

Six Key Concepts Needed to Master CNC Programming

Mike Lynch - CNC Concepts, Inc. - 847-639-8847 - lynch@cncci.com

Objective: *Learn what it takes to master G-code level CNC programming.*

Outline

- ❖ Key Concept 1: Know your machine from a programmer's viewpoint
 - Machine configurations: components, axes, and programmable functions
 - Visualizing program execution
 - Understanding program zero
 - Introduction to programming words
- ❖ Key Concept 2: You must prepare to create CNC programs
 - The importance of preparation
 - Preparation steps: mark-up print, create the process, select tooling and cutting conditions, do the math, plan the setup
- ❖ Key Concept 3: Understand the basic motion types
 - Motion commonalities
 - Motion types: rapid, linear, circular
- ❖ Key Concept 4: Compensation lets you deal with unpredictable tooling-related variables
 - Reasons for compensation, trial machining
 - Tool length compensation
 - Cutter radius compensation
 - Fixture offsets
- ❖ Key Concept 5: Provide structure to your CNC programs
 - Reason to provide structure: familiarization, consistency, rerunning tools
 - Four types of program format: Program start, tool end, tool start, program end
- ❖ Key Concept 6: You have special features to simplify programming
 - Canned cycles
 - Sub-programing
 - Other special features

Key Concept 1: Know the Machining Center from a Programmer's Viewpoint

You must come to know a CNC machining center from two distinctly different perspectives. In Key Concept 1, we look at the machine from a programmer's viewpoint.

Key Concept 1 is the longest of the Key Concepts. It contains several topics:

- Machine configurations
- Visualizing program execution
- Understanding the workpiece coordinate system
- Introduction to programming words

Machine Configurations of vertical machining centers

As a programmer, you must understand the characteristics of a CNC machining center. You must be able to identify its basic components—you must understand the moving components of the machine (called axes)—and you must know the various functions of your machine that are programmable.

Vertical Machining Centers

A vertical machining center has its spindle oriented vertically. The spindle, and therefore the cutting tool, point downwards toward the machine's table and the part. Because of this spindle/tool orientation, chips will tend to collect and build up on the workpiece, and may eventually interfere with machining operations. However, this is a very popular type of CNC machining center because it closely resembles the knee-mill—a popular type of conventional machine. For anyone with experience using a knee-mill, a vertical machining center should be quite familiar.

C-frame Style

A common type of vertical machining center is called a C-frame-style machining center because the headstock, column, and bed, when viewed from the left-hand side, form the letter "C". An automatic tool changer is mounted to the machine (usually on the left side) to allow tools to be loaded into the spindle without operator intervention. The table has a series of tee-slots and/or location/clamping holes to allow workholding devices (like a table vise) to be mounted on the table.

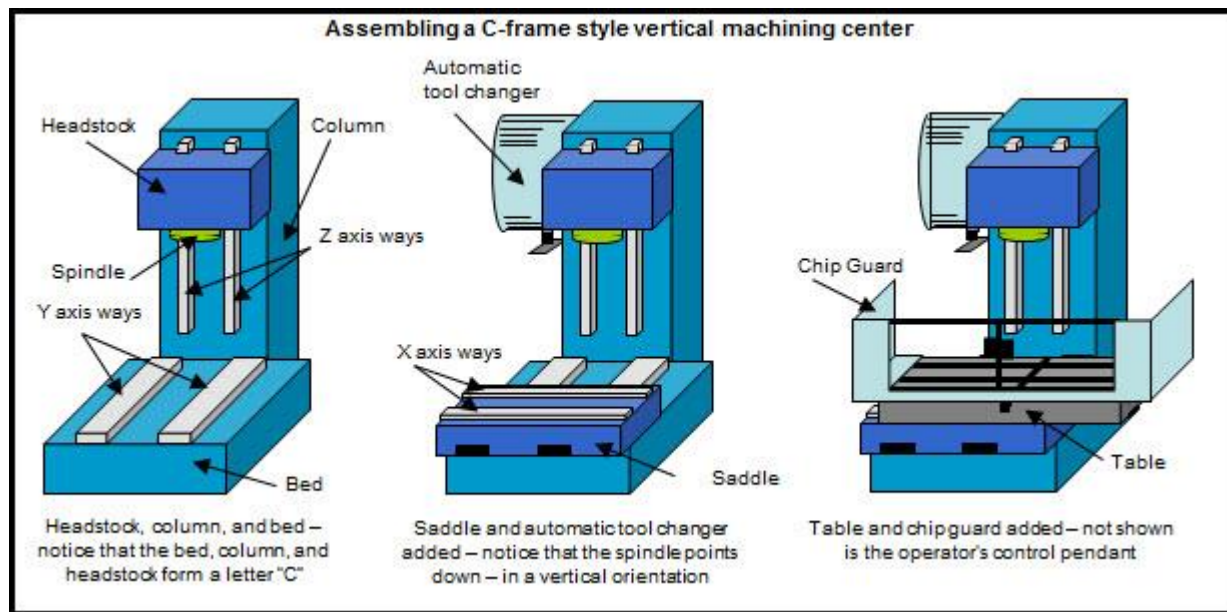


Figure 1.1: Primary components of a C-frame-style vertical machining center

Directions of motion (axes) for a C-frame-style vertical machining center

Basic vertical machining centers allow three directions of motion, or *axes*. These three basic motions are linear axes—allowing motion along a straight line. With a C-frame-style vertical machining center, the table can move left/right (the X-axis)—the table can move fore/aft (the Y-axis)—and the headstock or spindle can move up/down (the Z-axis). Figure 1.2 shows the axes of a C-frame-style vertical machining center.

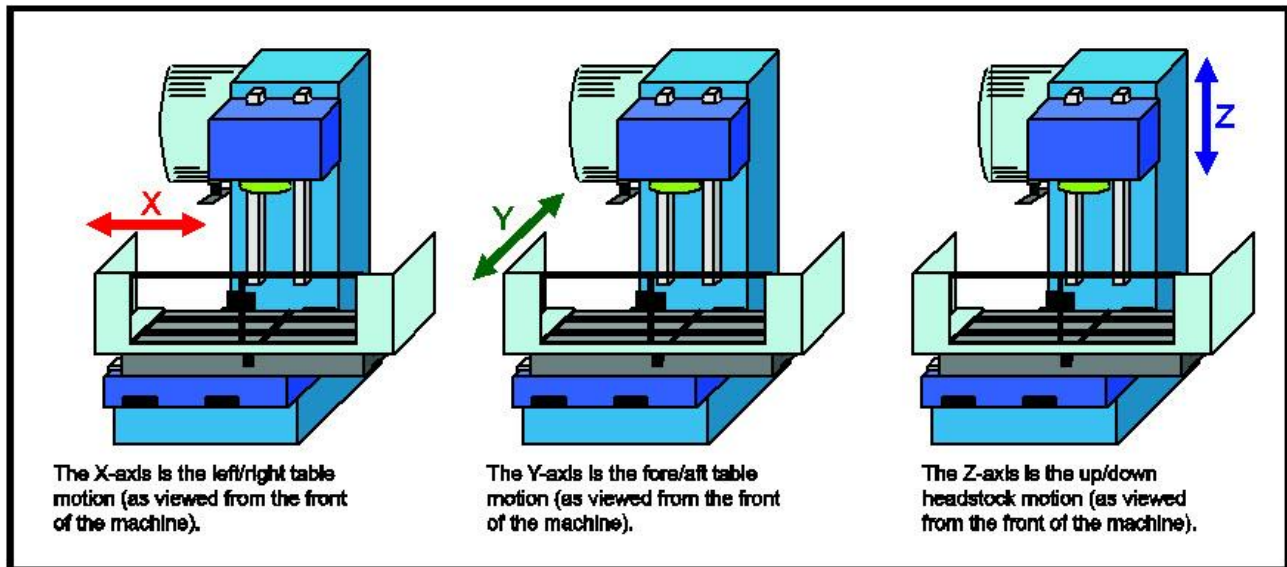


Figure 1.2: Directions of motion (axes) of a C-frame-style vertical machining center

With this kind of machine, notice that the cutting tool does not move in the X and Y-axis. The table and therefore the part moves in X and Y in relation to the tool. The tool only actually moves in the Z-axis.

Axis polarity

Though not depicted in figure 1.2, each axis has a polarity (plus and minus direction). As the table moves to the left, it is moving in the X-plus direction. As it moves to the right, it is moving in the X-minus direction. As the table moves toward you, it is moving in the Y-plus direction. As it moves away from you, it is moving in the Y-minus direction. As the headstock/cutting tool moves up, it is moving in the Z-plus direction. As it moves down, it is moving in the Z-minus direction.

Since the cutting tool does not move in the X and Y axes, it can be a little confusing (especially for programmers) to understand polarity by looking at table motion. From a programmer's viewpoint, it is much easier to understand polarity if you imagine that the cutting tool is moving in all axis. Figure 1.3 shows how to visualize polarity with this method.

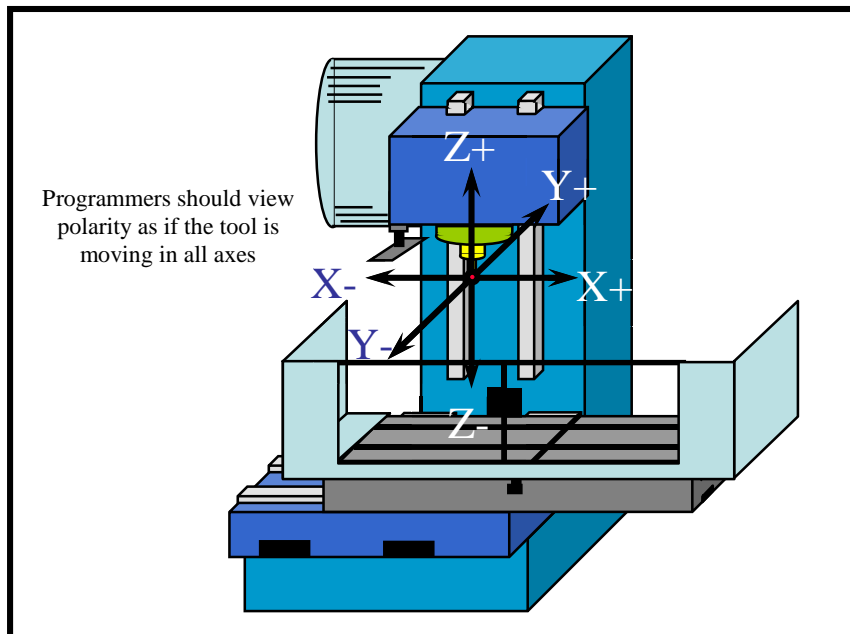
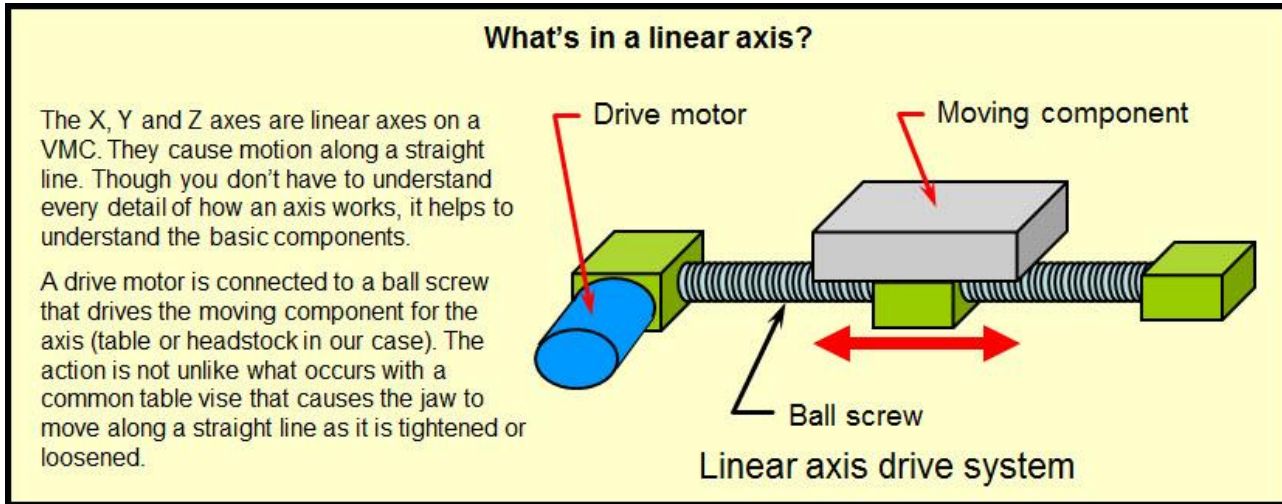


Figure 1.3: Viewing polarity as if the tool is moving in each axis

If you imagine that the cutting tool is moving in X and Y, determining polarity will be easier. As the cutting tool moves to the right, it is moving in the X-plus direction. (But remember, the cutting tool does not really move to the right in the X-axis—it is the relative motion of the tool and part as the table moves to the left—which again, is the X-plus direction.) As the cutting tool moves to the left, it is moving in the X-minus direction. As the tool moves away from you, it is moving in the Y-plus direction. As it

moves toward you, it is moving in the Y-minus direction. In Z, of course, the tool is really moving with the axis, so polarity is much easier to understand—up is Z-plus, down is Z-minus.



