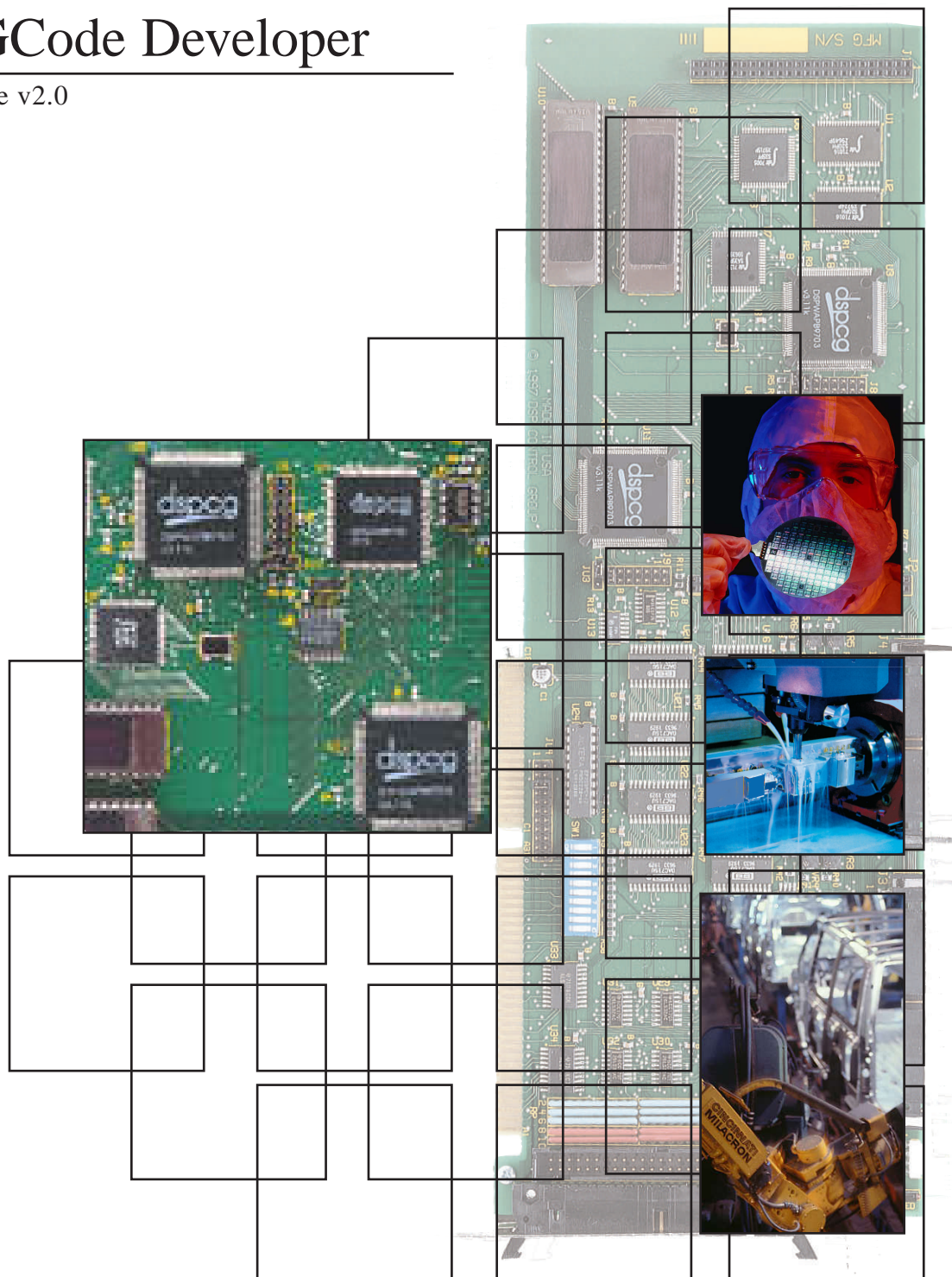


# Mx4 GCode Developer

User's Guide v2.0



**DSPCG**  
DSP Control Group, Inc.

# Contents

|  |     |
|--|-----|
| 1 Introduction .....                                 | 1-1 |
| Mx4 User's Guide .....                               | 1-2 |
| DSPL Programmer's Guide.....                         | 1-2 |
| 2 Installing Mx4 GCode Developer .....               | 2-1 |
| Mx4Pro Development Tools.....                        | 2-1 |
| Mx4 or Mx4 Octavia.....                              | 2-1 |
| Communication Settings .....                         | 2-1 |
| License Number.....                                  | 2-2 |
| Installing the Mx4 GCode Developer .....             | 2-2 |
| 3 Mx4 GCode Programming.....                         | 3-1 |
| G Code and Mx4 .....                                 | 3-1 |
| G Code File Components.....                          | 3-1 |
| G code file [filename.g] .....                       | 3-2 |
| Parameter file [g_dspl.par].....                     | 3-2 |
| Tool diameter compensation file [tooldiam.cmp] ..... | 3-3 |
| Tool length compensation file [toolleng.cmp] .....   | 3-3 |
| M06 tool change DSPL file [tool_chg.hll].....        | 3-4 |
| 4 Using The Development Tools.....                   | 4-1 |
| File Operations .....                                | 4-1 |
| Open GCode.....                                      | 4-2 |
| Refresh GCode .....                                  | 4-2 |

## Contents

|                                       |      |
|---------------------------------------|------|
| Edit GCode.....                       | 4-2  |
| Load GCode .....                      | 4-2  |
| Tools .....                           | 4-3  |
| Tool Chg .....                        | 4-4  |
| Setup .....                           | 4-5  |
| View Setup.....                       | 4-6  |
| X-Y Plane.....                        | 4-7  |
| X-Z Plane .....                       | 4-7  |
| Z-X Plane .....                       | 4-7  |
| Inch Mode .....                       | 4-7  |
| Milli-Meter Mode .....                | 4-7  |
| Graph .....                           | 4-8  |
| Text.....                             | 4-8  |
| Running or Debugging Your GCode.....  | 4-9  |
| Debug/Run Mode .....                  | 4-10 |
| Step GCode .....                      | 4-10 |
| Jog.....                              | 4-11 |
| Run GCode.....                        | 4-11 |
| Stop GCode.....                       | 4-12 |
| Scope.....                            | 4-12 |
| Video.....                            | 4-12 |
| Options .....                         | 4-13 |
| Select Editor .....                   | 4-14 |
| Enable/Disable Warning Messages ..... | 4-14 |
| Reset Mx4 .....                       | 4-14 |

## 5 Mx4 GCode Command Set..... 5-1

|  |     |
|--|-----|
| G Code Summary .....                       | 5-1 |
| M Code Summary .....                       | 5-2 |
| G and M Code Listings.....                 | 5-3 |
| G00 Rapid Motion .....                     | 5-5 |
| G01 Linear Motion .....                    | 5-5 |
| G02 Clockwise Circular Motion.....         | 5-6 |
| G03 Counter-clockwise Circular Motion..... | 5-6 |

|   |      |
|---|------|
| G04 Dwell .....                               | 5-7  |
| G09 Exact Stop Check.....                     | 5-7  |
| G17 Use XY Plane for Circular Motion.....     | 5-7  |
| G18 Use ZX Plane for Circular Motion .....    | 5-7  |
| G19 Use YZ Plane for Circular Motion .....    | 5-7  |
| G20 Millimeter Mode.....                      | 5-7  |
| G21 Inch Mode.....                            | 5-8  |
| G28 Return To Reference Point .....           | 5-8  |
| G29 Set Return Reference Point.....           | 5-8  |
| G40 Cutter Compensation Cancel .....          | 5-8  |
| G41 Cutter Compensation Left .....            | 5-8  |
| G42 Cutter Compensation Right .....           | 5-9  |
| G43 Tool Length Compensation + (plus) .....   | 5-9  |
| G44 Tool Length Compensation – (minus).....   | 5-9  |
| G49 Cancel G43 / G44.....                     | 5-9  |
| G52 Set Local Coordinate System.....          | 5-9  |
| G54-59 Select Work Coordinate System 1-6..... | 5-10 |
| G60 Uni-Directional Positioning .....         | 5-10 |
| G61 Exact Stop Modal .....                    | 5-10 |
| G90 Absolute Positioning.....                 | 5-10 |
| G91 Incremental Positioning .....             | 5-10 |
| G92 Define Relative Origin .....              | 5-11 |
| M00 Stop Program .....                        | 5-11 |
| M01 Optional Program Stop .....               | 5-11 |
| M02 Program End .....                         | 5-11 |
| M03 Spindle Forward (Clockwise).....          | 5-11 |
| M04 Spindle Reverse (Counter-clockwise).....  | 5-11 |
| M05 Spindle Stop .....                        | 5-12 |
| M06 Tool Change (generic PLC program) .....   | 5-12 |
| M60-M91 Output On.....                        | 5-12 |
| M92-M123 Output Off .....                     | 5-12 |

|                          |     |
|--------------------------|-----|
| 6 DSPL & G Code.....     | 6-1 |
| Inserting DSPL Code..... | 6-1 |

## Contents

|                                 |     |
|---------------------------------|-----|
| M06 Tool Change DSPL File ..... | 6-2 |
|---------------------------------|-----|

## 7 GCode Examples..... 7-1

|                          |     |
|--------------------------|-----|
| Example 1: Triangle..... | 7-1 |
| Example 2: Bracket ..... | 7-3 |
| Example 3: A-Shape.....  | 7-5 |

## Appendix A: How The Moves Are Generated..... A-1

|                                       |      |
|---------------------------------------|------|
| Continuous Offset Moves .....         | A-1  |
| Point-to-Point Trapezoidal Move.....  | A-8  |
| Intercepting a Trapezoidal Move ..... | A-9  |
| Point-to-Point S-Curve Move.....      | A-10 |
| Segment Moves With S-Curve.....       | A-11 |
| Trapezoidal Segment Moves.....        | A-12 |
| Cubic Spline Moves.....               | A-13 |
| PVT Hermite Spline .....              | A-14 |
| Blending Moves.....                   | A-15 |
| Linear Move.....                      | A-16 |
| “S” Curve Linear Move .....           | A-17 |
| Start-up Offset Moves .....           | A-18 |

# 1 Introduction

Mx4 GCode Developer is a complete Gcode development tool including a graphical interface (see Fig. 1-1) along with a powerful Gcode to DSPL cross-compiler. The graphical interface includes test/debug features such as STEP option as well as live compensated and uncompensated tool motion profile plots. In addition to Mx4 GCode's many features, it may be used in conjunction with the Mx4pro development tools to provide the user with an unparalleled motion control development platform. As an example, with Mx4pro, user's may observe real-time motion profiles with both time-based and xy scopes as the G code program flow is controlled from the Mx4 GCode interface.

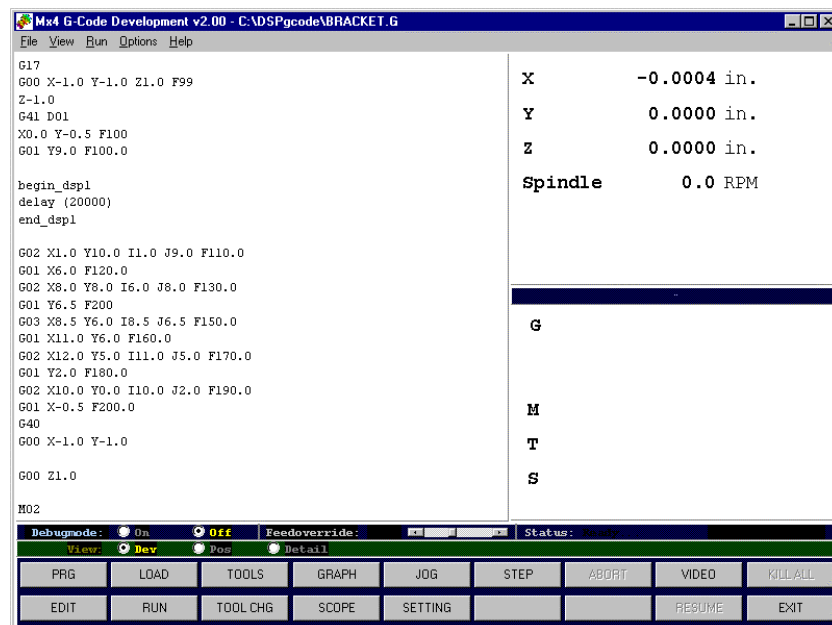


Fig. 1-1: Mx4 GCode Interface

## *Introduction*

This manual, *Mx4 GCode Developer User's Guide*, should be used in conjunction with the following manuals for the Mx4 controller [use associated manuals for Mx42 or Mx4 Octavia]:

### Mx4 User's Guide

This manual includes comprehensive information on Mx4's hardware, system tuning, memory organization, trouble shooting and more. *The Mx4 User's Guide* is the focal point for learning the technical details of the Mx4. All other manuals assume that users have already read the *Mx4 User's Guide*.

### DSPL Programmer's Guide

This manual will assist you with DSPL, DSPCG's high level programming language for the Mx4. DSPL has its own compiler and downloader, which are included as part of the Mx4 GCode and Mx4pro Development Tools.

# 2 Installing Mx4 GCode Developer

This chapter will guide you through the process of installing the Mx4 GCode Developer on a PC running Windows 95 or Windows NT. Before beginning the installation procedure, it is assumed that the Mx4Pro Development Tools have been installed previously on the system.

