.1		Programmable Mirror Image	
G52	00	Offsetting Coordinate Zero Point	Nonmodal
G53		Motion in Machine Coordinate System	
G54	12	Preset Work Coordinate System 1	Modal
G55		Preset Work Coordinate System 2	
G56		Preset Work Coordinate System 3	
G57		Preset Work Coordinate System 4	
G58		Preset Work Coordinate System 5	
G59		Preset Work Coordinate System 6	
G59.1		Preset Work Coordinate System 7	
G59.2		Preset Work Coordinate System 8	
G59.3		Preset Work Coordinate System 9	
G60	25	Synchronous Logic/Block Synchronization Mode	Modal
G60.1		Asynchronous Logic/Block Synchronization Mode	
G60.2		Autosynchronous Logic/Block Synchronization Mode	
G61	13	Exact Stop Mode	Modal
G62		Automatic Corner Override	
G63		Tapping Mode	
G64		Cutting Mode	
G65	00	Paramacro Call	Nonmodal
G66	14	Paramacro Modal Call	Modal
G66.1		Paramacro Modal Call	
G67		Paramacro Modal Call (Cancel)	
G68	16	Part Rotation	Modal
G69		Part Rotation (Cancel)	
G73	09	Deep Hole Peck Drilling Cycle (With dwell)	Modal
G74		Left-Hand Tapping Cycle	
G74.1		Left-Hand Solid Tapping Cycle	
G76		Boring Cycle (Spindle Shift)	
G80		Cancel or End Fixed Cycle	
G81		Drilling Cycle (No Dwell, Rapid Out)	
G82		Drilling Cycle (Dwell, Rapid Out)	
G83		Deep Hole Peck Drilling Cycle	
G84		Right-Hand Tapping Cycle	
G84.1		Right-Hand Solid Tapping Cycle	
G85		Boring Cycle (No Dwell, Feed Out)	
G86		Boring Cycle (Spindle Stop, Rapid Out)	
G87		Back Boring Cycle	
G88		Boring Cycle (Spindle Stop, Manual Out)	

A	Modal Group	Function	Type
G88.1	00	Pocket Milling Roughing Cycle	Nonmodal
G88.2		Pocket Milling Finishing Cycle	
G88.3		Post Milling Roughing Cycle	
G88.4		Post Milling Finishing Cycle	
G88.5		Hemispherical Milling (Roughing Cycle)	
G88.6		Hemispherical Milling (Finishing Cycle)	
G89	09	Boring Cycle (With Dwell, Feed Out)	Modal
G89.1	00	Irregular Pocket Milling (Roughing Cycle)	Nonmodal
G89.2		Irregular Pocket Milling (Finishing Cycle)	
G90	03	Absolute Mode	Modal
G91		Incremental Mode	
G92	00	Coordinate System Offset (Using Tool Positions)	Nonmodal
G92.1		Coordinate System Offset (Cancel)	
G92.2		Selected Coordinate System Offsets (Cancel)	
G93	05	Inverse Time Feed Mode	Modal
G94		Feed-per-minute mode	
G95		Feed-per-revolution Mode	
G98	10	Initial Level Return in Milling Cycles	Modal
G99		R-Point Level Return in Milling Cycles	

END OF APPENDIX

Symbols

Numbers

A

B

C

D

E

F

G

H

M

Q

R

S

T

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 8520-9.5 - April 2000

PN 176834