Programmable Functions of a Machining Center

A true CNC machining center will allow you to control just about all of its functions in a program. There should be very little or no operator intervention during a CNC machining cycle. Below are some common functions that can be programmed on most machining centers. While we do show the related CNC words used to command these functions, it is not our intention to teach programming commands. We are simply making you aware of the kinds of things a programmer can control in a program.

Spindle

The spindle of all machining centers can be programmed in at least three ways, activation (start/stop), direction (forward/reverse), and speed (in revolutions-per-minute or rpm). Some machining centers also provide multiple power or gear ranges (like the transmission of an automobile).

Spindle speed

You can precisely control how fast the spindle of a machining center rotates in one rpm increments. An S-word is used to specify spindle speed. If you want the spindle to rotate at 350 rpm, program S350. Since spindle speed is specified in whole numbers, you must not include a decimal point with the S-word. Also, the S-word by itself does not actually start or activate the spindle.

Spindle activation and direction

You can control which direction the spindle rotates—forward or reverse—and stop the spindle using M-codes. The forward direction is used for right-hand tooling. It will appear as counter-clockwise when viewed from in front of (below) the spindle. The reverse direction is used for left-hand tooling and will appear as clockwise when viewed from in front of the spindle.

Three M-codes control spindle activation. M03 turns the spindle on in the forward direction. M04 turns the spindle on in a reverse direction. M05 turns the spindle off.

What is an M code?

<p>M codes control many of the programmable functions on a machining center. In many cases, you can think of them as being like programmable on/off switches. "M" stands for "miscellaneous" or "machine" function.</p> <p>M codes are created by the machine tool builder, and will often vary from one machining center to another. Here we show some common M codes, but you must look in your machine tool builder's programming manual to find the complete list of M codes for a given machining center.</p>	<p>These M codes don't vary:</p> <p>M00: Program stop M01: Optional stop M03: Spindle on (forward) M04: Spindle on (reverse) M05: Spindle off M06: Activate tool changer M08: Flood coolant on M09: Coolant off M30: End of program M60: Activate pallet changer</p>	<p>These M codes vary:</p> <p>M ___ : Automatic door open M ___ : Automatic door close M ___ : Chip conveyor on M ___ : Chip conveyor off M ___ : _____ M ___ : _____ M ___ : _____ M ___ : _____ M ___ : _____ M ___ : _____</p>
--	--	---

Spindle range

Some, especially larger machining centers, have two or more spindle ranges. Spindle ranges are like the gears in an automobile transmission. Lower ranges are used for power—higher ranges are used for speed. With most modern machining centers, spindle

range selection is automatic and transparent. The spindle range will be automatically selected when you specify a spindle speed (S-word). For this reason, some programmers don't even know the machine that they are programming has two or more spindle ranges! For older machines, you may have to change gears in the program using M-codes—check the machine tool builder's programming manual.

Feedrate

As you know, a machining center has three linear axes, X, Y, and Z. You must be able to control how quickly these axes move, especially during machining. Feedrate is the rate at which the cutting tool will move during a machining operation. It is a programmable function for all machining centers. Feedrate is specified with an F-word (F for feedrate). For most machining centers, feedrate is specified in the distance moved in a minute, either inches-per-minute or millimeters-per-minute.

Most cutting tool manufacturers provide feedrate recommendations in distance-per-revolution or distance-per-tooth/flute. That is in inches-per-rev (ipr) or millimeters-per-rev (mmpr). To determine the inches-per-minute (ipm) feedrate, you must multiply the inches-per-revolution (ipr) value by the spindle speed used in the program in rpm.

Modern CNCs allow feedrate to be specified directly in distance-per-revolution (inches-per-revolution or millimeters-per-revolution) - which minimizes calculations. If both feedrate specifications are allowed (inches-per-minute and inches-per-revolution), G-codes are used to specify the feedrate type.

G94 is used to select feed-per-minute mode and G95 is used to select feed-per-revolution mode. Any F-word following a G94 will be considered as a feed-per-minute feedrate. Any F-word following a G95 will be considered as a feed-per-revolution feedrate.

To make programs as flexible as possible, it is recommended that G94 feed-per-minute be used for all normal machining; after all it is a simple calculation. The one application that will benefit from using feed-per-revolution programming is tapping, where the feedrate must be equal to the threads-per-revolution.

What is a G code?		
<p>G codes are called <i>preparatory functions</i>. They prepare the machine for what is coming up – in the current command and possibly in up-coming commands. In many cases, they set <i>modes</i>, meaning once a G code is <i>instated</i> it will remain in effect until the mode is changed or cancelled.</p> <p>Here we list a few common G codes, but don't worry if they don't make much sense yet. Upcoming discussions will clarify.</p>	<p>Common G codes:</p> <p>G00: Rapid motion G01: Straight line motion G02: Circular motion (CW) G03: Circular motion (CCW) G04: Dwell G20: Inch mode selection G21: Metric mode selection G28: Zero return command G40: Cancel cutter radius comp. G41: Cutter comp. left</p>	<p>G42: Cutter comp. right G43: Tool length comp. G81: Drilling cycle G83: Peck drilling cycle G84: Tapping cycle G90: Absolute mode G91: Incremental mode G98: Initial plane G99: R plane</p>
<p>Look in your control manufacturer's manual for a full list of G codes.</p>		

Coolant

Coolant is the fluid used to flush chips away from the cutting area. It also cools and lubricates the cutting operation. All machining centers provide flood coolant capability. Flood coolant is turned on and off with an M-code—M08 coolant on; M09 coolant off.

Automatic Tool Changer

Automatic tool changer designs vary from machine to machine, but programming methods remain similar, falling into one of two basic programming styles. We'll discuss the specific differences during Key Concept 5.

A T-word is used to select the next tool. This may rotate an automatic tool changer mechanism. With most machines, the T-word does not actually cause a tool change to occur. Again, it just rotates the magazine to its ready position (also called the waiting position) or specifies the next tool to be used.

ATC magazine stations are numbered. A machine having a tool changer magazine that can hold twenty tools will commonly have T-words ranging from T01 to T20. The T-word is used to specify the desired tool station number that will be brought to the ready position.

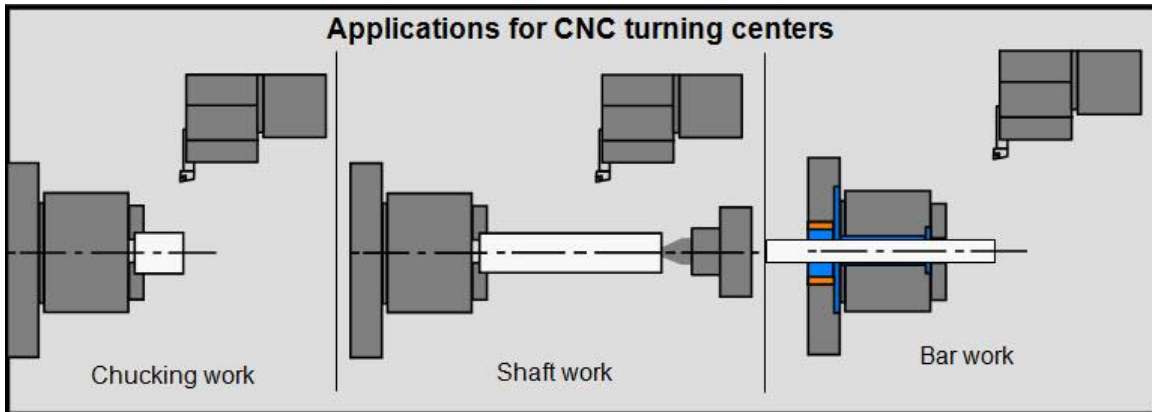
An M06 M-code is used to actually make the tool change. For most machining centers, M06 will cause the tool that is in the spindle to be placed back into the magazine—and the tool in the ready position will then be placed into the spindle. For example, the command

T05 M06

will cause tool station number 5 to rotate to the ready station, then the tool in station 5 is placed into the spindle, and whatever tool was previously in the spindle will be placed back into the magazine at the correct position.

Machine configuration for universal style slant bed turning centers

This style of turning center is called a *universal style* turning center because it can perform all three forms of turning applications – chucking work, shaft work, and bar work. This explains why it is the most popular type of turning center – it provides the most flexibility to CNC turning center users.



When raw material comes to the machine in the form of short slugs (like round bars cut to length), the application is called chucking (or chucker) work. The raw material is secured solely by the workholding device (commonly a three-jaw chuck).

With longer slugs (longer than about three to four times the raw material diameter), the workholding device by itself will not be sufficient to secure the workpiece for machining. For these applications, some form of work support device/s must be used (commonly a tailstock and/or steady-rest). This application is called shaft work.

With bar work, the raw material comes to the turning center in the form of a long bar (from four to fifteen feet long [1.2-5 meters], depending upon the type of bar feeder being used). Bar work requires a special bar support and feeding device (called a bar feeder). The bar is fed through the headstock and spindle into the working area. A workpiece is machined and cut off from the bar. The bar is then fed again for another workpiece to be machined.

Figure 1.1 shows a universal style slant bed turning center. The headstock houses a spindle to which the workholding device is mounted. Our illustration shows a three-jaw chuck, but other types of workholding devices can be used (collet chuck, expanding mandrel, etc.). To the right of the workholding device is the tailstock, which is used to support the right end of long workpieces – again, for shaft work. The turret of the turning center is used to hold cutting tools and it can be quickly rotated from one tool station to another. Current turning centers have turrets that hold from six to twelve cutting tools.

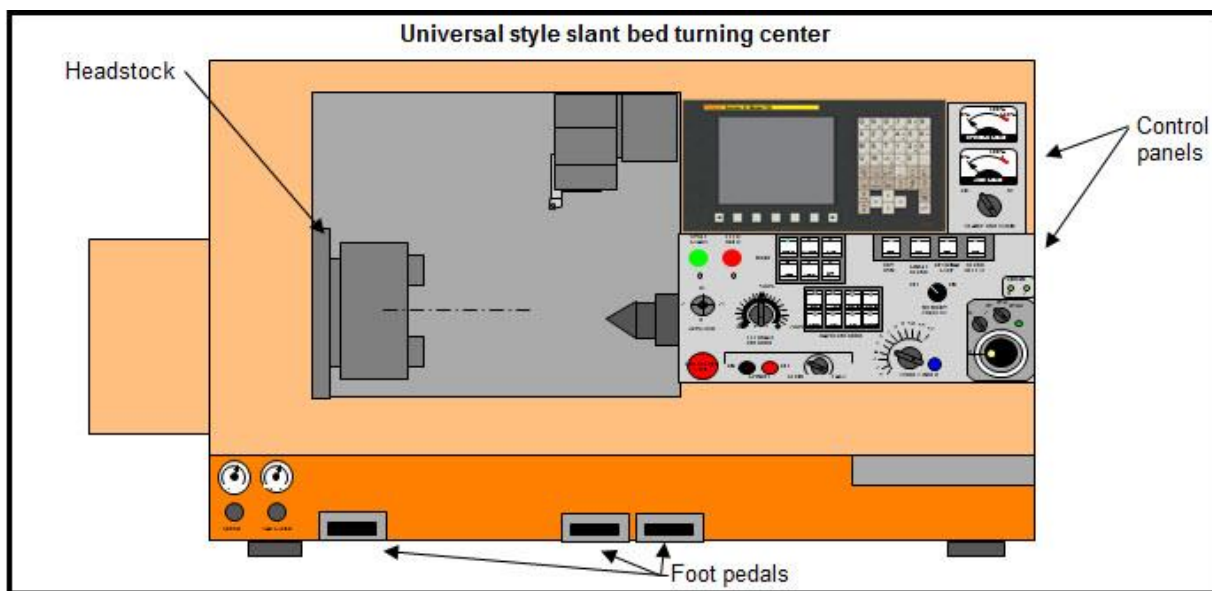


Figure 1.1 – A universal style slant bed turning center with its door removed

Directions of motion (axes) for a universal style slant bed turning center

All turning centers have at least two linear *axes* of motion. The turret (and cutting tool) will move along with these two axes. By *linear*, we mean the axis moves along a straight line.