## Mx4Pro Development Tools

---

Before the Mx4 GCode Developer may be installed and used, you must have installed the Mx4Pro Development Tools software. Installation of the Mx4Pro Development Tools will prompt you to define the following setup parameters:

### Mx4 or Mx4 Octavia

The type of controller which you are using, 4-axis Mx4 or 8-axis Mx4 Octavia, is selected during the Mx4Pro Development Tools setup and installation. This selection will be used by the Mx4 GCode Developer as well.

### Communication Settings

Bus or serial communication? Card address? Slave node address? Hardware interrupt IRQ selection? These parameters are defined by the Mx4Pro Development Tools setup and are likewise used by the Mx4 GCode Developer application.

## License Number

The license number which Mx4Pro Development Tools requires to operate also includes a GCode option. The Mx4 GCode Developer also requires that a valid license number with GCode option enabled be used.

## Installing the Mx4 GCode Developer



**Note:** If a previous version of Mx4 GCode Developer exists on your PC, it is recommended that you first uninstall the previous version before continuing with this installation.

To start the installation procedure of the Mx4 GCode Developer while using Windows 95, select **Run** under the **Start** button and enter SETUP (with the path to the GCode Developer media, CD or diskette), then press the Enter button. You may also double click on the icon that represents the drive and/or folder where the GCode Developer installation media is located (under **My Computer**) and double click on the Setup icon.

To start the setup procedure within Windows NT, select **Run** under the **File** menu of the "Program Manager Window" and enter SETUP (with the path to the GCode Developer media, CD or diskette), then press the Enter button. You may also double click on SETUP.EXE from within the **File Manager** focusing on the drive and/or folder where the GCode Developer installation media is located.

# 3 Mx4 & G Code Programming

## G Code and Mx4

---

The Mx4 family of controllers are typically programmed via Host-Based Programming, in which real time commands (RTCs) are transmitted from the host computer to the Mx4 controller, or high-level language based DSPL Programming, in which motion code is loaded to the memories of the Mx4 controller and executed from there. Mx4 GCode Developer utilizes the DSPL Programming capabilities of the Mx4 controller family. Mx4 GCode Developer consists of a graphical user interface and a powerful cross-compiler which converts G code files to DSPL code files. It is the converted DSPL code file, not the original G code file, which is transmitted to be executed on the Mx4. As with any DSPL program execution, real time commands (RTCs) may be executed even as the DSPL program is executing.

## G Code File Components

---

A G code application using the Mx4 GCode Developer consists of five (5) parts:

- G code file [filename.g]
- parameter file [g\_dspl.par]
- tool diameter compensation file [tooldiam.cmp]
- tool length compensation file [toolleng.cmp]
- M06 tool change DSPL file [tool\_chg.hll]

Each component of an application is described in detail.

## G code file [*filename.g*]

The G code file contains the G and M codes which make up the motion application. It is a text file, and is identified with the .g file name extension. The G code file may contain any of the G and M codes which are supported by Mx4 Gcode [see *chapter 5 Mx4 GCode Command Set*].

### Syntax

G code notations as well as the supported G and M codes are included in chapter 5. The G code file may contain blank lines, and comments may be added to G code files following the semicolon (;). For example,

```
G00 X1.0 Y2.5           ;this is a comment
;G00 X2.0 Y3.3          this entire G code line is commented out
```

## parameter file [*g\_dspl.par*]

The parameter file *g\_dspl.par* contains initialization parameters for the G code file such as acceleration limits and conversion factors. The *g\_dspl.par* file must be located in the same directory (folder) as the installation of the Mx4 Gcode Developer (ie: *Mx4GCode*).

The *g\_dspl.par* file contains the following parameters to be initialized by the user,

|                     |   |
|---------------------|---|
| ACC(X,Y,Z,C)        | Rapid Motion (G00) acceleration         |
| VEL(X,Y,Z,C)        | Rapid Motion (G00) velocity             |
| MAXACC(X,Y,Z,C)     | max acceleration (G01)                  |
| KI(X,Y,Z,C)         | CTRL integral gain                      |
| KP(X,Y,Z,C)         | CTRL proportional gain                  |
| KF(X,Y,Z,C)         | CTRL velocity feedforward gain          |
| KD(X,Y,Z,C)         | CTRL differential gain                  |
| CountsPerInch       | counts per inch conversion factor       |
| CountsPerMilliMeter | counts per millimeter conversion factor |
| Inch_Or_MM_PerRev   | conversion factor for spindle axis      |
| SpindleSpeed        | spindle axis speed                      |
| MaxJogSpeed         | max speed during jog function           |
| System1-6           | work offset coordinate (G54-G59)        |

## tool diameter compensation file [tool diam.cmp]

The compensation file *tooldiam.cmp* contains tool diameter compensation parameters for use with the G41 and G42 cutter compensation G codes. The *tooldiam.cmp* file must be located in the same directory (folder) as the installation of the Mx4 Gcode Developer (ie: *Mx4GCode*).

Tool diameters are stored in the *tooldiam.cmp* file in units of either inches or millimeters (depending on if the G20 or G21 G code is active). The file includes a column of tool numbers and a column of tool diameters for the associated tools. For example,

|   |      |
|---|------|
| 1 | 1.5  |
| 2 | 1.25 |
| 5 | 1.00 |

specifies the tool diameters for tools 1,2, and 5 as 1.5, 1.25, and 1.00, respectively.

The tool diameters are referenced when using the G41 and G42 cutter compensation G codes. For example,

G41 D2

specifies cutter compensation left using tool number 2.

## tool length compensation file [tool leng.cmp]

The compensation file *toolleng.cmp* contains tool length compensation parameters for use with the G43 and G44 cutter compensation G codes. The *toolleng.cmp* file must be located in the same directory (folder) as the installation of the Mx4 Gcode (ie: *Mx4GCode*).

Tool lengths are stored in the *toolleng.cmp* file in units of either inches or millimeters (depending on if the G20 or G21 G code is active). The file includes a column of tool numbers and a column of tool lengths for the associated tools.

## *Mx4 & G Code Programming*

For example,

|   |      |
|---|------|
| 6 | 1.25 |
| 8 | 1.75 |
| 9 | 1.00 |

specifies the tool lengths for tools 6, 8, and 9 as 1.25, 1.75, and 1.00, respectively.

The tool lengths are referenced when using the G43 and G44 tool length compensation G codes. For example,

G44 D8

specifies tool length compensation - using tool number 8.

**M06 tool change DSPL file  
[tool\_chg.hll]**

The M code M06 tool change allows for a generic user-definable tool change program to be called as a subroutine from the main G code file. The M06 M code utilizes the *tool\_chg.hll* DSPL file which contains the framework for including selectable DSPL subroutine calls. See Chapter 6: DSPL & G Code for details.

# 4 Using The Development Tools

This chapter will guide you through the use and operation of the Mx4 GCode Developer graphical user interface.

## File Operations

The Mx4 GCode file operations involve commands or operations which include loading, editing, compiling, and downloading G code or parameter setup files.

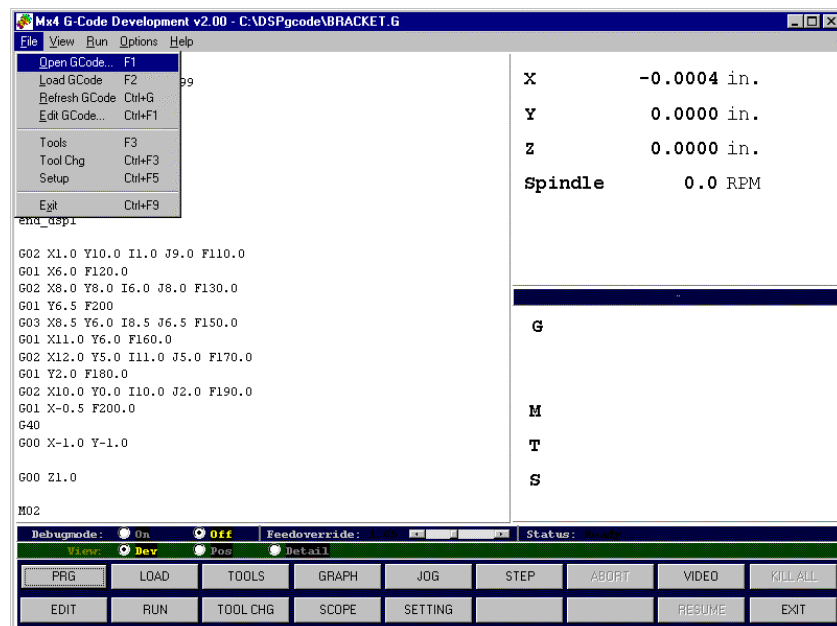


Fig. 4-1: File Menu

## Open GCode

Selects the G code file to open in the development tool. Once a file has been opened, it may be edited, compiled, run, etc. You may open a G-Code ASCII file by either selecting Open G Code in the file menu or pressing PRG key in the main display window. Assuming that the GRAPH window is opened, opening a pre-loaded program will also open its' graphic file. Otherwise to see the graph program file must first be loaded.

## Refresh GCode

If a G code file has been opened, and subsequently edited, the user must issue the Refresh GCode command in order to update the development tool with the edited version of the G code file (i.e.: file updates within the development tool are not made automatically after editing a G code file).

## Edit GCode

This feature is used to invoke the editor on the G code file which is opened in the development tool. The editor may be selected by the options Select Editor command. After a file has been edited, it must be refreshed (see Refresh Gcode) in order for the updated file to be loaded into the development tool.

## Load GCode

Invokes the cross-compiler which converts the G code file into a DSPL (the Mx4's native language) file and downloads this file to the Mx4 controller. This function also generates the XY/YZ/ZX plane view display. Any errors and/or warnings which occur during the compile will be displayed. The download will not be executed.

## Tools

Invokes the Select Tools screen, for entering Tool Diameter, Tool Length also X and Y Offset.

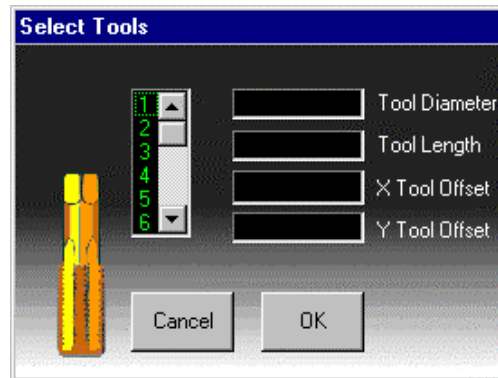
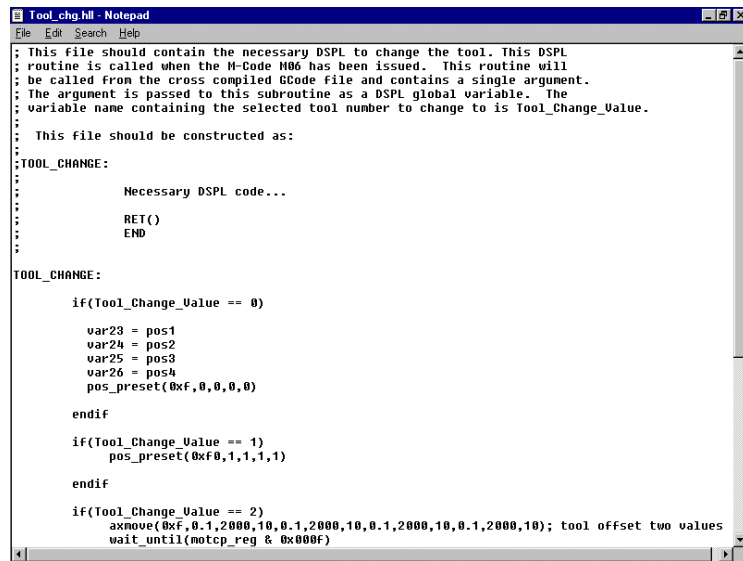


Fig. 4-2: Select Tools

## Tool Chg

Invokes the text editor, loading the tool change DSPL file tool\_chg.hll for editing.