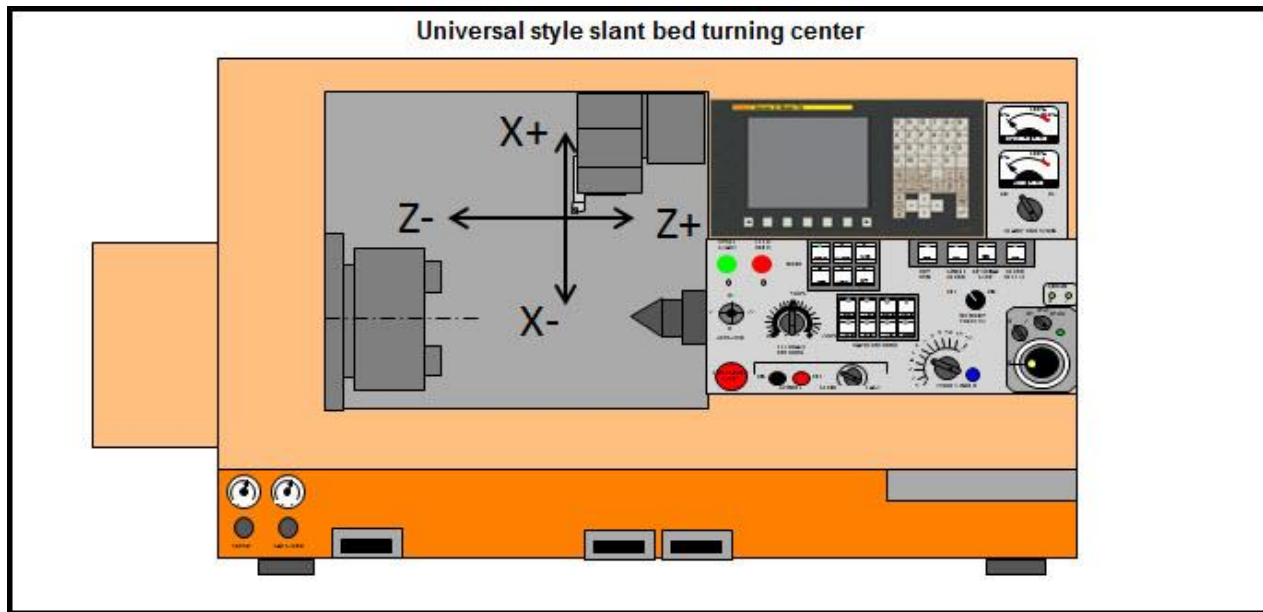


Figure 1.2 – The two most basic axes of motion for a CNC turning center

The *diameter-controlling* axis (up/down motion of the turret as shown in Figure 1.2) is the *X-axis*. The *length-controlling* axis (right/left motion of the turret as shown in Figure 1.2) is the *Z-axis*. Figure 1.2 shows these directions of motion along with the polarity (+/-) for each.

These two most basic directions of motion will remain exactly the same for almost all types of turning centers (only a handful of turning center manufacturers stray from what we show in Figure 1.2.) *The X-axis will always be the diameter-controlling axis* – and X minus is always the direction that causes the cutting tool to move to a smaller diameter (toward the spindle centerline). *The Z-axis will always be the length controlling axis* – and Z minus will always be the direction that causes the cutting tool to move toward the workholding device.

X is specified in diameter

Though we may be a little ahead of ourselves, the X-axis is designated in *diameter* for almost all turning centers. That is, if a diameter of 3.0 inches must be machined, the designation for the X-axis will be X3.0. There are some (especially older) turning centers that require the X-axis to be specified with radial values. For these machines, the word X1.5 will cause the tool to be positioned to a 3.0 inch diameter. Note that it is much easier to work with a turning center when the X-axis is designated in diameter – which is why most current model turning centers do so.

Programmable functions of turning centers

A true CNC turning center will allow you to control just about any of its functions from within a program. There should be very little operator intervention during a CNC cycle. Here we list some common functions that can be programmed on all *true* turning centers. While we do show the related CNC words used to command these functions, our intention here is not to teach programming commands (yet). It is to simply make you aware of the kinds of things a programmer can control through a program.

Spindle

The spindle of all turning centers can be programmed in at least three ways, activation (start/stop), direction (forward/reverse), and speed (in either surface feet/meters per minute or revolutions per minute). Many turning centers additionally provide multiple power ranges (like the transmission of an automobile).

Spindle speed

You can precisely control how fast the spindle of a turning center rotates. An *S-word* is used for this purpose. There are two ways to specify spindle speed. When the spindle is in rpm mode, an S-word of S500 specifies a speed of 500 revolutions per minute (rpm). When the spindle is in *constant surface speed mode*, an S-word of S500 specifies a speed of 500 surface feet per minute (sfm), assuming you are working in the inch measurement system. (If you work in the Metric measurement system, S500 will specify 500 meters per minute when in constant surface speed mode.)

We will describe the two spindle speed modes – as well as how to determine how and when to use them.

Spindle activation and direction

You can also control which direction the spindle rotates – *forward or reverse*. The forward direction is used for right hand tooling (when machining occurs toward the workholding device). It will appear as counter-clockwise when viewed from in front of the machine. The reverse direction is used for left hand tooling and will appear as clockwise when viewed from in front of the spindle.

Three *M-codes* control spindle activation. M03 turns the spindle on in the forward direction (used with right-hand tools). M04 turns the spindle on in a reverse direction (for left-hand tools). M05 turns the spindle off.

Spindle range

Many, especially larger turning centers, have two or more spindle ranges. Spindle ranges are like the gears in an automobile transmission. Generally speaking, lower ranges are used for power – higher ranges are used for speed. With most turning centers, spindle range selection is done with M-codes. While the specific M-code numbers for spindle range selection will vary from one machine tool builder to another, many turning center use M41 to select the low range and M42 to select the high range. We'll use these two M-codes (M41: low and M42: high) to specify spindle range selection throughout this text.

Feedrate

You know that all turning centers have at least two linear axes, X and Z. You also know that the cutting tool (for most turning centers) moves along with these two axes. It is the motion of the cutting tool while it is in contact with the workpiece that causes machining to occur. It is important that the motion *rate* (how quickly the tool moves) be appropriate to the machining operation being performed. In CNC turning center terms, this motion rate is called *feedrate*.

An F-word is used to specify feedrate. And like spindle speed, feedrate can be specified in two ways. It can be specified in *per minute* fashion or in *per revolution* fashion. As the names imply, when feedrate is specified in per minute fashion, it specifies how far the cutting tool will move during one minute. When feedrate is specified in per revolution fashion, it specifies how far the cutting tool will move during one spindle revolution.

Also as with spindle speed (at least in constant surface speed mode), feedrate specification is related to the measurement system you use. In the inch mode, feedrate is specified in either inches per minute (ipm) or inches per revolution (ipr). In metric mode, feedrate is specified in either millimeters per minute (mmpm) or millimeters per revolution (mmpr).

- F-word – Feedrate specification
- G20 – Inch mode
- G21 – Metric mode
- G98 – Feed per minute mode
- G99 – Feed per revolution mode

Here are a few examples of feedrate specification:

```
N010 G20 G98 F4.0 (4.0 inches per minute)
N020 G20 G99 F0.015 (0.015 inches per revolution)
N030 G21 G98 F100.0 (100.0 millimeters per minute)
N040 G21 G99 F0.5 (0.5 millimeters per revolution)
```

Turret index and offset selection

A T-word specifies which cutting tool will be used. For turning centers that have a turret, the T-word will actually cause the turret to index to the specified turret station. But there's a little more to the T-word than turret index.

For most machines, the T-word is a four-digit word. The first two digits specify the turret station and *geometry offset* to be used with the tool (geometry offsets assign *program zero*). The second two digits of the T-word specify the *wear offset* to be used with the tool (wear offsets allow the operator to make minor adjustments).

The command

```
N020 T0404
```

will cause these three things to occur:

- the turret to index to station number four (first two digits)
- geometry offset number four will be selected (first two digits)
- wear offset number four will be selected (second two digits)

Almost all current model turning centers have *bi-directional turrets*. That is, the turret can rotate in either direction. When a T-word is given, most machines will cause the turret to automatically rotate in a direction that provides the shortest rotational distance to the specified tool

With gang style turning centers, of course, there is no turret to index. Only two things will happen with the previous command: geometry and wear offset number four will be selected.

Most programmers will make the wear offset number the same number as the turret station number and geometry offset number.

Coolant

All turning centers allow programmable control of *flood coolant*. Coolant is commonly used to cool the workpiece during machining and to lubricate the machining operation. Two M-codes are used to control coolant. Almost all turning center manufacturers use M08 to turn on flood coolant and M09 to turn it off.

Visualizing program execution

A CNC programmer must possess the ability to visualize the movements a CNC machine will make as it executes a program. The better a person can visualize what the machining center must do in order to machine a part, the easier it will be to create a workable CNC program.

In order to be able to write a part program for a CNC machining center, you must be able to visualize (see in your mind) the movements of the machine axes required to machine the part geometry. You must also be able to visualize the activation of the various machine functions required including spindle start and stop, tool selection, and coolant flow. Experience machining parts with a conventional milling machine may be useful when visualizing a CNC machining center executing a part program.

When a machinist prepares to machine a workpiece on a conventional milling machine, they have the advantage that everything they need for the job is right in front of them. The machine, cutting tools, workholding setup, and workpiece engineering drawing are all at hand to be used or referenced. Because of this, it is unlikely that the machinist will make a basic mistake like forgetting to start the spindle before trying to machine the workpiece.


When a programmer writes a program, they only have the workpiece engineering drawing to reference. The machine, tooling, workholding setup, and material blank are not physically available. For this reason, a programmer must be able to visualize what will happen during the execution of the program. A beginner programmer will be prone to forget certain things—sometimes very basic things like starting the spindle prior to machining the workpiece.

Visualization is necessary to develop any kind of instructions

Consider the visualization it takes to write a set of travel instructions to get someone from the airport to your company.

Before you can write the instructions, you must visualize the path from the airport to your company. If you cannot visualize the path, you cannot write the instructions. Worse, if you think you can visualize the path but you are wrong, you will write incorrect instructions and the person following them will get lost.

In similar fashion, if you cannot visualize the path of a cutting tool and the auxiliary operations, such as selecting the tool and starting the spindle, you cannot write an effective program for the CNC that will drive the cutter to make the right part.



- 1) Take airport exit to Highland Dr. Turn left.
- 2) Take Highland Dr. 4 mi to Elm St. Turn right.
- 3) Take Elm St. 1 mi To March Ave. Turn left.
- 4) Take March Ave 2 mi to Lance Dr. Turn right.
- 5) Take Lance Dr. 1/2 mi to company (on right).

Program Structure

Like the sentences that make up a set of instructions, a CNC program is made up of blocks. Each block is made up of words. Each word is made up of a letter address (N, X, Z, T, etc.) and a numerical value. The figure below shows the beginning of a CNC program that illustrates this basic program structure.

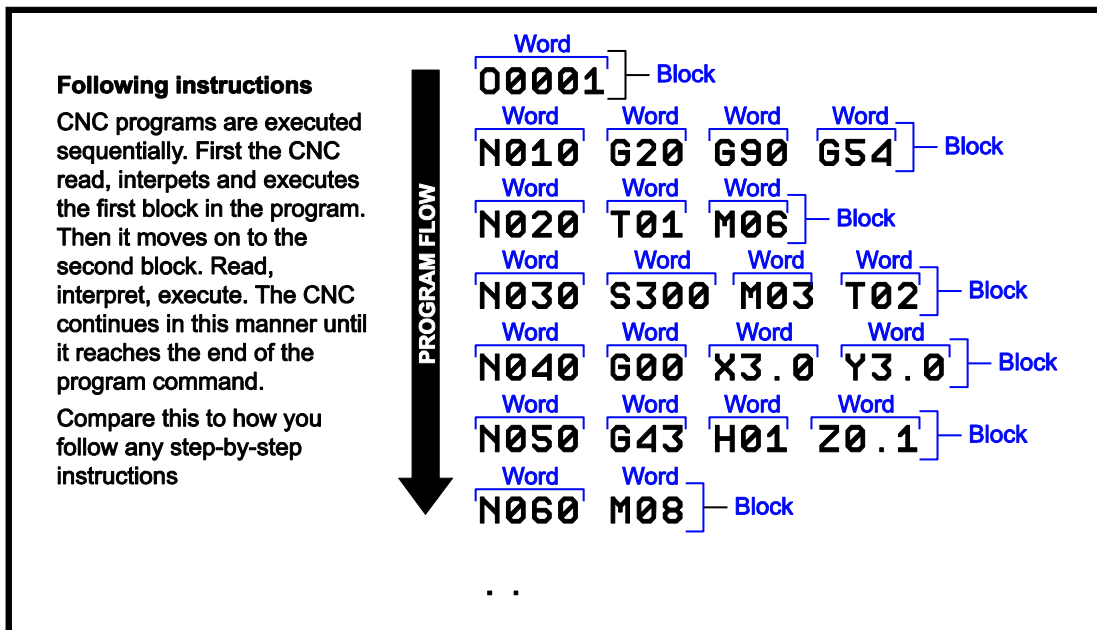


Figure 1.12: The flow of a CNC program

Order of Program Execution

You can compare writing a CNC program to writing a set of step-by-step instructions. For example, say you have just purchased a bookcase that requires assembly. The instructions you receive will be in sequential order. You perform step number one before proceeding to step number two. Each step will include an explanation of what it is you are supposed to do to complete the step. As you complete the procedure in each step, you are one step closer to finishing the complete assembly.