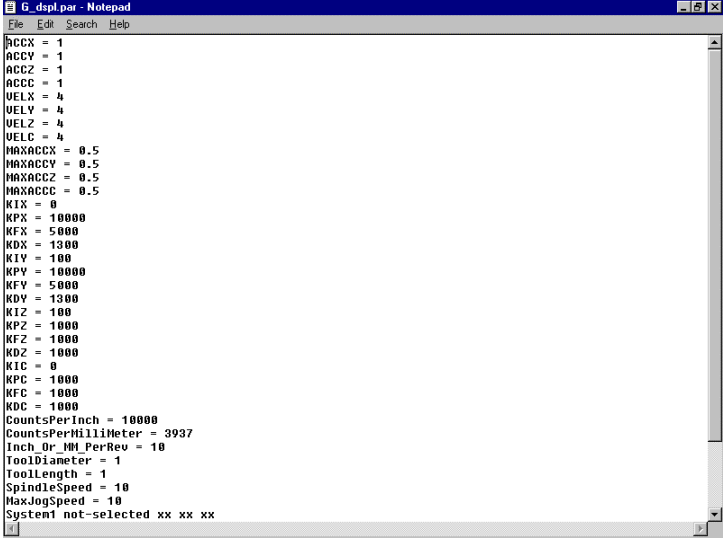
```
Tool_chg.hll - Notepad
File Edit Search Help

; This file should contain the necessary DSPL to change the tool. This DSPL
; routine is called when the M-Code M06 has been issued. This routine will
; be called from the cross compiled GCode file and contains a single argument.
; The argument is passed to this subroutine as a DSPL global variable. The
; variable name containing the selected tool number to change to is Tool_Change_Value.
;
; This file should be constructed as:
;
; TOOL_CHANGE:
;
;     Necessary DSPL code...
;
;     RET()
;     END
;
TOOL_CHANGE:
    if(Tool_Change_Value == 0)
        var23 = pos1
        var24 = pos2
        var25 = pos3
        var26 = pos4
        pos_preset(0xf,0,0,0,0)
    endif
    if(Tool_Change_Value == 1)
        pos_preset(0xf,1,1,1,1)
    endif
    if(Tool_Change_Value == 2)
        axmove(0xf,0.1,2000,10,0.1,2000,10,0.1,2000,10); tool offset two values
        wait_until(notcp_reg & 0x000f)
```

Fig. 4-3: Tool Chg

## Setup

Invokes the text editor, loading the parameter file g\_dspl.par for editing.



```
G_DSPLX = 1
G_DSPLY = 1
G_DSPLZ = 1
G_DSPLC = 1
VELX = 4
VELY = 4
VELZ = 4
VELC = 4
MAXACCX = 0.5
MAXACCV = 0.5
MAXACCZ = 0.5
MAXACCC = 0.5
KIX = 0
KFX = 10000
KFY = 5000
KDX = 1300
KDY = 100
KPY = 10000
KFV = 5000
KDV = 1300
KIZ = 100
KFZ = 1000
KFZ = 1000
KDZ = 1000
KIC = 0
KPC = 1000
KFC = 1000
KDC = 1000
CountsPerInch = 10000
CountsPerMilliMeter = 3937
Inch_Or_MM_PerRev = 10
ToolDiameter = 1
ToolLength = 1
SpindleSpeed = 10
MaxJogSpeed = 10
System1 not-selected xx xx xx
```

Fig. 4-4: Setup

## View Setup

The Mx4 GCode view setup operations involve commands or operations to setup and tailor the GCode graphical interface per the user's requirements.

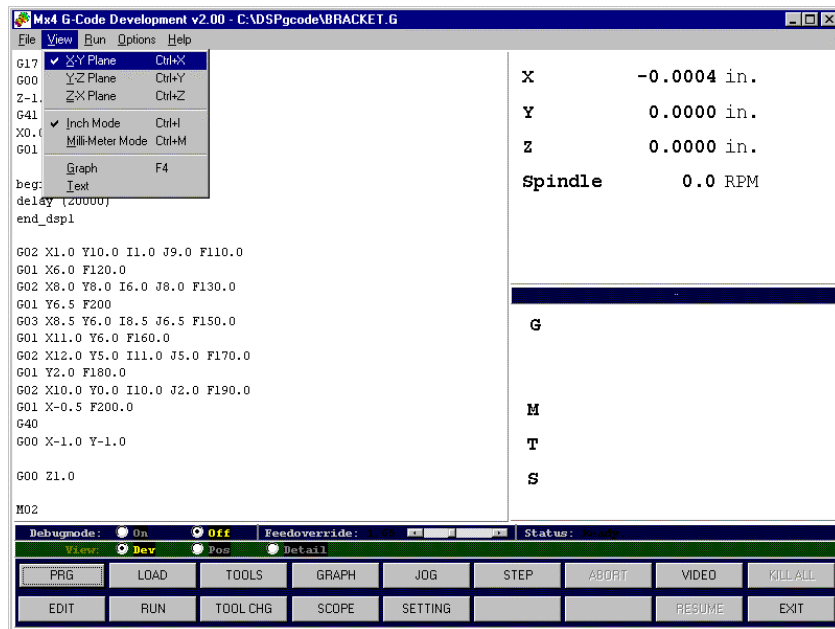


Fig. 4-5: View Menu

## X-Y Plane

Selects the XY plane to be displayed in the motion/compensated path display window. The size of the displayed path may be increased by left-clicking the mouse within the display window, or decreased by right-clicking within the display window. The display may be shifted within the window by 'dragging' the mouse within the window (holding down the left mouse key) in the direction which the path display is to be shifted.

## Y-Z Plane

Selects the YZ plane to be displayed in the motion/compensated path display window. The size of the displayed path may be increased by left-clicking the mouse within the display window, or decreased by right-clicking within the display window. The display may be shifted within the window by 'dragging' the mouse within the window (holding down the left mouse key) in the direction which the path display is to be shifted.

## Z-X Plane

Selects the ZX plane to be displayed in the motion/compensated path display window. The size of the displayed path may be increased by left-clicking the mouse within the display window, or decreased by right-clicking within the display window. The display may be shifted within the window by 'dragging' the mouse within the window (holding down the left mouse key) in the direction which the path display is to be shifted.

## Inch Mode

Selects inch units for the X/Y/Z/Spindle position display.

## Milli-Meter Mode

Selects millimeter units for the X/Y/Z/Spindle position display.

## Graph

Invokes the Graph Pop-up window which displays the Part and Tool Path.

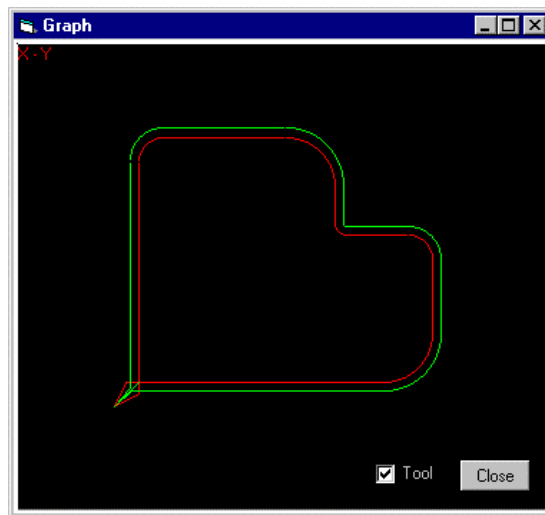


Fig. 4-6: Graph

## Text

Selects text display size on the graphical interface.

# Running or Debugging Your G Code

The Run menu includes the commands to control the execution of G code in the run or debug mode of operation.

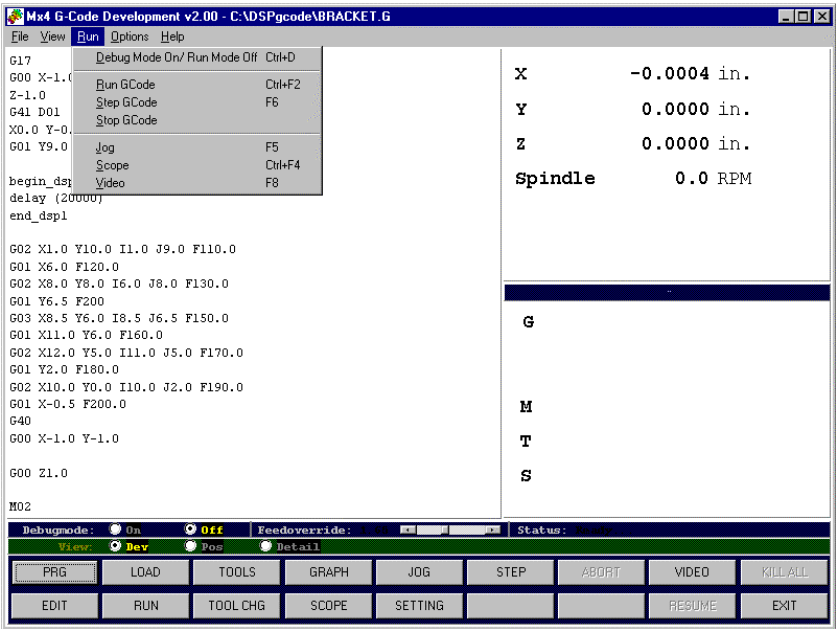


Fig. 4-7: Run Menu

## Debug Mode On / Run Mode Off Debug Mode Off / Run Mode On

Turns debug mode on and run mode off, and vice versa. Debug mode is used when the user is using the Step execution command. When a G code file is compiled in debug mode, the user may step execute through the individual lines of the G code file with the Step command (of course, the file must be downloaded to the Mx4 first). The Step command is used in Debug Mode : On. To execute a G code file in run mode, the run mode must be turned on (and thus Debug Mode : Off), the file is compiled and downloaded, and the execution is initiated via the Run command.

## Step GCode

Step is used in debug mode to execute individual lines of a G code file, controlled by the user. Before Step can be used, the development tool must be placed in debug mode, and the G code file must be compiled and downloaded to the Mx4 controller. As the user then executes the Step command, a line of G code is executed. Subsequent G code lines may be executed by repeated execution of the Step command. The G code line which is 'active' is hi-lighted in the file text window.

## Jog

Invokes Jog Pop-up window which allows the user to manually jog the axes and set Work Offset Coordinate for G54 through G59 from within the development tool.

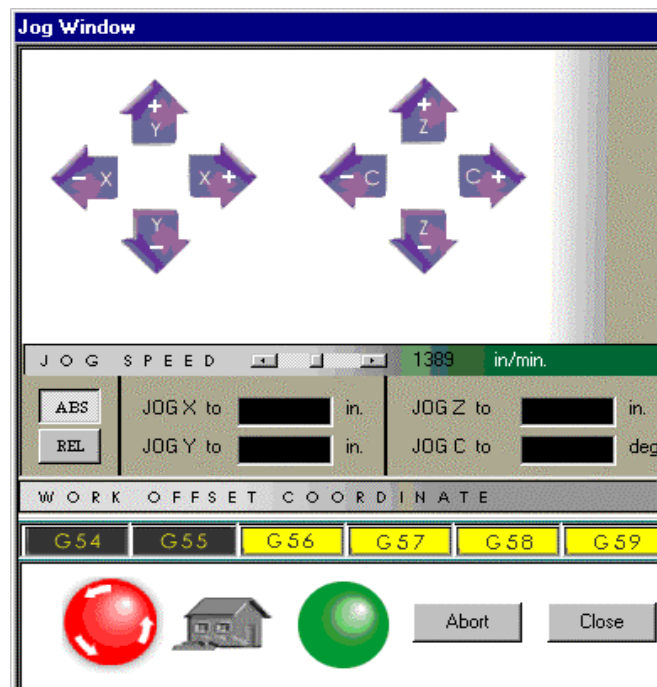


Fig. 4-8: Jog Window

## Run GCode

Run GCode is used to initiate the execution of a G code file which has been compiled (in run mode) and downloaded to Mx4. Again, note that the development tool must be in run mode before the Run GCode command can be

## *Using the Developer*

used. As the G code file is being executed, the 'active' G code line is hi-lighted in the file text window.

## Stop GCode

Used to stop the execution of a G code file. Stop GCode reverses the effect of the Run GCode command.

## Scope

Enables multiple dual trace time and xy scopes that helps user's through the development stage.

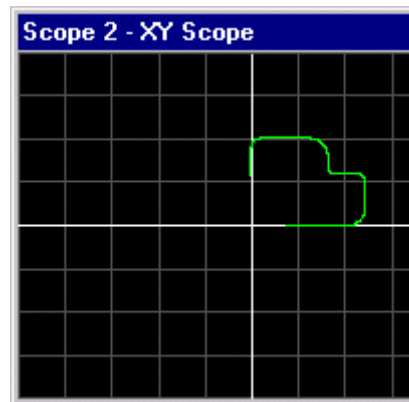


Fig. 4-9: XY Scope

## Video

Invokes the Remote Video Camera Driver to view the actual system while in operation.



**Note :** Once development of a G code program is completed, the developed program may be run outside of the development tool interface. Remember that once the G code file is compiled, it is in the form of a DSPL file. DSPL files may be downloaded to the memories of the Mx4 controller, and run on the controller without the presence of the development tool interface or host computer (see *DSPL Programmer's Guide*).

## Options

Options in the Mx4 GCode development tool include selecting the file editor, resetting the Mx4 controller, and enabling the compiling warning messages.

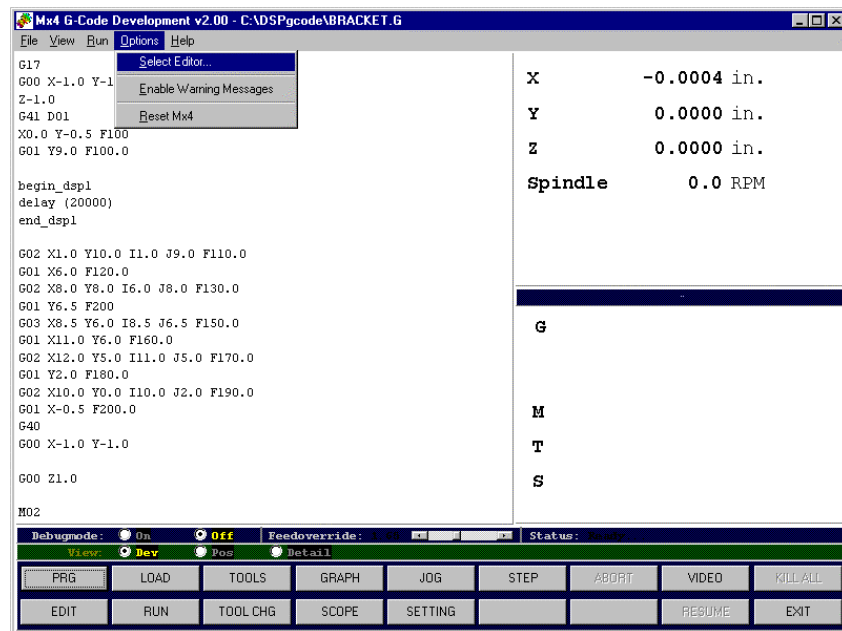


Fig. 4-10: Options Menu

## Select Editor

The file editor used (see File Operations) can be selected by the user via the Select Editor command. Gcode files, as well as the parameter and compensation files are text files, so any text editor may be used.