A CNC program is also executed in sequential order. The CNC will read, interpret, and execute the first block in the program. It will then go on to the next block. Read, interpret, execute. The CNC continues reading, interpreting and executing blocks until the end of the program command is reached.

The CNC will make no assumptions and executes each command in the program explicitly. Compare this to the set of instructions for assembling the bookcase. The manufacturer may assume that you have a screwdriver and that you know how to use it. Poorly written assembly instructions may be rather vague in describing exactly what you are supposed to do in a given step so that different people may interpret the instructions slightly differently. In contrast, each CNC command will have only one resultant machine action or set of actions.

An Example of Program Execution

To stress the sequential order of execution, and the visualization that is necessary to write programs, let's look at a very simple machining center example. We will first show the steps a machinist will perform to machine a very simple workpiece on a conventional milling machine. Then we will show the equivalent CNC program that will perform the same machining operation on a CNC machining center. In each case, we assume that the workpiece is already mounted in a vise on the table.

The figure below shows the print for this machining operation. In this case, we are simply drilling a 0.500 inch diameter hole to 1.0 deep.

This is a very simple example to illustrate the sequential order by which a machinist will machine a workpiece and to visualize the steps necessary to write a CNC program to perform the same machining operation.

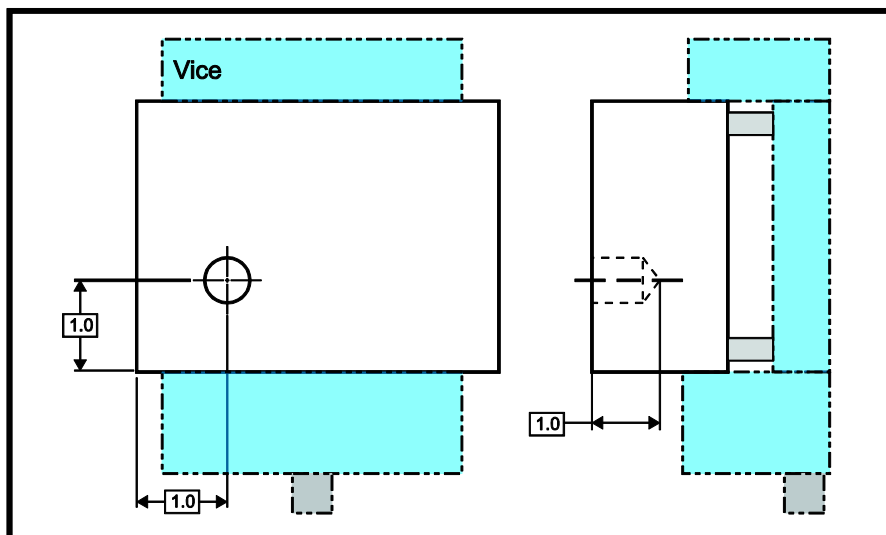


Figure 1.13: Drawing for example illustrating program execution flow

Manual milling machine procedure:

First let's look at the steps a machinist will take on a conventional milling machine or drill press.

1. Mount the tool (drill) in the spindle.
2. Turn spindle on CW and set the spindle speed to 600 RPM.
3. Move the table slides to position the workpiece under the drill.
4. Advance the headstock quill to move the drill close to the surface of the part.
5. Spray coolant on the tip of the tool
6. Advance the quill at the desired feedrate to drill the 0.500 hole, adding more coolant as required.
7. Retract the drill from the hole.
8. Move the tool away and turn off the spindle.

CNC program:

Now here is a CNC program to drill the 0.500 diameter hole in the workpiece on a CNC machining center.

```

O0001 (Program number)
N010 G20 G90 (Select inch & absolute programming modes)
N020 G54 (Set the program zero point)
N030 T01 M06 (Load the drill into spindle)
N040 S600 M03 (turn spindle on CW at 600 RPM)
N050 G00 X1.0 Y1.0 (Move tool above the hole in X and Y)
N060 G43 H01 Z0.1 M08 (Rapid to workpiece surface, instate tool length compensation, turn coolant on)
N070 G01 Z-1.0 F3.5 (Drill hole at 3.5IPM)
N080 G00 Z0.1 M09 (Retract drill, coolant off)
N090 G91 G28 Z0. M05 (Rapid to Z-ref position, spindle off)
N100 G28 X0. Y0. (Rapid to X/Y reference positions)
N110 M30 (End of program)

```

Understanding the rectangular coordinate system for machining centers

A programmer must be able to specify positions to which cutting tools will move as they machine a workpiece. The easiest way to do this is to specify each position relative to a common origin point called the workpiece coordinate system zero or program zero.

You know that machining centers have three linear axes—X, Y, and Z. You also know these axes move and that they have a polarity (plus and minus directions). In order to machine a workpiece in the desired manner, each axis must be moved in a

controlled manner. One of the ways you must be able to control each axis is with precise positioning within the workpiece coordinate system.

The workpiece coordinate system has an origin point that is called the workpiece coordinate system zero. It allows you to specify all positions or coordinates from this central location. As a programmer, you will be choosing the location for the workpiece coordinate system zero—and if you choose it wisely, many of the coordinates you will use in the program will come directly from your workpiece engineering drawing, meaning the number and difficulty of the calculations required to create a program can be reduced.

A Graph Analogy

A simple graph helps understanding of the CNCs workpiece coordinate system. Since everyone has had to interpret a graph at one time or another, we can relate what you already know to CNC coordinates. The figure below is a graph showing a company’s productivity over a year.

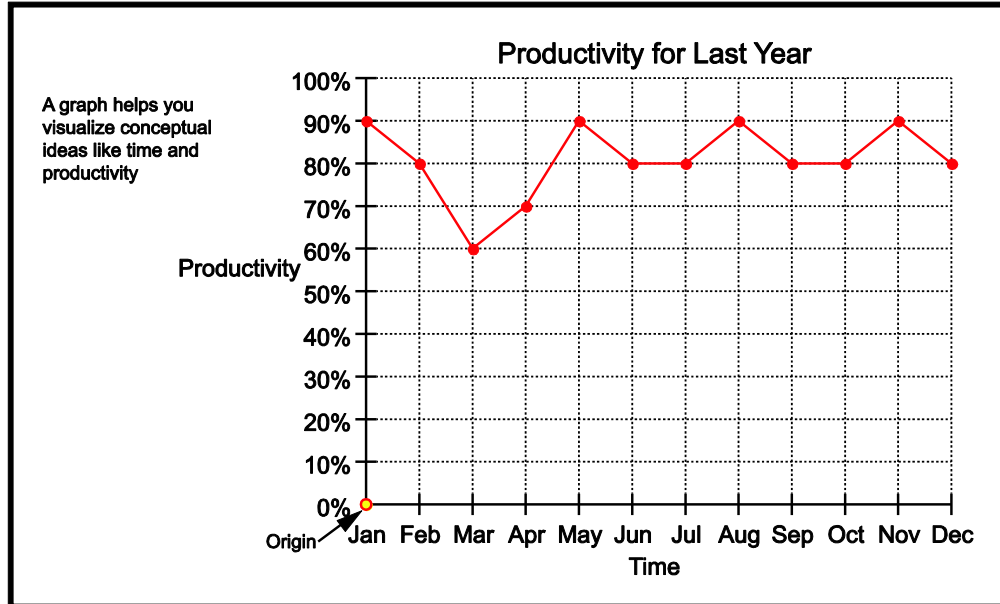


Figure 1.14: Graph example used to illustrate a coordinate system

In the figure above, the horizontal base line represents time. The increment of the time baseline is specified in months. One whole year is the range January through December. The vertical baseline represents productivity. The increment for this baseline is specified in 10% increments and ranges from 0% to 100% productivity. The point at which the horizontal baseline and the vertical baseline cross is called the origin.

In order to make this graph, a person must have the productivity data for the year. To plot the point for January, they locate January on the horizontal baseline and then move up vertically until they are parallel with the value of 90% on the vertical baseline. To plot the point for February, they locate February on the horizontal baseline and then move up vertically until they are parallel with the value of 80% on the vertical baseline. This procedure is repeated for every month of the year. Once all of the points are plotted, a line or curve can be passed through each point to show anyone at a glance how the company did last year.

A graph is very similar to the workpiece coordinate system used with CNC. Look at the figure below.

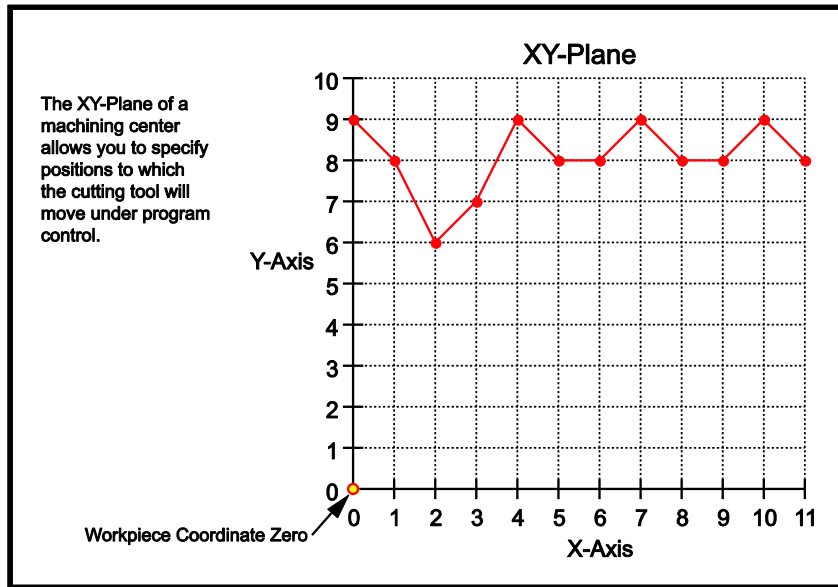


Figure 1.15: The coordinate system of a machining center (XY plane)

For the workpiece coordinate system used with CNC machining centers, the horizontal baseline represents the positions or coordinates of the X-axis. The vertical baseline represents the positions or coordinates of the Y-axis. (The Z-axis is at a right-angle to the page, toward and away from you. For now, let's concentrate on the X and Y axes.)

The increment of each base line is given in linear units. Working in the inch mode, each increment is given in smallest increment programmable on the CNC. For many CNCs, the smallest programmable increment is 0.0001 inches, meaning each CNC axis has a very fine grid. If you work in the metric mode, the increment will be in millimeters. In the metric mode, the smallest programmable increment typically is 0.001 mm.

The range for each axis is the amount of travel in the axis (from one over-travel limit to the other).

What About the Z-Axis?

The figure above describes only two of the machining center's axes, X and Y. The Z-axis behaves in exactly the same manner as X and Y. There is a zero point, and plus and minus polarity, the same programmable increment and the range is the maximum travel for the Z-axis. When taken all together, the X, Y, and Z provide you with a three dimensional grid. It is within this grid that you will be specifying positions (coordinates) that your tools move to, as we show in the figure below.