## Enable/Disable Warning Messages

Allows the operator to turn off/on the automatic Warning Message Pop-up window that appears when the cross-compiler is executed.

## Reset Mx4

Resets the Mx4 controller.

# 5 Mx4 GCode Command Set

The Mx4 GCode command set encompasses both G and M codes.

## G Code Summary

---

|     |                                   |
|-----|-----------------------------------|
| G00 | Rapid Motion                      |
| G01 | Linear Motion                     |
| G02 | Clockwise Circular Motion         |
| G03 | Counter-Clockwise Circular Motion |
| G04 | Dwell                             |
| G09 | Exact Stop Check                  |
| G17 | Use XY Plane for Circular Motion  |
| G18 | Use ZX Plane for Circular Motion  |
| G19 | Use YZ Plane for Circular Motion  |
| G20 | Millimeter Mode                   |
| G21 | Inch Mode                         |
| G28 | Return to Reference Point         |
| G29 | Set Return Reference Point        |
| G40 | Cutter Compensation Cancel        |
| G41 | Cutter Compensation Left          |
| G42 | Cutter Compensation Right         |
| G43 | Tool Length Compensation +        |
| G44 | Tool Length Compensation -        |
| G45 | Tool Offset Increase              |
| G46 | Tool Offset Decrease              |
| G47 | Tool Offset Doubled Increase      |
| G48 | Tool Offset Doubled Decrease      |
| G49 | Cancel G43 and G44                |
| G52 | Set Local Coordinate System       |
| G54 | Select Work Coordinate System 1   |
| G55 | Select Work Coordinate System 2   |
| G56 | Select Work Coordinate System 3   |

#### *Mx4 GCode Command Set*

|        |  |
|--------|--|
| G57    | Select Work Coordinate System 4          |
| G58    | Select Work Coordinate System 5          |
| G59    | Select Work Coordinate System 6          |
| G60    | Uni-Directional Positioning              |
| G61    | Exact Stop Modal                         |
| G70-72 | Bolt Hole Circle, ARC and Along an Angle |
| G90    | Absolute Positioning                     |
| G91    | Incremental Positioning                  |
| G92    | Define Relative Origin                   |

## M Code Summary

---

|         |                                     |
|---------|-------------------------------------|
| M00     | Stop Program                        |
| M01     | Optional Program Stop               |
| M02     | Program End                         |
| M03     | Spindle Forward (Clockwise)         |
| M04     | Spindle Reverse (Counter-Clockwise) |
| M05     | Spindle Stop                        |
| M06     | Tool Change                         |
| M60-91  | Output On                           |
| M92-123 | Output Off                          |

## G and M Code Listings

---

The following notation will be used:

**C** Auxiliary external rotary axis

The C address character is used to specify motion for the optional external axis C. It specifies an angle in degrees for the rotary axis. It is always followed by a signed number and up to three fractional decimal positions.

**D** Tool diameter selection

The D address character is used to select the tool diameter or radius used for cutter compensation. The number following must be between 0 and 50.

**F** Feed rate

The F address character is used to select the feed rate applied to any interpolation functions. It is either in inches per minute with four fractional positions or mm per minute with three fractional positions.

**H** Tool length offset selection

The H address character is used to select the tool length offset entry from the offsets memory. The number following must be between 0 and 50.

**I** Center distance from the starting point of the arc in x direction

It is either in inches with four fractional positions or mm with three fractional positions.

#### *Mx4 GCode Command Set*

|          |   |
|----------|---|
| <b>J</b> | Center distance from the starting point of the arc in y direction<br><br>It is either in inches with four fractional positions or mm with three fractional positions.   |
| <b>P</b> | Delay time<br><br>The P address character is used to enter a time in seconds (for a G04 dwell).   |
| <b>S</b> | Spindle speed<br><br>Indicates the percentage of peak speed (1-100).  |
| <b>T</b> | Tool selection code<br><br>The T address character is used to select the tool for the next tool change. The number following must be between 1 and 16.  |
| <b>X</b> | Linear X-axis motion<br><br>The X address character is used to specify motion for the X-axis. It specifies a position or distance along the X-axis. It is either inches with four fractional positions or mm with three fractional positions. |
| <b>Y</b> | Linear Y-axis motion<br><br>The Y address character is used to specify motion for the Y-axis. It specifies a position or distance along the Y-axis. It is either inches with four fractional positions or mm with three fractional positions. |
| <b>Z</b> | Linear Z-axis motion<br><br>The Z address character is used to specify motion for the Z-axis. It specifies a position or distance along the Z-axis. It is either inches with four fractional positions or mm with three fractional positions. |

## G00 Rapid Motion

|   |  |
|---|--|
| X | Motion command in x direction (Optional) |
| Y | Motion command in y direction (Optional) |
| Z | Motion command in z direction (Optional) |

This G code provides a rapid traverse of three axes of the machine. The auxiliary C (rotary axis) can also be moved by this command. This G code is modal so that a previous block with G00 causes all following blocks to be rapid motions until another group (e.g. G01) is specified. The rapid traverse rate is dependent on the maximum speed specified for each axis (g\_dspl.par).

Generally, rapid motions will not be in straight lines. All of the axes specified are moved at the same time but will not necessarily complete their motions at the same time. The block will wait until all motions are complete.

## G01 Linear Motion

|   |  |
|---|--|
| F | Feed rate in inches (mm) per minute      |
| X | Motion command in x direction (Optional) |
| Y | Motion command in y direction (Optional) |
| Z | Motion command in z direction (Optional) |

This G code provides straight linear motion from a (one to three dimensional) point to another. All axes start and stop at the same time, whereas with G00 although axes start simultaneously they don't necessarily finish together.

The rotary axis, C, may be commanded by G01. When C is commanded the resulting motion trajectory will be helical. The speed of all axes are specified by feed rate (F command) so that the helical path will have consistent speed. Rotary axis speed depends on rotary axis diameter setting.

## G02 Clockwise Circular Motion

|   |  |
|---|--|
| F | Feed rate in inches (mm) per minute                  |
| I | Distance along x axis to center of circle (Optional) |
| J | Distance along y axis to center of circle (Optional) |
| K | Distance along z axis to center of circle (Optional) |
| X | Motion command in x direction (Optional)             |
| Y | Motion command in y direction (Optional)             |
| Z | Motion command in z direction (Optional)             |

This G code provides clockwise circular motion of two of the axes. The G codes G17, G18 and G19 determine the circular planes in XY, ZX or YZ directions respectively. The X, Y and Z are used to specify the end point of the motion. If arguments for selected X, Y or Z are not specified the end point of the arc will be the same as the starting point for that axis. The center for the arc in X, Y or Z directions is specified by I, J or K respectively. These values specify the distant form the center of the arc to the starting point of the arc.

## G03 Counter-clockwise Circular Motion

|   |  |
|---|--|
| F | Feed rate in inches (mm) per minute                  |
| I | Distance along x axis to center of circle (Optional) |
| J | Distance along y axis to center of circle (Optional) |
| K | Distance along z axis to center of circle (Optional) |
| X | Motion command in x direction (Optional)             |
| Y | Motion command in y direction (Optional)             |
| Z | Motion command in z direction (Optional)             |

This G code provides counter-clockwise circular motion of two of the axes. The G codes G17, G18 and G19 determine the circular planes in XY, ZX or YZ directions respectively. The X, Y and Z are used to specify the end point of the motion. If arguments for selected X, Y or Z are not specified the end point of the arc will be the same as the starting point for that axis. The center for the arc in X, Y or Z directions is specified by I, J or K respectively. These values specify the distant form the center of the arc to the starting point of the arc.

## G04 Dwell

P      The dwell time in seconds

This G code provides delay in the program. The block containing G04 will delay for the time specified in the P code.

## G09 Exact Stop Check

This G code provides exact stop. Using this code, makes rapid and interpolated moves decelerate to a stop before the next block is processed. This command is non-modal which means the following blocks are not affected by it. In exact stop, moves take longer time when compared to continuous cutter motion.

## G17 Use XY plane for circular motion

This mode is modal and applies to all following circular motions (see G02 and G03).

## G18 Use ZX plane for circular motion

This mode is modal and applies to all following circular motions (see G02 and G03).

## G19 Use YZ plane for circular motion

This mode is modal and applies to all following circular motions (see G02 and G03).

## G20 Millimeter Mode

G20 sets the position unit for all subsequent G code lines to millimeter mode. For example, X3.0 is interpreted as 3 mm.

## G21 Inch Mode

G21 sets the position unit for all subsequent G code lines to inch mode. For example, X3.0 is interpreted as 3 inches.

## G28 Return To Reference Point

The G28 code is used to return to the machine zero position on all axes. If an X, Y, or Z code is specified on the same block, only those axes will be moved and they will be moved to the specified positions in the current coordinate system and then they will be moved to machine zero. The intermediate position, if specified, is saved for use in the G29. If no X, Y, or Z is specified, all axes will be moved directly to machine zero.

## G29 Set Return Reference Point

The G29 code is used to move the axes to a position via a previously –set reference point. The reference is defined with the G29. This command is normally given with the axes positioned at machine zero. The axes which are selected in this block are moved first to the intermediate reference point and then they are moved to the X, Y, or Z specified. The positions are interpreted in the current coordinate system.

## G40 Cutter Compensation Cancel

G40 cancels cutter compensation G41 and G42.

## G41 Cutter Compensation Left

This G code selects cutter compensation left. That is, the tool is moved to the left of the programmed path to compensate for the diameter of the tool. A Dxx (where xx is an integer value) must be programmed to select the tool size from the setup file.

## G42 Cutter compensation Right

This G code selects cutter compensation right. That is, the tool is moved to the right of the programmed path to compensate for the diameter of the tool. A Dxx (where xx is an integer value) must be programmed to select the tool size from the setup file.

## G43 Tool length compensation +(plus)

This G code selects tool length compensation in a positive direction. That is, the tool length offset is added to the Z axis position command. A non-zero Hxx (where xx is an integer value) must be programmed to select the proper offset from the setup file.

## G44 Tool length compensation –(minus)

This G code selects tool length compensation in negative direction. That is, the tool length offset is subtracted from the Z axis position command. A non-zero Hxx (where xx is an integer value) must be programmed to select the proper offset from the setup file.

## G49 cancel G43 / G44

This G code cancels the tool length compensation. H0 will also cancel tool length compensation.

## G52 Set local coordinate system

This G code selects the work coordinate systems which can be set with G92.

## G54-59 Select work coordinate system 1-6

These codes select one of the six user coordinate systems stored within the offsets memory. All subsequent references to axes positions will be interpreted in the new coordinate system.

## G60 Uni-Directional Positioning

This G code is used to provide positioning always from the plus direction. It is non-modal so does not effect the following blocks.

## G61 Exact Stop Modal

Rapid and interpolated moves will decelerate to an exact stop before another block is processed. It is modal and therefore it affects the following blocks.

## G90 Absolute Positioning

This G code enforces absolute positioning in the user's selected coordinate. It is modal, thus affects the following blocks until otherwise (G91) specified. Each axis which is moved will be placed at the position coded in the command block.

## G91 Incremental Positioning

This G code enforces incremental positioning in the user's selected coordinate. It is modal, thus affects the following blocks until otherwise (G90) specified. Each axis which is moved will be moved by the amount coded in the command block.

## G92 Define relative origin

This command does not move any of the axes; it only changes the values stored as user work offset. Yasnac style of coordinates: all of the values of X, Y, and Z in this block are stored into the offset entry for G52. Fanuc style of coordinates: uses G92 to change all the user work coordinates.

## M00 Stop program

This code is used to stop the program. It also stops the spindle. The program pointer will advance to the next block and stop. A cycle start will continue program operation from the next block.

## M01 Optional Program Stop

Same as M00, except that it works only if the OPTIONAL STOP is turned on (user may set the DSPL variable defined in filename.hll as Optional\_Stop)

## M02 Program end

Same as M00, except program pointer is not advanced to the next block.

## M03 Spindle Forward (Clockwise)

Starts spindle in clockwise direction at the speed set in g\_dspl.par. The block is delayed until spindle reaches 90% commanded speed. This M code can be used in conjunction with the S parameter which will allow the speed of the spindle to be programmed as a % of peak speed.

## M04 Spindle Reverse (Counter-Clockwise)

Starts spindle in counter-clockwise direction at the speed set in g\_dspl.par. The block is delayed until spindle reaches 90% of commanded speed. This M code can be used in conjunction with the S parameter which will allow the speed of the spindle to be programmed as a % of peak speed.

## M05 Spindle Stop

This code is used to stop the spindle. The block is delayed until the spindle slows below 10 RPM.

## M06 Tool change (Generic PLC Program)

This M code implements tool change as specified by Tnn where nn is the tool number. Essentially, a generic PLC subroutine specified by the user can be called based on the tool number. See *Mx4 GCode Programming: M06 tool change DSPL file [tool\_chg.hll]*.

## M60-M91 Output On

Turn Mx4 controller discrete output on. M60-M91 correspond to outputs OUTP1-OUTP32, respectively. For example, M62 turns on output OUTP3. See *Mx4 User's Guide* chapter 2 for hardware information on Mx4's outputs.

## M92-M123 Output Off

Turn Mx4 controller discrete output off. M92-M123 correspond to outputs OUTP1-OUTP32, respectively. For example, M92 turns off output OUTP1. See *Mx4 User's Guide* chapter 2 for hardware information on Mx4's outputs.