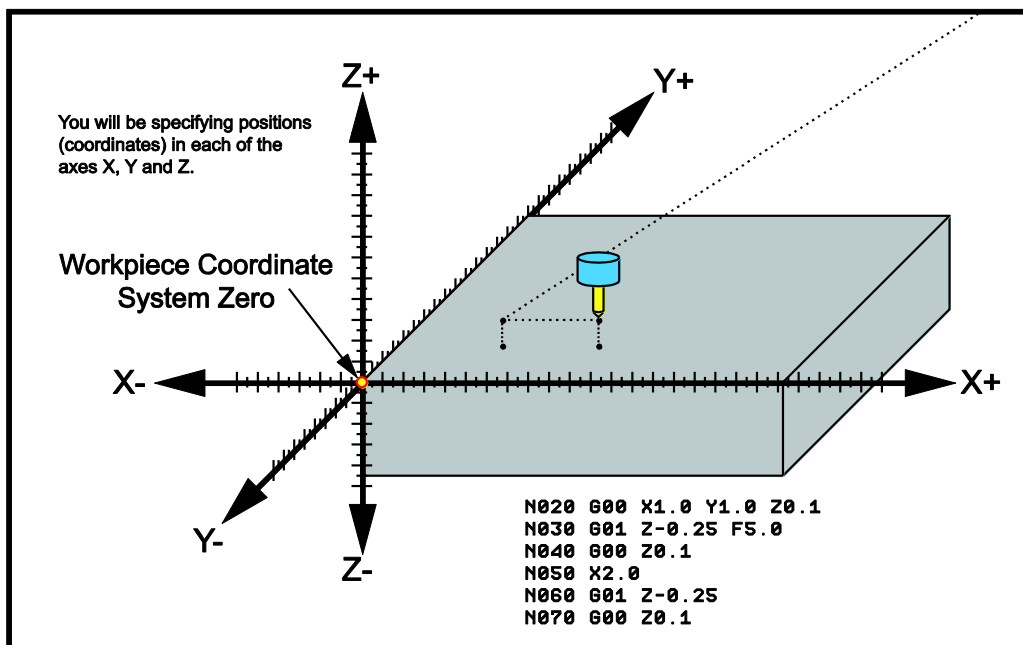


Figure 1.16: Three dimensional coordinate system of a machining center

Understanding Polarity

In the graph example previously illustrated, all the points plotted are above and to the right of the origin point. The area above and to the right of the two baselines is called a quadrant. This particular quadrant is quadrant number one where the coordinates in both axes are positive. The person creating the productivity graph intentionally planned for coordinates to fall in quadrant number one in order to make the graph easy to read. It is not uncommon on CNC machines that end points within the program fall in other quadrants. When this happens, at least one of the coordinates must be specified as negative or minus value.

Each CNC axis has a polarity. You also know that since sometimes the cutting tool moves and sometime the table moves, it can be confusing to remember which way is plus and which way is minus. (Consider the X and Y axes of a C-frame style vertical machining center, for example. As the table moves to the left, it is moving in the X plus direction.) To make things easier, we asked you to view polarity as if the tool was actually moving in each axis.

The workpiece coordinate system makes determining the polarity of coordinates used in a program very simple. The figure below shows the polarity for the X-axis in the workpiece coordinate system.

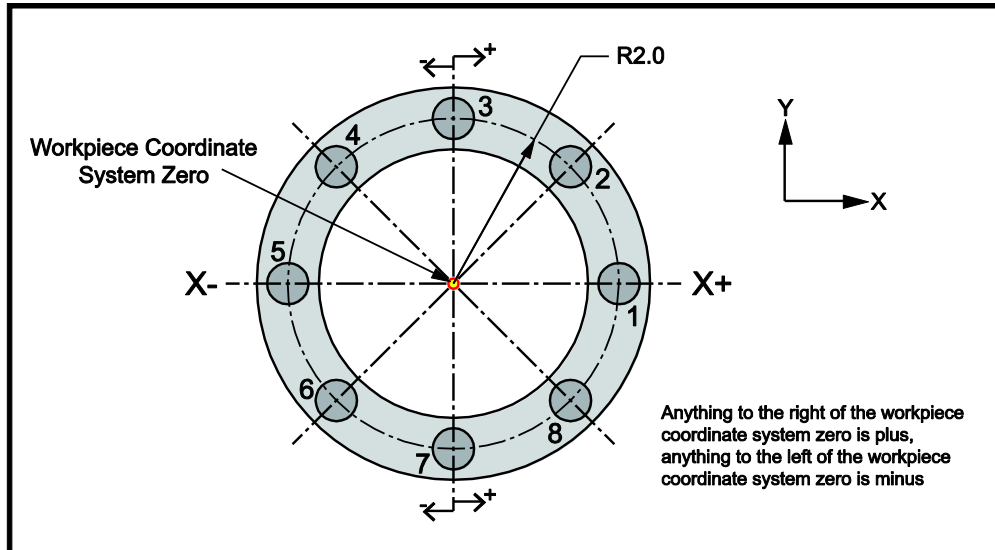


Figure 1.17: X-axis polarity

Notice that polarity is based upon the location of the workpiece coordinate system zero—as it will be for all axes. For X, anything to the right of the workpiece coordinate system zero is positive (plus). Anything to the left of workpiece coordinate system zero is negative (minus).

A CNC will assume that a coordinate is positive (plus) unless a minus sign is specified. The CNC word X2.0, for example, specifies a position along the X-axis of positive 2.0-inches. Some CNCs may generate an alarm if the plus sign is included within the word, meaning you must let the control assume positive values. It is common practice to only include a polarity sign for negative (-) values.

The workpiece coordinate system zero must also be specified in the Z-axis. The figure below shows the XZ plane (looking at a vertical machining center from the front).

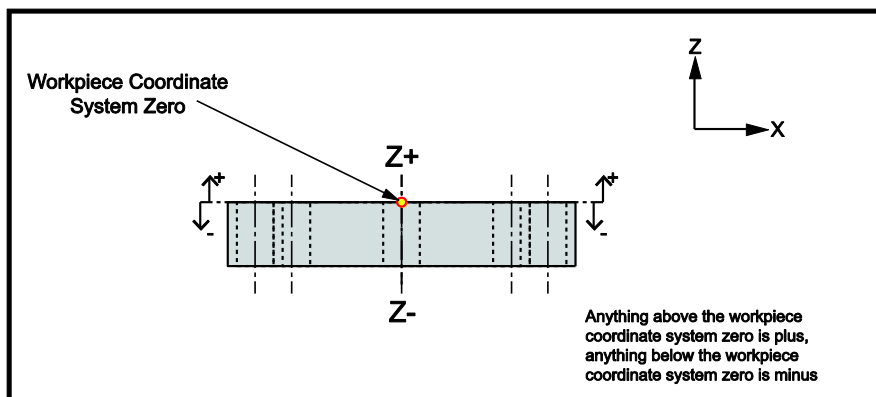


Figure 1.18: Polarity for the Z-axis

Anything above workpiece coordinate system zero in Z is positive (plus). Anything below workpiece coordinate system zero point is negative (minus).

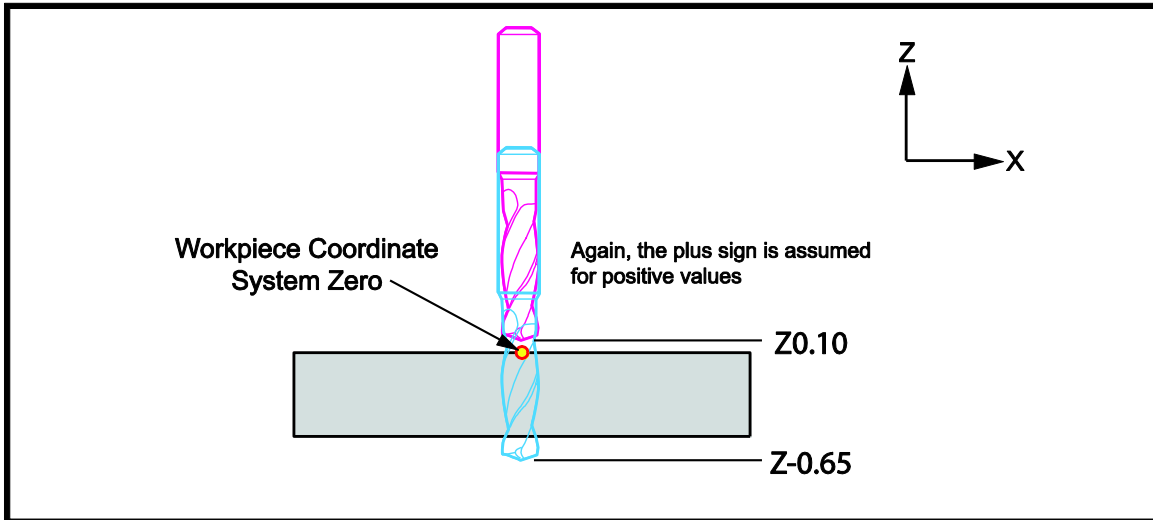


Figure 1.19: Example of Z-axis polarity

Notice that we have selected the top of the workpiece as the workpiece coordinate system zero in Z (which is a common program zero point for vertical machining center applications). Any tool position above the top of the workpiece is positive (plus) in Z. Any position below the top of the workpiece is negative (minus) in Z.

Wisely Choosing the Workpiece Coordinate System Zero Location

As the programmer, you determine the workpiece coordinate system zero point location for every program you write. Theoretically, the workpiece coordinate system zero could be placed at any location. As long as all the coordinates used in your program are specified from that workpiece coordinate system zero point, the program will function correctly. However, a wise selection of the workpiece coordinate system zero point will make programming much easier. It may also make it easier for the setup person.

Absolute Versus Incremental Modes

Though we have not actually said so yet, when you specify coordinates from the workpiece coordinate system zero point, it is called the absolute positioning mode. The absolute positioning mode is specified using a G90 word. Once a G90 is specified, all coordinates are taken to be from the program zero point since G90 is modal.

Everything introduced to this point has been related to the absolute positioning mode. And again, the point of reference for absolute positioning mode is the workpiece coordinate system zero point.

There is another method of axis positioning called the incremental positioning mode. G91 is used to specify incremental mode. Unlike absolute mode, the point of reference for all specified positions in the incremental mode is the tool's current position—the location of the tool at the beginning of the motion.

In the incremental mode, each movement is specified as a distance and direction from the tool's current position to the next position. At first glance, it may seem easier to work in the incremental mode than in the absolute mode. But you will soon find that programming with incremental positioning is quite difficult and error-prone. And by the way, if you make a mistake in a series of incrementally specified motions, every movement from the mistake on will also be incorrect.

While there are some excellent applications for incremental mode (we'll show them in Key Concept 6), beginning programmers should work exclusively in the absolute mode. All examples in this course (with the exception of some we show in Key Concept 6) use the absolute mode.

Any series of motions can be commanded in either the absolute or incremental mode. Look at the figure below.

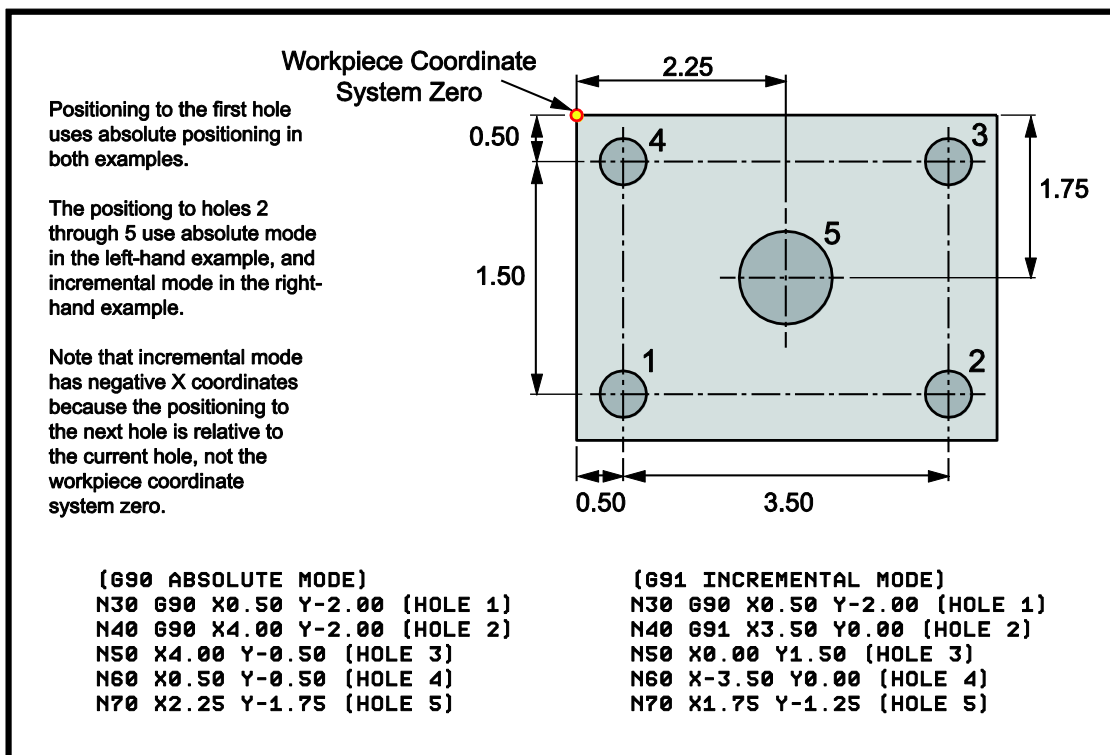


Figure 1.20: Movements can be specified in both the absolute and incremental positioning mode.

As you can see, absolute positioning makes more sense. Coordinates often match print dimensions – but even when they don't—the point of reference for each position is the same—the workpiece coordinate system zero. Incremental positioning doesn't make much sense. Positions are nothing more than a whole series of disjointed movements, each taken from the tool's previous position.

Key Concept 2: You must prepare to create programs

While this Key Concept does not involve any programming words or commands, it is among the most important of the Key Concepts. The better prepared you are to write a CNC program, the easier it will be to develop a workable program.

Preparation for programming is especially important for entry-level programmers. For the first few programs you write, you will have trouble enough remembering the various CNC words—remembering how to structure the program correctly—and in general—you'll have trouble getting familiar with the entire programming process. The task of programming is infinitely more complicated if you are not truly prepared to write the program in the first place.

Preparation and Time

Without adequate preparation, writing a CNC program can be compared to working on a jigsaw puzzle. A person doing the puzzle has no idea where each individual piece will eventually fit. The person makes a guess and attempts to fit the pieces together. Since the person has no idea as to whether pieces will fit together, it is next to impossible to predict how long it will take to finish the puzzle.

In similar fashion, if you attempt to write a CNC program without adequate preparation, you will have a tendency to piece-meal the program together in much the same way as a person doing a jigsaw puzzle. You will not be sure that anything will work until it is tried. The program may be half finished before it becomes obvious that something is seriously wrong. Worse, the program may be completed and being verified on the CNC machining center before a critical error is found.

CNC machine time is much more expensive than your time. There is no excuse for wasting precious machine time for something as avoidable as a lack of preparation.

Preparation and Safety

Wasted time is but one of the symptoms of poor preparation. Indeed, it may be the least severe one. Poor preparation will often result in all kinds of mistakes.

A CNC machining center will follow a CNC program's instructions exactly. While the CNC may display an alarm if it cannot recognize a given command, it will give absolutely no special consideration to motion mistakes. Indeed, a CNC cannot detect most motion mistakes. The level of problem encountered because of motion mistakes ranges from minor to catastrophic.

Plan the Sequence of Machining Operations

Process sheets, also called routing sheets, are used by most manufacturing companies to specify the sequence of machining operations that must be performed on a workpiece during the manufacturing process. The person who actually prepares the process sheet must have a good understanding of machining practice, and must be well acquainted with the various machine tools the company owns. This person determines the best way to produce the workpiece in the most efficient and inexpensive possible way, given the company’s available resources.

The sequence of machine operations used to machine the workpiece will have a dramatic impact on the success of the program. If the process is correct, the workpiece will be machined efficiently and pass inspection. If the process is poor, the workpiece will not be machined correctly no matter how well the program is written. If you are new to developing a sequence of machining operations, you should seek help whenever there is a question as to whether your planned machining process will work.

Part no.:		Date:		Machining Process Planning Form		
Part name:		Programmer:				
Machine:		Material:				
Seq.	Operation Description	Tool	Station	Speed	Feed	Note

Figure 2.1: Example sequence of operations form

Notice how this form guides you to document the sequence of machining operations that your program will use. Months or years after a CNC program is developed, there may be a need to revise it. If the person doing the revision can review a completed machining process planning form for the workpiece, it will be much easier to make the necessary changes.

The last reason we will give to plan the process first is to simply help you remember the operations to perform during programming. Remember, beginners tend to make mistakes of omission. You will have enough to think about when it comes to remembering the various commands needed in the program. The process planning form is a step-by-step set of instructions to machine the workpiece. It can be used as a check-list. Without this form, you will be prone to omitting important operations from the CNC program.

Develop the Cutting Conditions

Before a program can be written, cutting conditions must be determined for all the cutting tools used in the program. Each cutting tool will need a spindle speed in revolutions-per-minute (rpm) and a feedrate in inches-per-minute (ipm). For roughing tools, like rough milling cutters and rough boring bars, you must also determine a depth-of-cut for the tool—as well as how much finishing stock you will leave for the finishing tool. You must also determine whether or not to use coolant and, if so, what kind—flood, mist, through-the-tool, or high pressure, based upon the workpiece and cutting tool materials.

It is helpful to come up with the cutting conditions needed in the program while developing the sequence of machining operations—before you write the program. This will keep you from having to break out of your train of thought while programming. Using a machining process planning form like the one above, you will be able to document the speeds and feeds needed for programming.

The data provided by cutting tool manufacturers typically includes the cutting speed (in surface-feet-per-minute) and feedrate (in either inch-per-revolution or inch-per-tooth). This information is based upon the cutting tool material (high-speed steel, carbide, ceramic, etc.) and the workpiece material (mild steel, medium carbon steel, high carbon steel, stainless steel, aluminum, etc.). When appropriate, cutting tool manufacturers will also specify whether or not you should use coolant—as well as the recommended depth of cut for a roughing tool. In some cases, they will even provide recommendations about how the cutting tool should move as it machines the workpiece.

For machining center programs, you must of course, calculate the speed in rpm and feedrate in ipm. For rpm, you must know the recommended speed in sfm and the cutting tool diameter. For ipm, you must know the rpm and ipr feedrate.

Here is an example of a filled-in process planning form.

Part no.:		Date:		Machining Process Planning Form		
Part name: exercise part		Programmer: your name				
Machine: machining center		Material: mild steel				
Seq.	Operation Description	Tool	Station	Speed	Feed	Note
1	Rough face mill the top surface	4.0" face mill		382	12.224	
2	Rough mill the 2 ends	1.0" end mill		267	5.34	
3	Finish face mill the top surface	4.0" face mill		477	9.54	
4	Finish mill the 2 ends	1.0" end mill		325	5.2	
5	Center drill 5 holes	#3 center drill		1430	5.72	
6	Drill 2 1/4-inch diameter holes	1/4 drill		1070	4.28	
7	Drill 2 1/2-inch diameter holes	1/2 drill		535	3.74	

Figure 2.3: Example Machining Process Planning Form with Feeds and Speeds

Do the Math and Mark-up the Print

As stated in the Preface of this text, the word numerical in computer numerical control implies a strong emphasis on numbers and math. Most college curriculums related to CNC do require a strong math background. However, most forms of CNC equipment require less math than you might think. Believe it or not, many CNC machining center programs can be completely prepared solely with simple addition and subtraction. A basic knowledge of right angle trigonometry is also helpful, but not always mandatory.

Marking up the Workpiece Engineering Drawing

You must have a good copy of the workpiece engineering drawing. You should have your own working copy, and be allowed to do whatever you need to with the print to help you with the programming task. Your marked-up copy of the print should be kept with the program as part of the documentation for the program.

Depending on the complexity of the workpiece to be produced, interpreting a workpiece drawing can range from quite simple to very difficult. Once you study the workpiece engineering drawing and understand the machining operations that must be performed, you should mark-up the print in any way that makes programming easier. The first thing we recommend is to take a high-lighting pen of a bright color and mark those surfaces on the print that require machining operations by your CNC program. Especially helpful for complicated workpieces, this helps you narrow down just what the program must do.

Additionally, you should indicate any information required for programming on the print. The location of program zero, the placement of workholding devices (fixtures, vises, etc.), and clamps should be included on the marked-up print. If there is room on the print, you can also include coordinates needed for programming.

Doing the Math

How dimensions are described on the workpiece engineering drawing will determine with how much math is required to write the program. In progressive companies, design engineers use *datum surface dimensioning* techniques. When datum surface dimensioning is used, each dimension on the print will be specified from one surface in each axis (the datum surface). This dramatically reduces the amount of math required to write a program.

The figure below shows an elaborate example of calculating coordinates needed for a program,

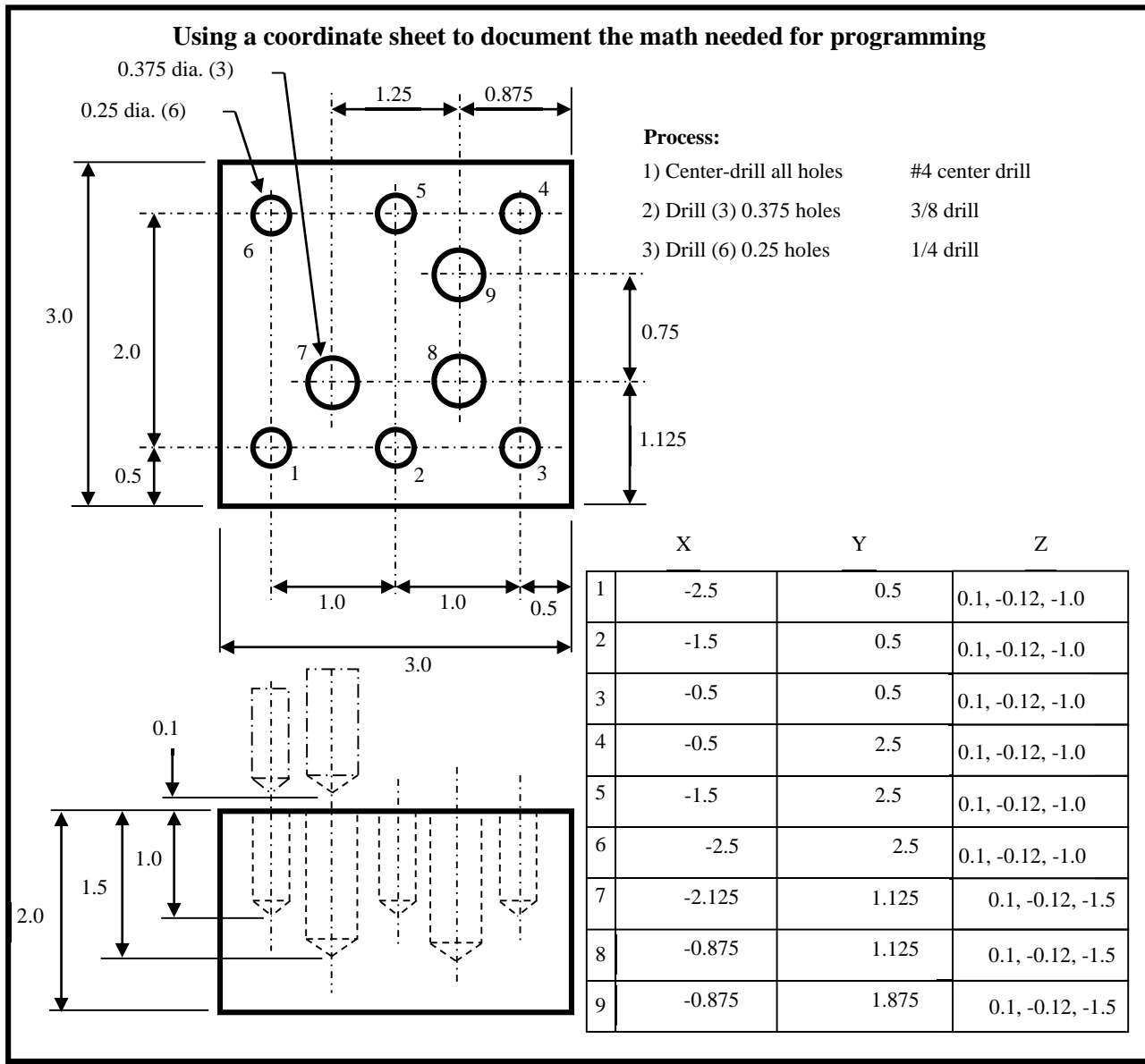


Figure 2.7: Example showing how to document the math needed in a program

In the Z column of the coordinate sheet in the figure above, notice there are three values. The first (0.1) is the approach position. The same approach distance is used for all tools. The second (-0.12) is the hole bottom position for the center drill. The third (-1.0 or -1.5) is the hole bottom position for the drill.

Plan the Workholding Set-up

The programmer is usually responsible for developing the workholding setup required to hold the workpiece during machining. Even for simple work holding setups, you should make a drawing or sketch indicating how the setup is to be made. For example, a sketch showing where a vise is placed on the machine's table may adequately instruct the setup person.

Most companies use a setup sheet to help the setup person understand everything they need to know about how a given setup must be made. Most setup sheets will include a sketch of the setup (possibly even a photograph of the setup once it has been made), the location of program zero, a list of cutting tools (including a list of components needed for each tool), and in general, any other instructions necessary for getting the job up and running. The figure below shows an example of a universal setup sheet. We call it a universal setup sheet because this form is used for all setups made on a given CNC machine tool.