# 6 DSPL & G Code

The Mx4 GCode Developer allows the user the ability of combining standard G and M codes along with DSPL command lines and subroutines, yielding a very flexible and powerful application programming environment.

## Inserting DSPL Code

---

DSPL code may be inserted directly into a G code file using the `begin_dspl` and `end_dspl` identifiers. The entire DSPL command set is available to the G code programmer. Pre-defined macros, subroutines, IO routines and interrupt scanning are just a few examples of DSPL programming within G code.

A simple syntax example,

```
G01 X4.5
begin_dspl
POS_PRESET (0x2, 1000)
CTRL (0x2, 0,1200,500,3000)
end_dspl
G00 Y3.0
```

The `POS_PRESET` and `CTRL` DSPL lines will be inserted between the DSPL generated from the G01 and G00 G code lines.

When the G code blocks encounter a `begin_dspl` / `end_dspl` block, the program flow in that section is determined by the DSPL code in the block. When the `end_dspl` block is executed, the G code program continues as normal.

## M06 Tool Change DSPL File

---

The M code M06 tool change allows for a generic user-definable tool change program to be called as a subroutine from the main G code file. The M06 M code utilizes the *tool\_chg.hll* DSPL file which contains the framework for including selectable DSPL subroutine calls. The *tool\_chg.hll* file must be located in the same directory (folder) as the installation of the Mx4 Gcode (ie: *Mx4GCode*).

An example *tool\_chg.hll* DSPL file,

```
;This file should contain the necessary DSPL code to change the
;tool. This DSPL routine is called when the M code M06 has been
;issued. This routine will be called from the cross-compiled G
;code file and contains a single argument. The argument is passed
;to this subroutine as a DSPL global variable. The variable name
;containing the selected tool change routine is Tool_Change_Value.

;This file should be constructed within the following framework,
;
;TOOL_CHANGE:
;
;    Necessary DSPL code...
;
;    RET()
;    END
;
;For example,

TOOL_CHANGE:

    if(Tool_Change_Value == 1)

        axmove(0xf,0.1,1000,10,0.1,1000,10,0.1,1000,10,0.1,1000,10)
        wait_until(motcp_reg & 0x000f)
        pos_preset(0xf,0,0,0,0)
    endif

    if(Tool_Change_Value == 2)

        axmove(0xf,0.1,2000,10,0.1,2000,10,0.1,2000,10,0.1,2000,10)
        wait_until(motcp_reg & 0x000f)
        pos_preset(0xf,0,0,0,0)
    endif

RET()
END
```

### *DSPL & G Code*

The above *tool\_chg.hll* DSPL file contains code for tool change 1 and tool change 2 routines. The Tool\_Change\_Value variable is 'passed' into the DSPL code via the M06 M code T notation. For example,

```
T2M06
```

would initiate the tool change DSPL code TOOL\_CHANGE with Tool\_Change\_Value equal to 2.

*DSPL & G Code*

This page intentionally blank.

# 7 GCode Examples

## Example 1: Triangle

---

The following example, triangle.g, is included with the Mx4 GCode Developer installation. This example includes cutter compensation. Fig. 7-1 illustrates the example loaded into the development tool.

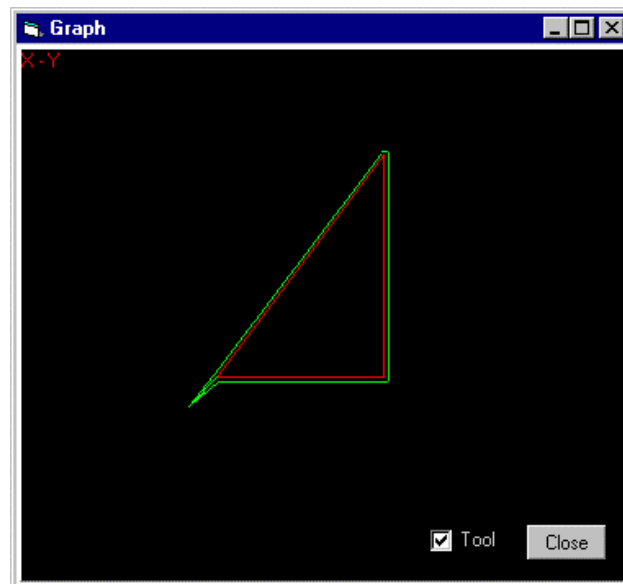


Fig. 7-1: Example 1, Triangle Graph

## GCode Examples

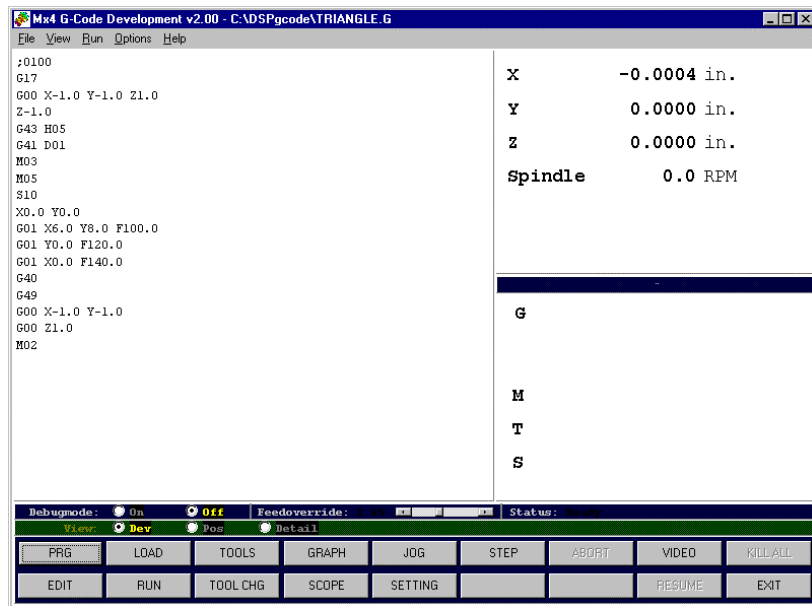


Fig. 7-2: Example 1, Triangle

```

0100
G17
G00 X-1.0 Y-1.0 Z1.0
Z-1.0
G43 H05
G41 D01
M03
M05
S10
X0.0 Y0.0
G01 X6.0 Y8.0 F100.0
G01 Y0.0 F120.0
G01 X0.0 F140.0
G40
G49
G00 X-1.0 Y-1.0
G00 Z1.0
M02

```

## Example 2: Bracket

---

The bracket example, bracket.g, utilizes circular moves (G02, G03). Fig. 7-4 illustrates the example loaded into the development tool.

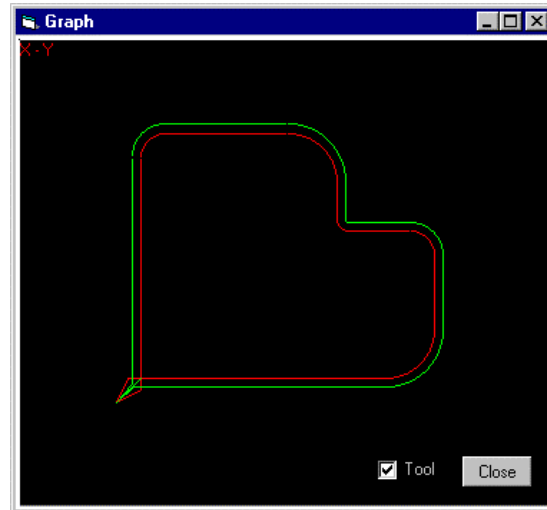


Fig. 7-3: Example 2, Bracket Graph

## GCode Examples

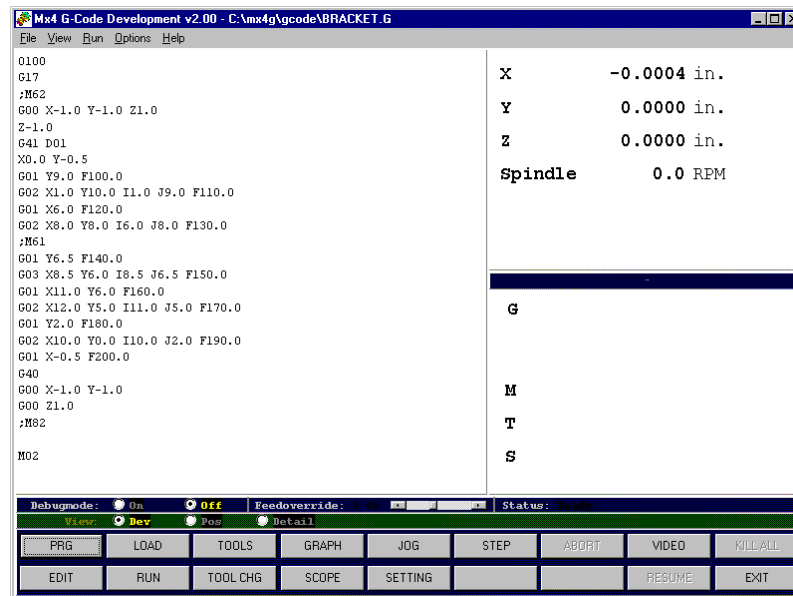


Fig. 7-4: Example 2, Bracket

```

O100
G17
G00 X-1.0 Y-1.0 Z1.0
Z-1.0
G41 D01
X0.0 Y-0.5
G01 Y9.0 F100.0
G02 X1.0 Y10.0 I1.0 J9.0 F110.0
G01 X6.0 F120.0
G02 X8.0 Y8.0 I6.0 J8.0 F130.0
G01 Y6.5 F200
G03 X8.5 Y6.0 I8.5 J6.5 F150.0
G01 X11.0 Y6.0 F160.0
G02 X12.0 Y5.0 I11.0 J5.0 F170.0
G01 Y2.0 F180.0
G02 X10.0 Y0.0 I10.0 J2.0 F190.0
G01 X-0.5 F200.0
G40
G00 X-1.0 Y-1.0
G00 Z1.0
M02

```

## Example 3: A-Shape

---

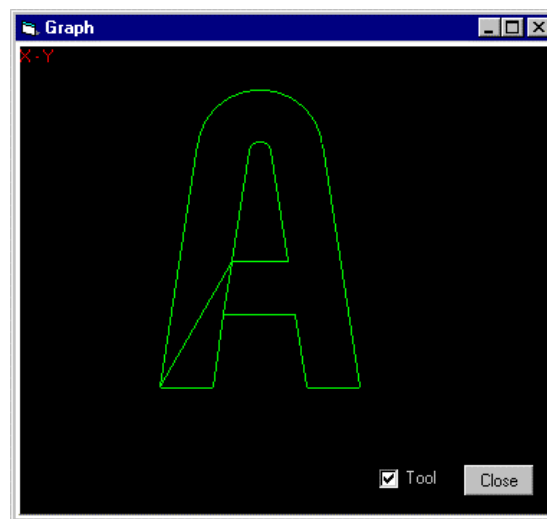


Fig. 7-5: Example 3, A-Shape Graph

## GCode Examples

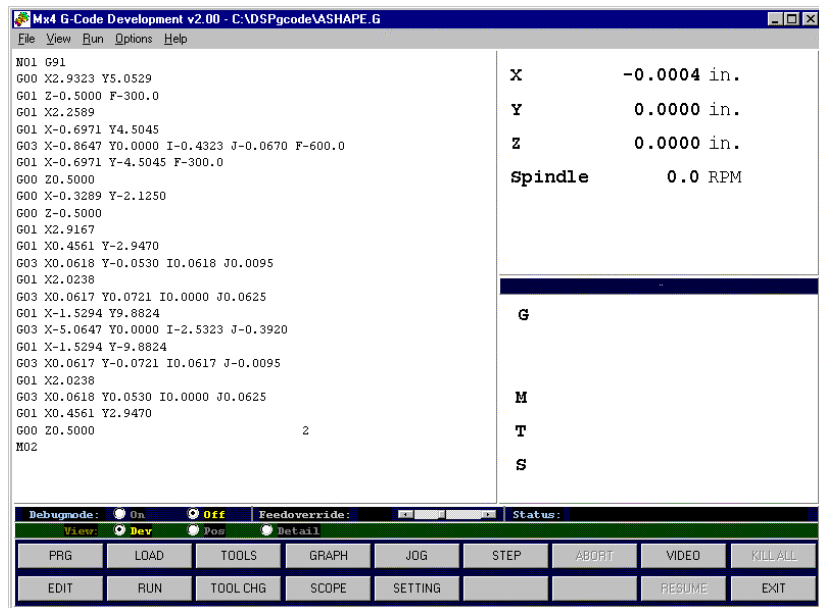


Fig. 7-6: Example 3, A-Shape

```

N01 G91
G00 X2.9323 Y5.0529
G01 Z-0.5000 F-300.0
G01 X2.2589
G01 X-0.6971 Y4.5045
G03 X-0.8647 Y0.0000 I-0.4323 J-0.0670 F-600.0
G01 X-0.6971 Y-4.5045 F-300.0
G00 Z0.5000
G00 X-0.3289 Y-2.1250
G00 Z-0.5000
G01 X2.9167
G01 X0.4561 Y-2.9470
G03 X0.0618 Y-0.0530 I0.0618 J0.0095
G01 X2.0238
G03 X0.0617 Y0.0721 I0.0000 J0.0625
G01 X-1.5294 Y9.8824
G03 X-5.0647 Y0.0000 I-2.5323 J-0.3920
G01 X-1.5294 Y-9.8824
G03 X0.0617 Y-0.0721 I0.0617 J-0.0095
G01 X2.0238
G03 X0.0618 Y0.0530 I0.0000 J0.0625
G01 X0.4561 Y2.9470
G00 Z0.5000
M02
  
```

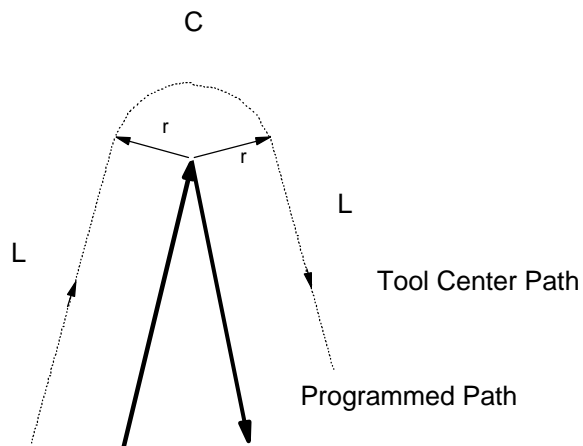
# Appendix A:

## How The Moves Are Generated

---

### Continuous Offset Moves

(A G-Code Strategy)

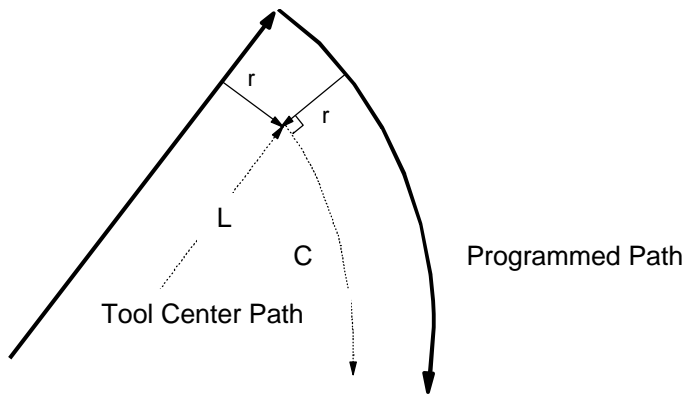


Moving around the outside of an acute angle. Going from one line to another

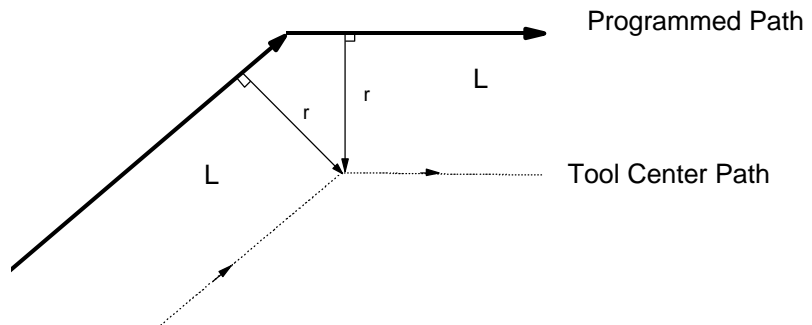
L: Linear  
C: Circle  
r: Radius

## Continuous Offset Moves

(A G-Code Strategy)



Moving around the inside of an acute angle. Going from line to circle

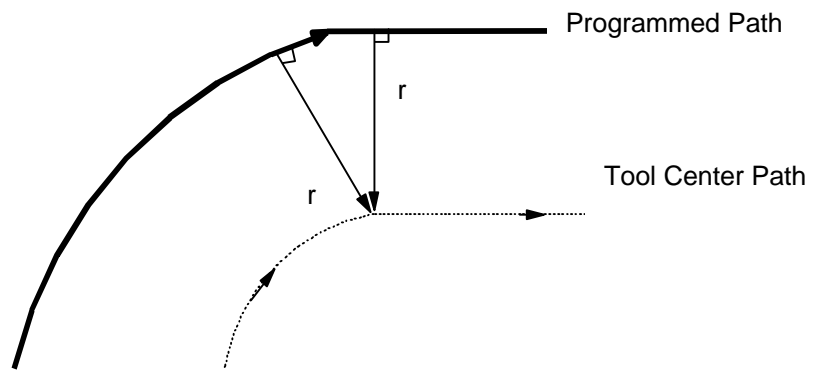


Moving around the inside of an obtuse angle. Going from line to line

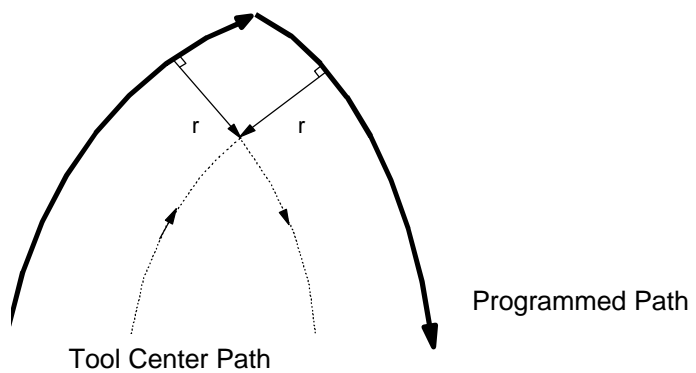
L: Linear  
C: Circle

## Continuous Offset Moves

(A G-Code Strategy)



Moving around the inside of an obtuse angle. Going from circle to line

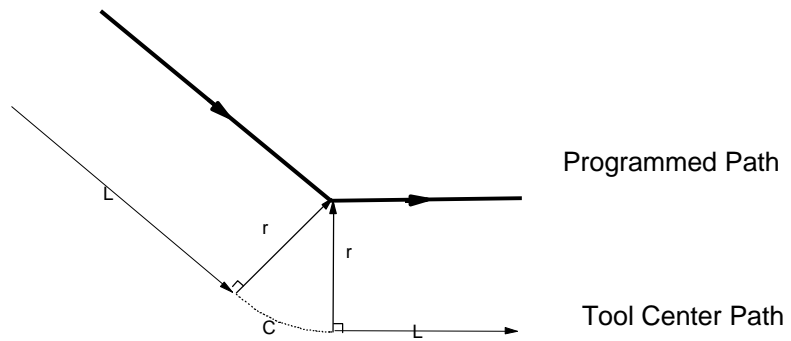


Moving around the inside of an acute angle. Going circle to circle

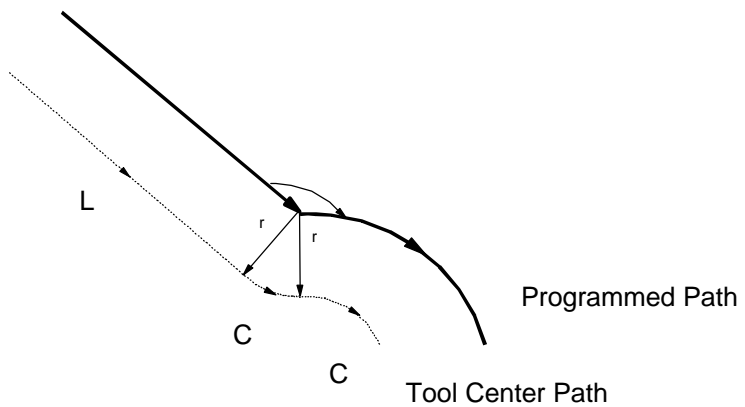
L: Linear  
C: Circle

## Continuous Offset Moves

(A G-Code Strategy)



Moving around the outside of an obtuse angle. Going from line to line

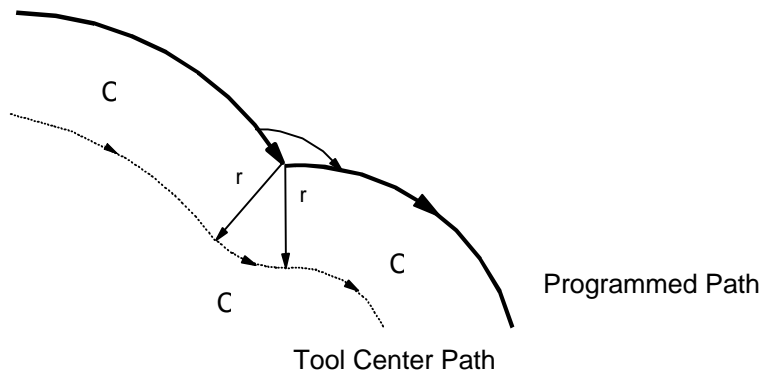


Moving around the outside of an obtuse angle. Going from line to circle

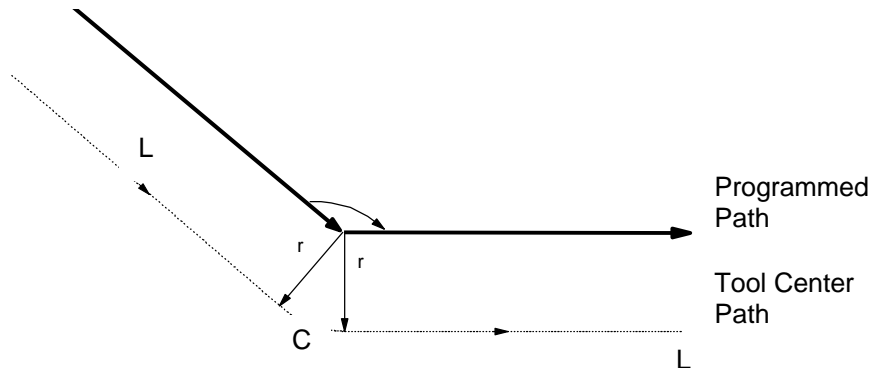
L: Linear  
C: Circle

## Continuous Offset Moves

(A G-Code Strategy)



Moving around the outside of an obtuse angle. Going from circle to circle

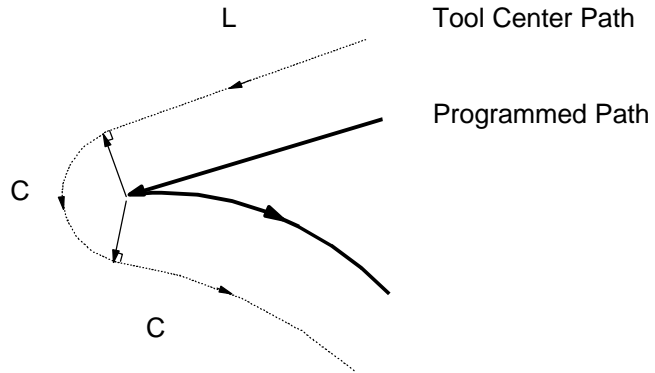


Moving around the outside of an obtuse angle. Going from line to line

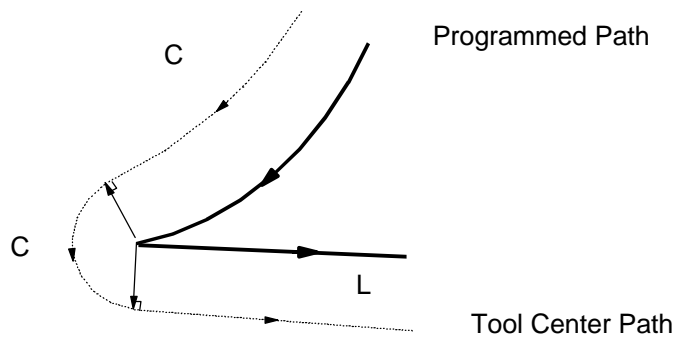
L: Linear  
C: Circle

## Continuous Offset Moves

(A G-Code Strategy)



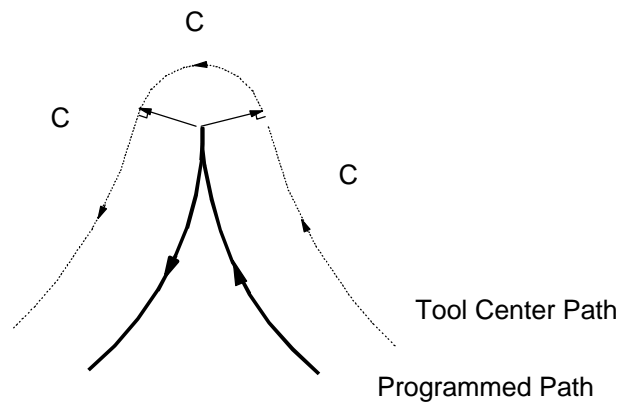
Moving around the outside of an acute angle. Going from line to circle