Part no.:		Date:		Setup Sheet
Part name:		Programmer:		
Machine:		Program no:		
Stat.	Tool description	Offset	Insert	Instructions:
Fixture/vise:				Sketch of setup:
Clamps:				
Other notes:				

Figure 2.10: An example universal setup sheet

This universal setup sheet does not include a detailed list of components for the cutting tools used in the program. Many companies do include a more complete tool list, possibly on a separate page. This setup sheet also does not include a complete list of the workholding tools. Again, many companies do include this kind of information so someone (probably *other* than the setup person) can be gathering all needed components even before the setup is made.

Key Concept 3: You must understand the basic motion types

Motion control is at the heart of any CNC machine tool. CNC machining centers have at least three ways that motion can be commanded. Understanding the motion types you can use in a program will be the focus of Key Concept 3.

What is Interpolation?

When a single linear axis is moving (X, Y, or Z), the motion will be along a perfectly straight line. For example, look at the figure below.

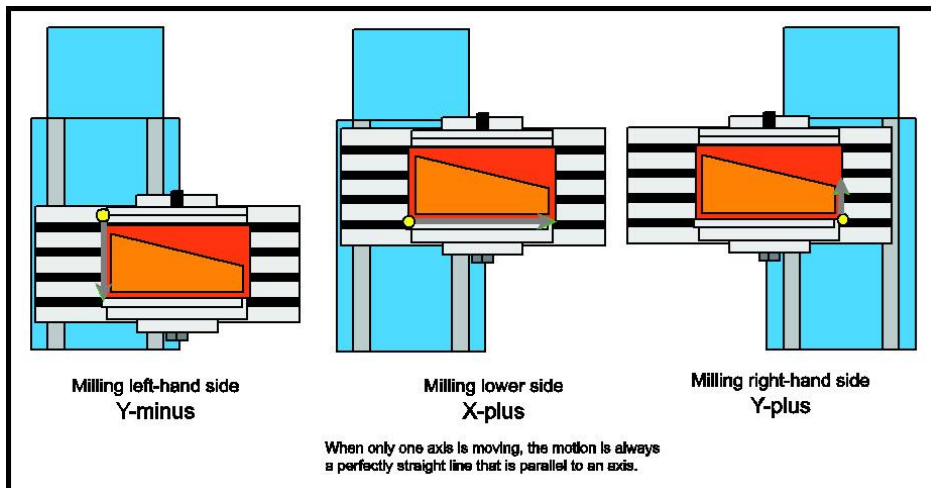


Figure 3.21: A perfectly straight motion will occur if only one axis is moving

When milling the left side of the workpiece (left view above), only the Y-axis is moving. And since the Y-axis is a linear axis, the motion is perfectly straight and parallel to the Y-axis. When milling the lower surface (middle view above), only the X-axis is moving and the motion is perfectly straight and parallel to the X-axis. The same goes for milling the right side (right view above)—only the Y-axis is moving and the motion has to be perfectly straight.

But notice that the upper side of this workpiece is tapered. It will require that both the X and Y axes to move in a coordinated manner, as shown in the figure below.

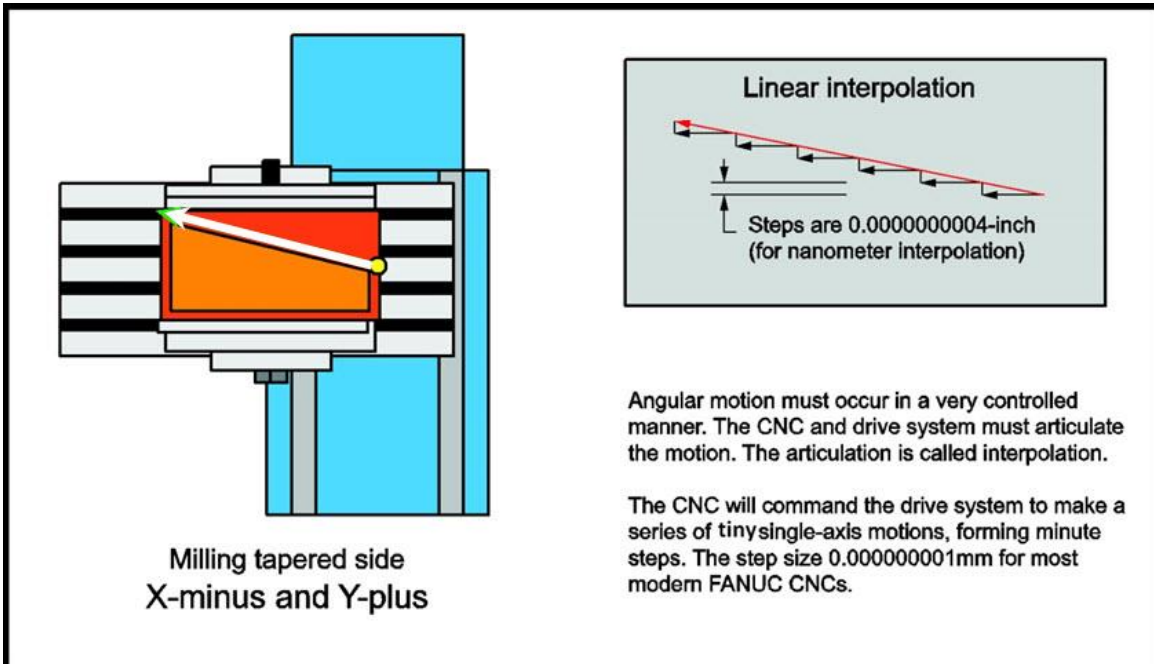


Figure 3.22: Milling the upper side of this workpiece requires both the X and Y axes move in a controlled manner

Two axes must move if the tool motion is at an angle as shown in the figure above. In CNC terms, this kind of motion is called interpolation.

The CNC breaks the two-axis motion up in to a series of very tiny steps. The step size for current FANUC CNCs is 0.000000001 millimeters or 0.000000004-inches. Even with older CNCs the steps are so small that you cannot see or measure them with most measuring devices. The smaller the step size, the finer the machine's resolution and the more precisely it will follow your commanded motions. For all intents-and-purposes, all machined surfaces will appear to be perfectly straight and without steps.

Actually, it should be comforting to know that there are only three primary ways to cause axis motion—rapid, linear interpolation, and circular interpolation. Just about every motion a CNC machining center makes can be divided into one of these categories. Once you master these three motion commands, you will be able to generate the motions required to machine a workpiece. The figure below illustrates these 3 primary motion types.

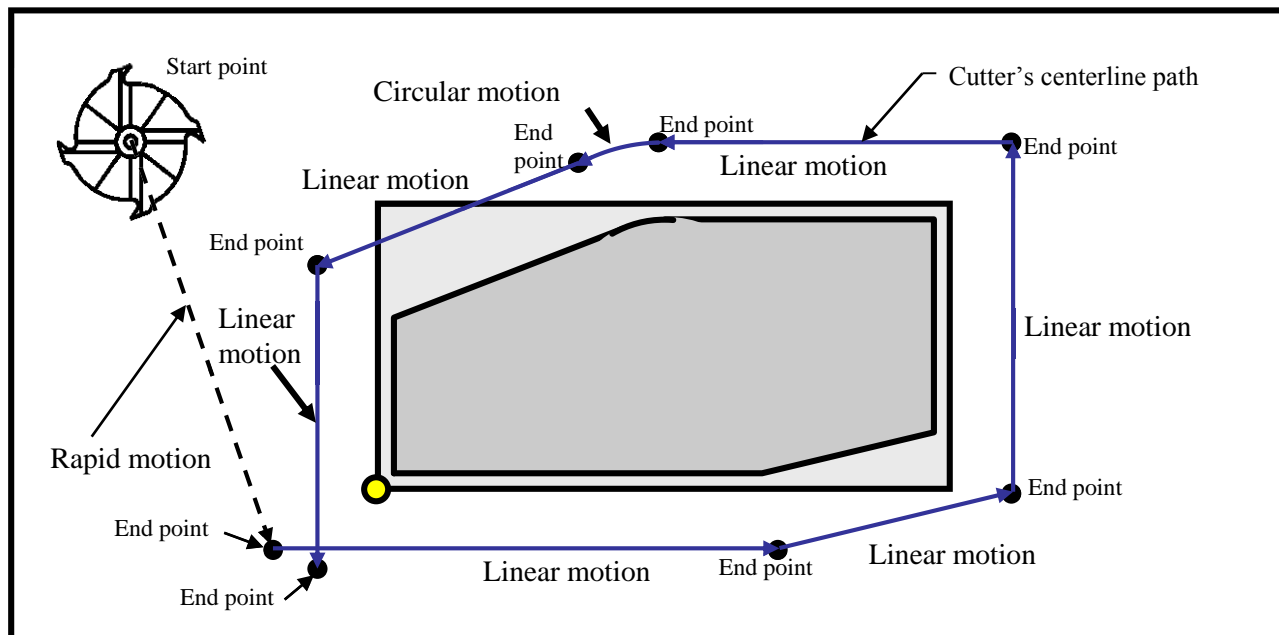


Figure 3.23: The three kinds of motion available with all CNC machining centers

As you will see, it is actually quite easy to specify motion commands within a CNC program. In general, each motion will require you to specify the kind of motion (rapid, linear interpolation, or circular interpolation) along with the motion's end-point (the coordinates at the end of the motion). Linear and circular interpolated motion additionally require that you specify the rate at which the axes will move (feedrate). Circular interpolation also requires that you specify either the radius of the arc or the coordinates of the center of the arc.

Motion Type Commonalities

All motion types share five things in common:

Modal - this means that once programmed a motion type will remain in effect until it is changed. When more than one consecutive movement of the same motion type is programmed, you need only include the motion type G-code in the first block of the series of movements.

End-point specification - each motion command requires the coordinates of the end-point of the motion to be specified. The CNC assumes the start-point for the motion is the current tool position. Think of the motion commands that form a tool-path as being a series of connect-the-dots.

Absolute or incremental modes - all motion commands are affected by whether or not you specify coordinates in the absolute or incremental positioning mode. In the absolute positioning mode (G90), the end-points are specified relative to the workpiece coordinate system zero. In the incremental positioning mode (G91), the end-points are specified relative to the tool's current position. Beginning programmers should concentrate on specifying coordinates in the absolute positioning mode.

Specify only moving axes - only specify the axes that will move in a motion command block. If specifying a motion in only one axis, only one axis specification (X, Y, or Z) needs be included in the motion command. Axes that are not moving can and should be left out of the command.

Leading zero suppression - the leading zeros can be left out of the G-codes related to motion types commands. This means the actual G-codes used to instate the motion types can be programmed in one of two ways. G00 and G0 (stated G-zero-zero and G-zero) mean exactly the same thing to the control, as do G01 and G1, G02 and G2, G03 and G3. However, all examples in this course include the leading zero (G00, G01, G02, G03).

G00 Rapid Motion (also called positioning)

Rapid motion is used to position a cutting tool at high-speed. Under normal conditions, G00 (stated G-zero-zero) will cause the machine to move at its fastest possible rate—which is called the machine's rapid rate. The rapid rate will vary from one machine to another—but it is always very fast. Several current CNC machining centers boast rapid rates well over 1,500 inches-per-minute, which means they move 25-inches in just one second.

Due to this very fast—and somewhat scary—motion rate, most machine tool builders will allow you to override the machine’s rapid rate during a program’s verification using a multi-position switch called rapid traverse override. Though this feature varies from machine to machine, most machining centers allow you to slow the rapid rate significantly. This relieves some of the stress of running a program for the first time, and minimizes the potential for problems if a mistake has been made in a rapid motion command.

The figure below shows an example of rapid motion.