Moving around the outside of an acute angle. Going from circle to line

## Continuous Offset Moves

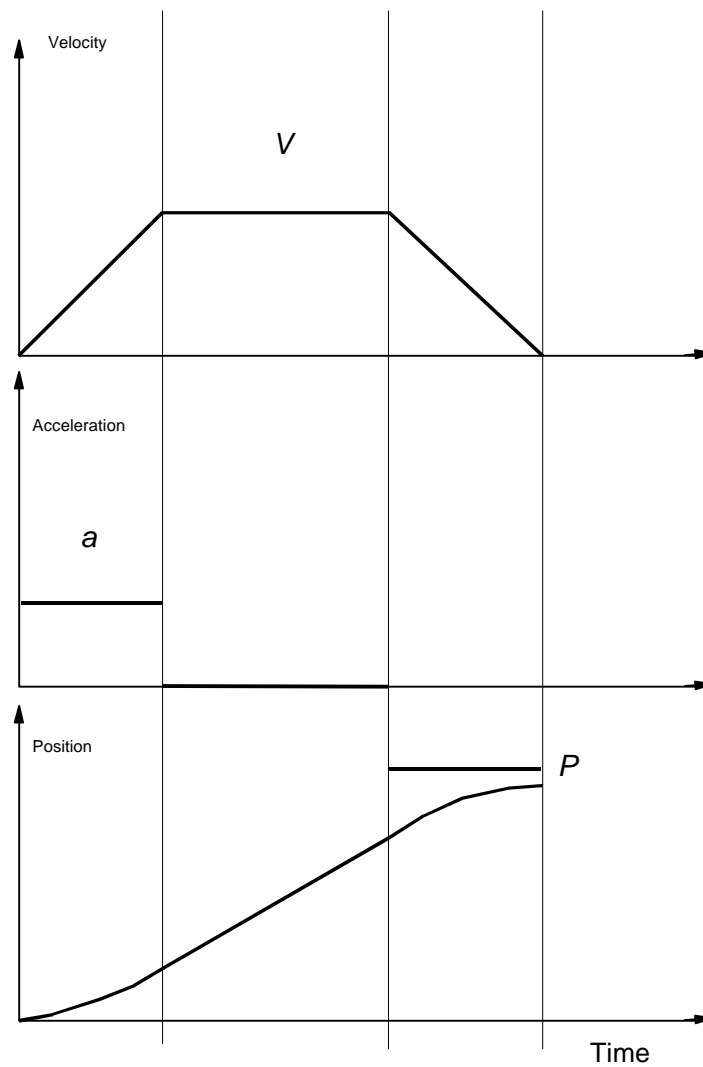
(A G-Code Strategy)



Moving around the outside of an acute angle. Going from circle to circle

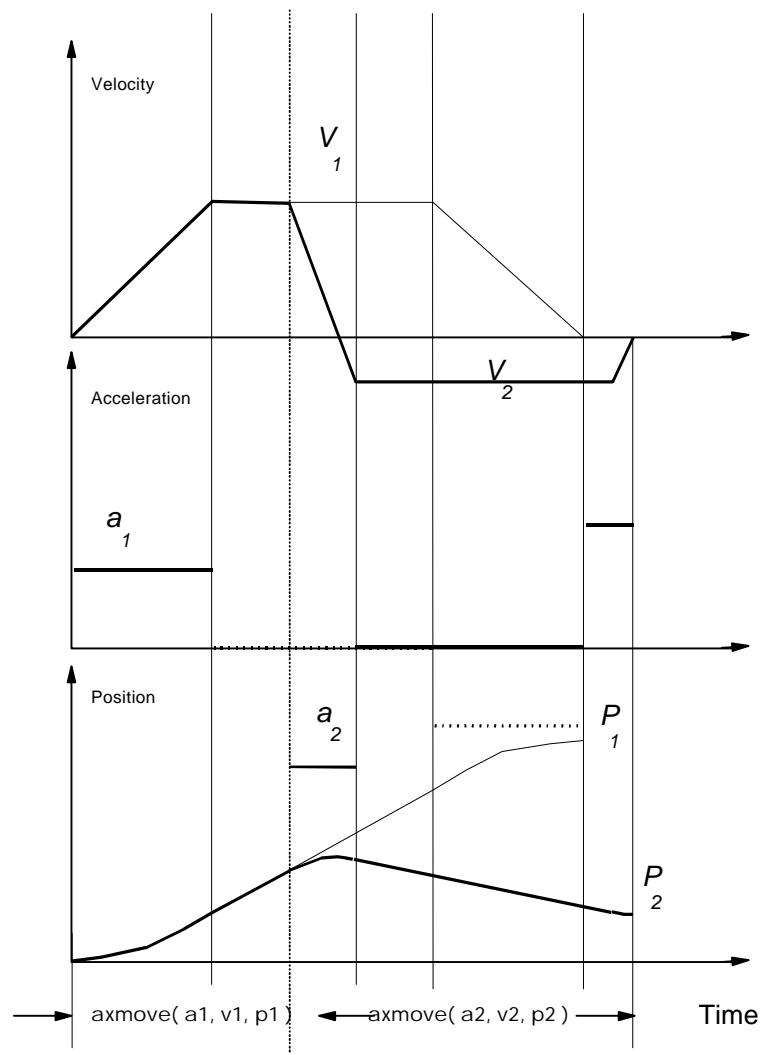
## Point-to-Point Trapezoidal Move

(A DSPL Command)



The move stops at any programmed position and may be intercepted by another trapezoidal move

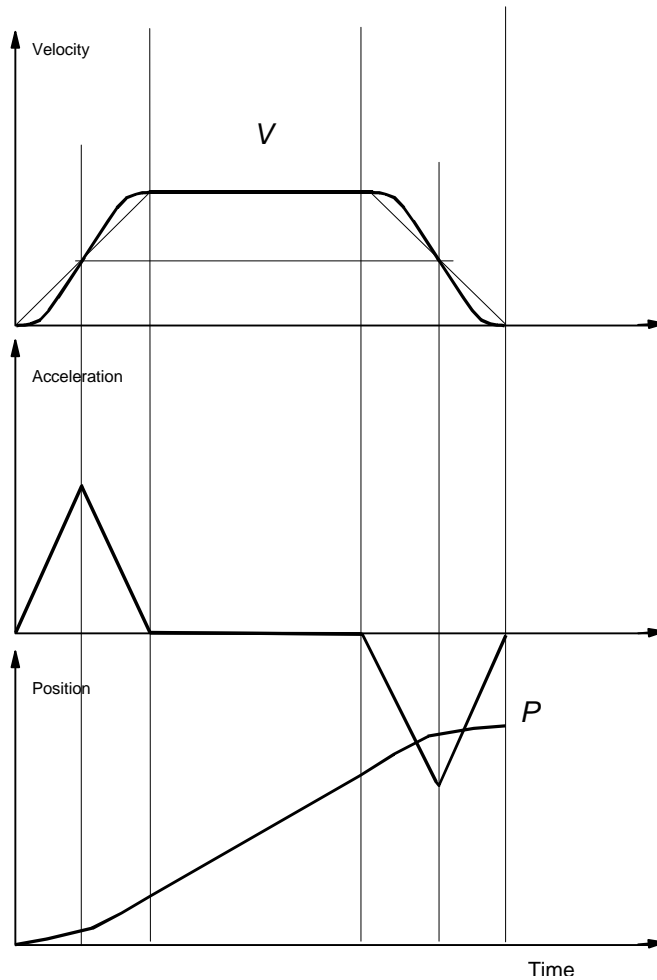
## Intercepting a Trapezoidal Move



The first move with programmed acceleration, end position and speed can be intercepted by another move with an entire different set of arguments at any time.

## Point-to-Point “S” Curve Move

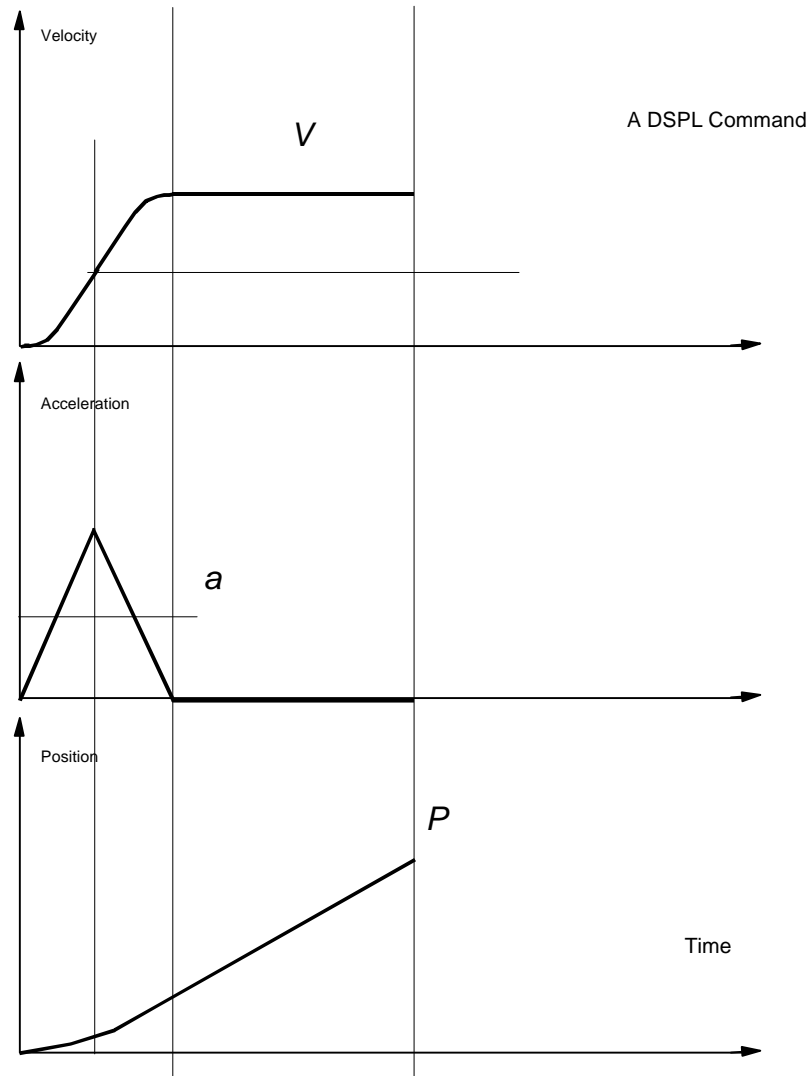
(A DSPL Command)



The arguments for this command are similar to those for a trapezoidal move. The Mx4 will automatically calculate jerk to perform the move in the same time as the trapezoidal move.

## Segment Moves With S-Curve

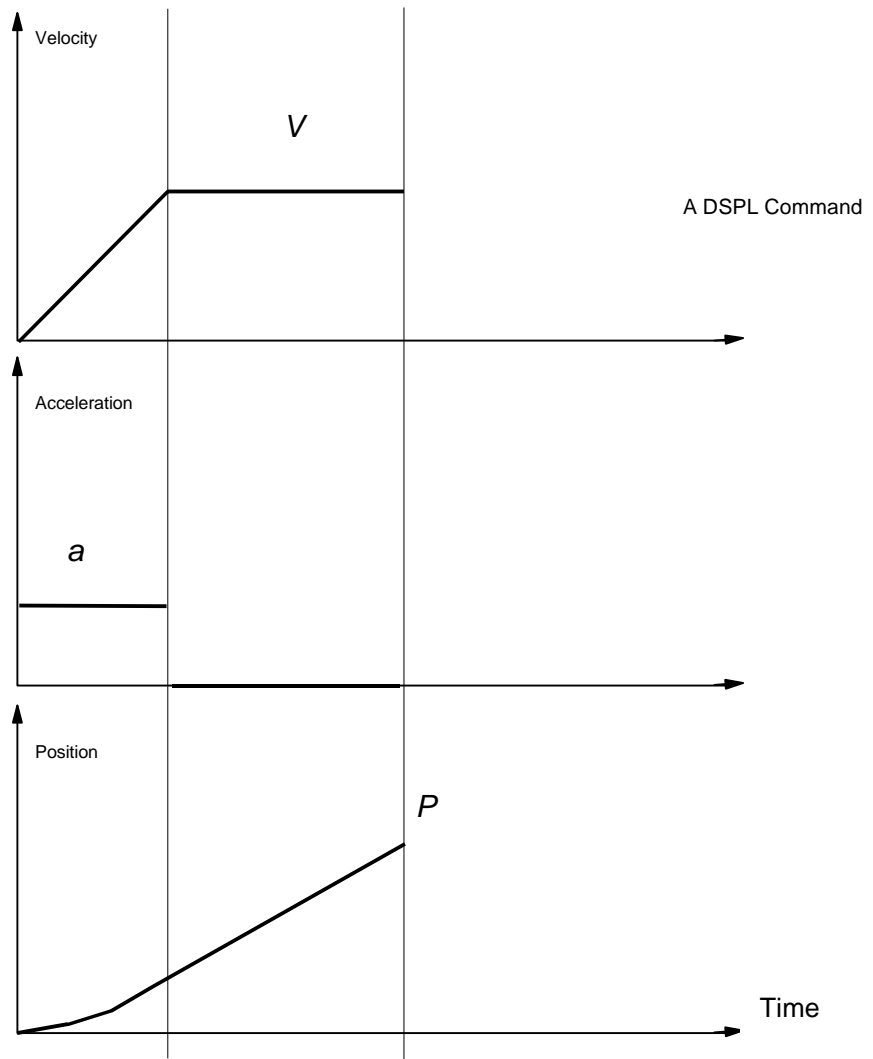
( A DSPL Command )



A segment is similar to the trapezoidal move, only its end position will be at the programmed slow velocity. Furthermore, DSPL "segment\_s" instruction uses the "s curve" profile to accelerate.

## Trapezoidal Segment Moves

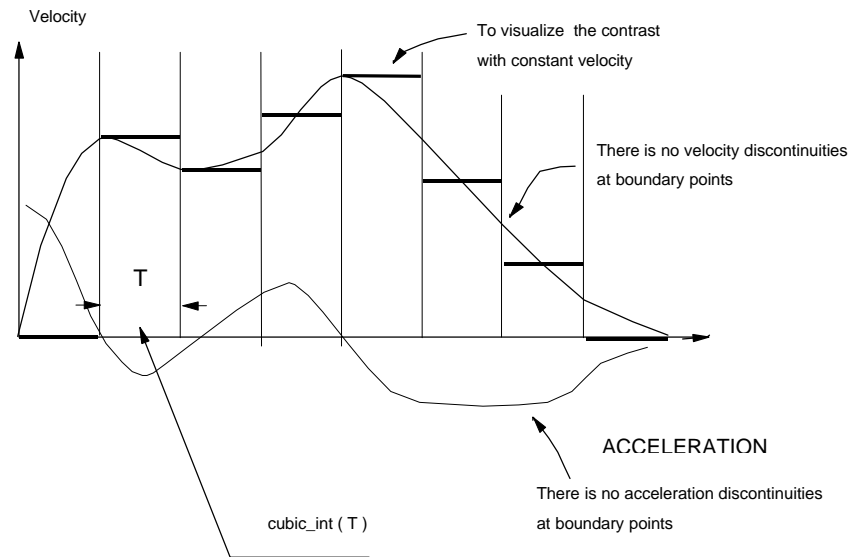
( A DSPL Command )



A segment is similar to the trapezoidal move, only its end position will be at its programmed slew velocity.

## Cubic Spline Moves

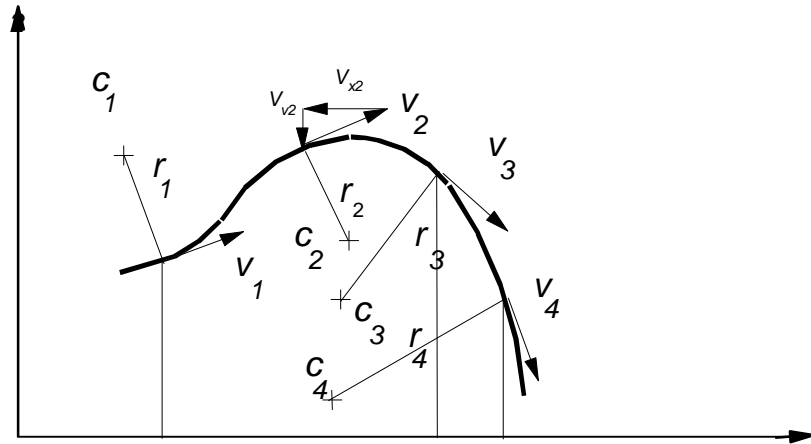
( A DSPL Command )



Motion control applications requiring fine moves from one point (a point with up to four components for positions and velocities) to another require cubic spline interpolation. Mx4 can run cubic splines either in contouring mode (the host continually updates Mx4's DPR with a new set of points), or in table mode (Mx4's table is pre-loaded with a set of points only once). In table mode the user array can be up to 2,000 points long. Each point specifies the position and velocity for only one motor.

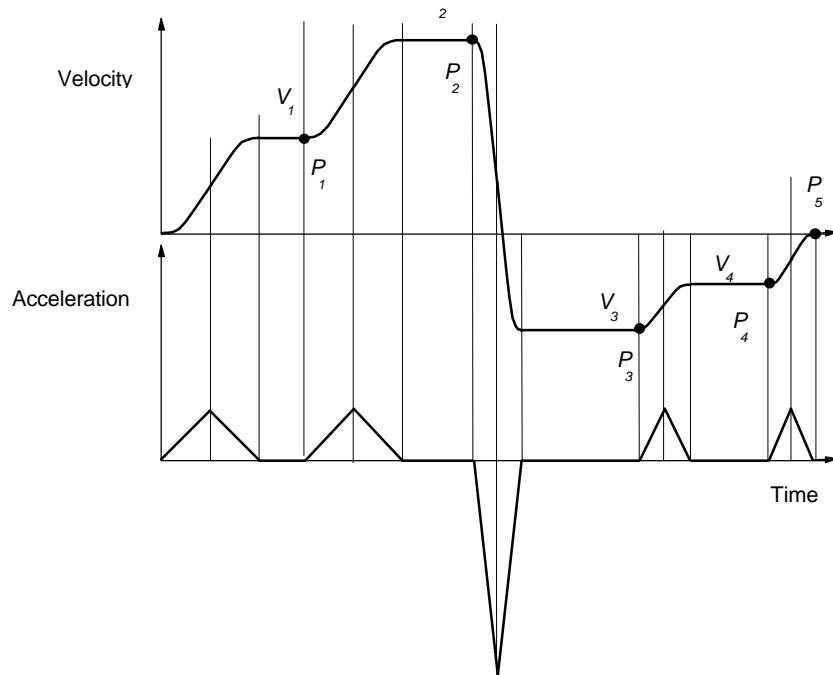
## PVT Hermite Spline

( A DSPL/RTC Command )



The Hermite Spline or PVT (position, velocity and time) allows the user to specify a set of position, velocity and corresponding time interval for each axis. The velocity for each axis is the resolved component of vector velocity (shown in the figure as  $V_{x2}$  and  $V_{y2}$ ). This vector is tangent to the radius of a local curvature.

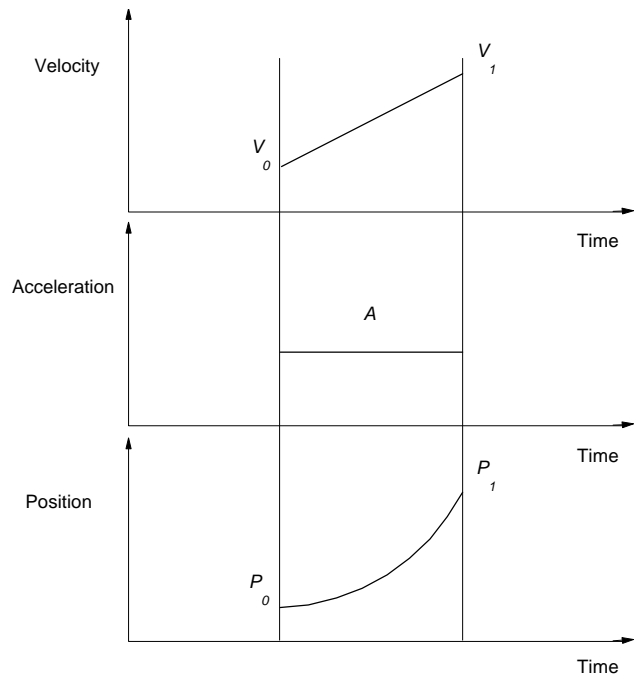
## Blending Moves



Either one of the DSPL Segment Moves, (segment or segment\_s) can be used in blending move applications. Blending allows the migration of one move to another at an exact programmed position without a need to stop the motion.

## Linear Move

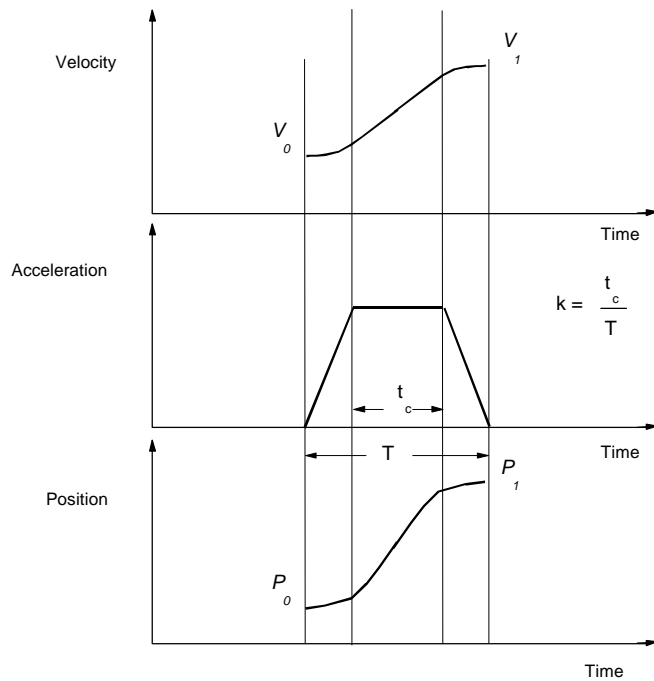
( A DSPL Command )



Constrained by a programmed maximum acceleration, the “linear\_move” command will linearly ramp the speed from its initial value to a new programmed value at a programmed position.

## "S" Curve Linear Move

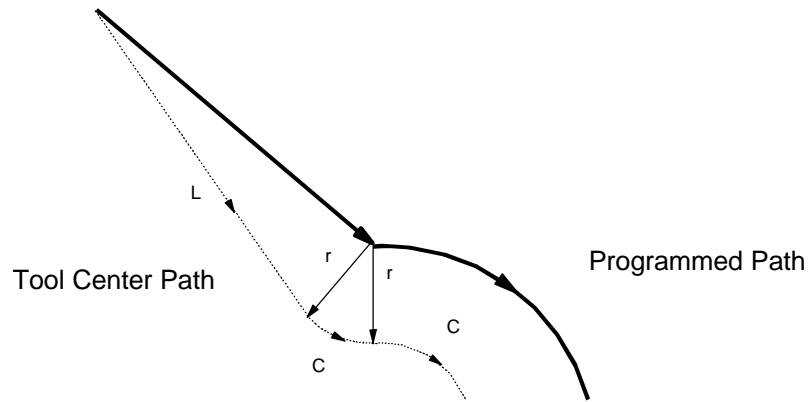
( A DSPL Command )



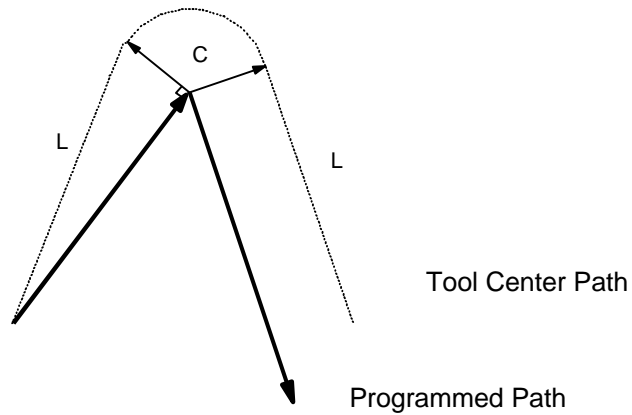
Constrained by a programmed maximum acceleration, the "linear\_move" command will linearly ramp the speed from its initial value to a new programmed value at a programmed position. The acceleration used with this command is "s" curve and its jerk is determined by: a, k and T.

## Start-up Offset Moves

( A G-Code Strategy )



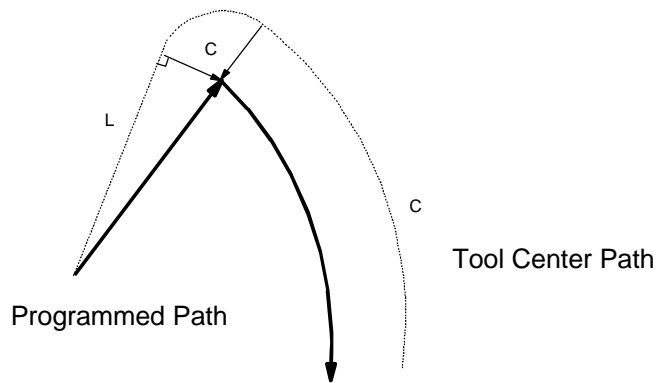
Starting around the outside of an obtuse angle. Going from line to circle



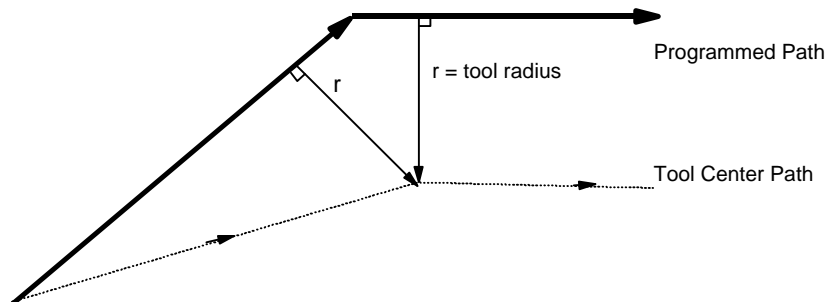
Starting around the outside of an acute angle. Going from line to line

## Start-up Offset Moves

( A G-Code Strategy )



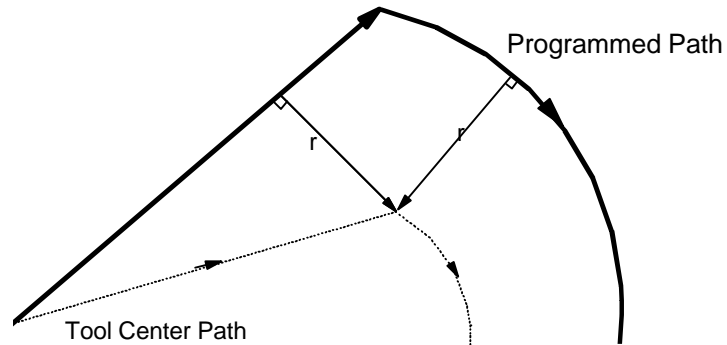
Starting around the outside of an acute angle. Going from line to circle



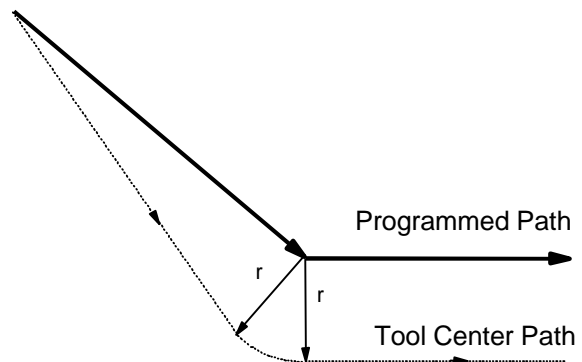
Starting around the inside of an obtuse angle. Going from line to line

## Start-up Offset Moves

( A G-Code Strategy )



Starting around the inside of an obtuse angle. Going from line to line



Starting around the inside of an obtuse angle. Going from line to line