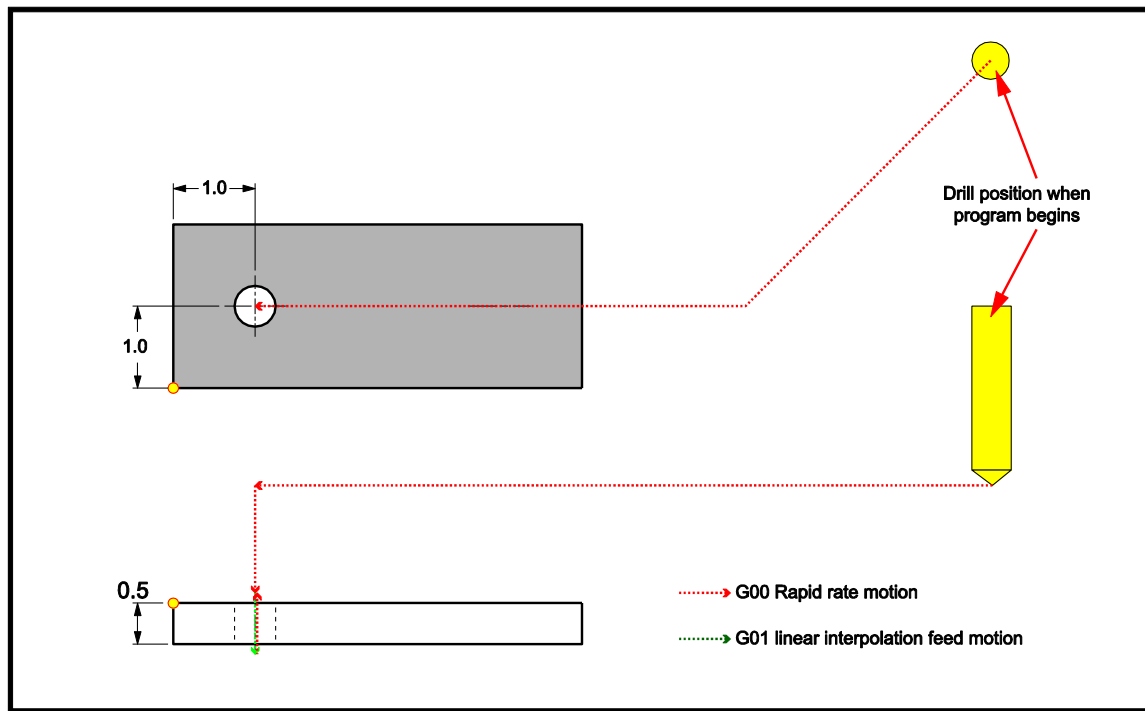


Figure 3.2: Rapid motion example

Program with comments:

```
O0010 (Program number)
N010 G20 G90 G54 (Select inch & absolute modes, workpiece coordinate system setting offset #1)
N020 T01 M06 (select drill in tool location 1)
N030 S400 M03 (start spindle fwd at 400 rpm)
N040 G00 X1.0 Y1.0 (Rapid to hole-location)
N050 G43 H01 Z0.1 (Instate tool length compensation, rapid to just above work surface)
N060 G01 Z-0.7 F5.5 (Feed linear to hole-bottom at 5.5 ipm)
N070 G00 Z0.1 (Rapid retract from hole)
N080 G91 G28 Z0 (Rapid to Z-axis zero return position)
N090 G28 X0.0 Y0.0 (Rapid to X/Y axes zero return position)
N100 M30 (End of program)
```

The tool is well away from the workpiece when the program in the figure above starts (possibly at the zero return position). In block N010, G20 tells the CNC that the coordinates are specified in inches. The G90 tells the CNC that all up-coming coordinates will be specified from the workpiece coordinate system zero. The G54 word tells the CNC to look in workpiece coordinate system offset 1 to find the workpiece coordinate zero offset values. Block N020 selects the tool. In block N030, the S400 M03 starts the spindle at 400 rpm in the forward direction.

The G00 word in block N040 specifies the rapid motion mode, so all motions from this point will be at rapid until the motion type is changed. In this block, the drill will move at the rapid rate to above the hole in XY (X1.0 Y1.0).

When do you Use Rapid Motion?

Though it may be obvious, you should use the rapid motion command whenever the cutting tool is not machining the workpiece during the motion. In this way, you can minimize a program’s air-cutting time (reducing cycle time). This includes approaching

surfaces to be machined, retracting tools to the machine's tool changing position, and any other non-cutting operation that occurs (getting from one cutting position to another). A good rule-of-thumb is "If the tool is not cutting, it should be moving at rapid".

Certain other commands automatically cause the machine to move at its rapid rate. A G28 command, which sends the axes to the zero return position will also be done at the rapid rate. The command,

```
N100 G91 G28 Z0.0
```

for example, will send the machine to its Z-axis zero return position. This is the tool change position for most vertical machining centers. Though a G00 is not included in this command, the machine will move at its rapid rate during this move.

G01 Linear Interpolation (straight-line motion)

This motion type causes the machine to move along a perfectly straight path in one, two, or three axes. The control will calculate and interpolate the path between the start-point and the end-point of the motion automatically, no matter what angle of motion is required. So you simply specify the end-point for the motion. The G-code used to command this motion type is G01 (stated G-zero-one).

The motion rate for a linear interpolation move is programmable. It is specified with an F-word (specifying a feedrate for the motion). With most machining centers, the feedrate is typically specified in per-minute fashion (inches-per-minute in the inch mode or millimeters-per-minute in the metric mode). Today's machining centers also allow feedrate to be specified in distance-per-revolution (inches-per-revolution or millimeters-per-revolution). If your machining center allows both feedrate types, two G-codes will be used to specify which feedrate mode you desire (G94 for distance-per-minute, G95 for distance-per-revolution).

Like motion types, the feedrate word is modal. If a series of cutting motions will be machined at the same feedrate, the F-word need only be included in the first cutting motion block.

The linear interpolation motion command is used primarily to machine straight surfaces. Examples of when a G01 command can be used include drilling a hole to depth in the Z-axis and milling a straight or angular surface.

The figure below shows an example of linear interpolation cutting commands.

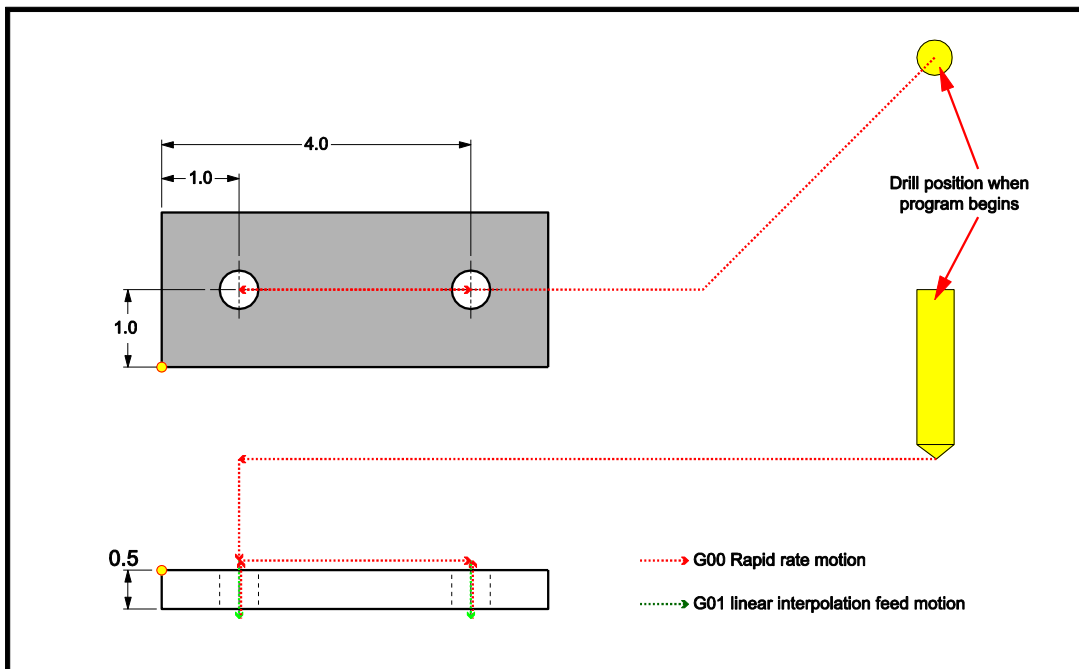


Figure 3.3: Linear interpolation cutting command example

Program with comments:

```

O0011 (Program number)
N010 G20 G90 G94 G54 (Inch, absolute, ipm modes, select work offset #1)
N020 S600 M03 (Start spindle fwd at 600 rpm)
N030 G00 X1.0 Y1.0 (Rapid to first hole-location)
N040 G43 H01 Z0.1 M08 (Instate tool length compensation, rapid to just above work surface, start coolant)
N050 G01 Z-0.72 F4.0 (Drill hole to bottom at 4.0 ipm)
N060 G00 Z0.1 (Rapid retract from hole)
N070 X4.0 (Rapid to second hole)
N080 G01 Z-0.72 (Drill second hole at 4.0 ipm)
N090 G00 Z0.1 (Rapid retract from hole)
N100 G91 G28 Z0 (Rapid to Z-axis zero return position)
N110 G28 X0.0 Y0.0 (Rapid to X/Y axes zero return position)
N120 M30 (End of program)

```

G02 and G03–Circular Interpolation (circular motion)

Milling operations commonly require the machining of circular workpiece attributes. Consider, for example, the milling of a circular pocket. Frankly speaking, just about the only time you'll need to command a circular motion is when side milling. When circular motion is commanded, two axes (usually X and Y) will be moving together to form the motion.

Circular motion can be either clockwise or counter-clockwise. Two G-codes are involved–G02 specifies clockwise motion while G03 specifies counter-clockwise motion.

To determine whether a given XY circular motion is clockwise or counter-clockwise (G02 or G03), view the motion from the perspective of the cutter. In most cases, this means viewing the motion from above the workpiece engineering drawing. See the figure below for an example of clockwise and counter-clockwise motion. Notice that we're simply viewing the motion from above the workpiece engineering drawing.

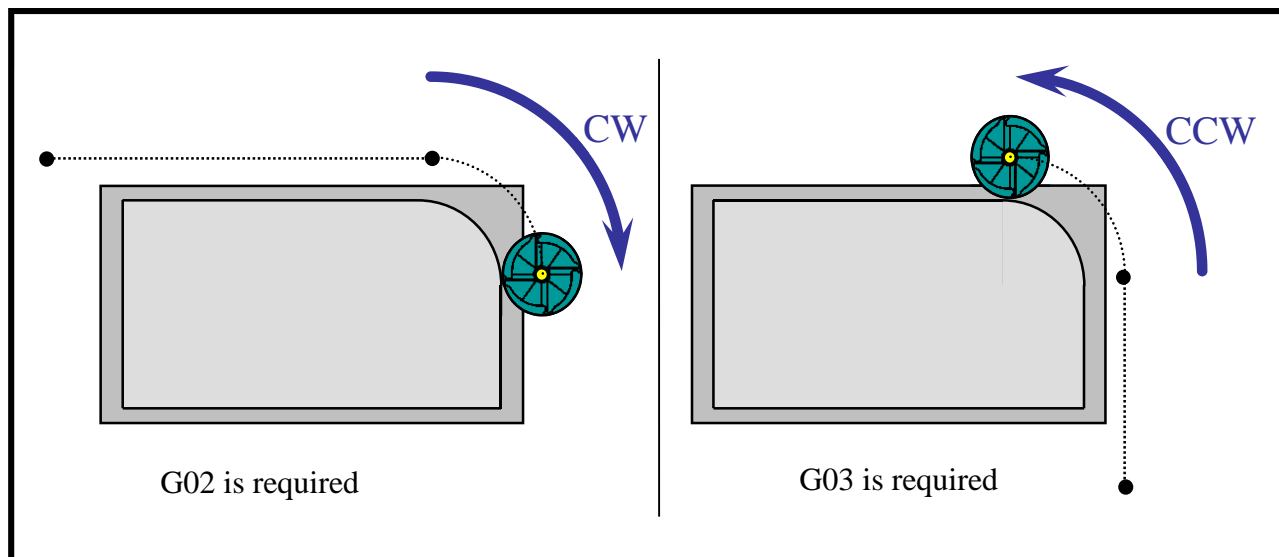


Figure 3.5: Clockwise versus counter-clockwise circular motion

Like linear interpolation, circular interpolation requires that a feedrate be specified (with an F-word). Feedrate is specified in distance-per-minute (inches- or millimeters-per-minute). And, feedrate is modal. Even if a feedrate is originally specified in a linear interpolations command, it will remain effective during subsequent circular interpolation commands.

Also as with linear interpolation motion, circular interpolation commands require that the end point of the circular motion be specified. The tool's position prior to the circular interpolation motion is the starting point for the circular motion.

Specifying Arc Size with the R-word

Circular motion commands also require that you specify the arc size of the circular path you are commanding. Today's CNCs allow you to do so with a simple R-word. With the R-word, you specify the radius of the arc being machined.

Note that the value of the R-word must correspond to the path you are programming-tool centerline or work surface. When programming the cutter's centerline path (which we will demonstrate until we present cutter radius compensation), the R-word must reflect the radius of the cutter's centerline path. When programming the work surface path (using cutter radius compensation), the R-word must reflect the workpiece radius being machined.

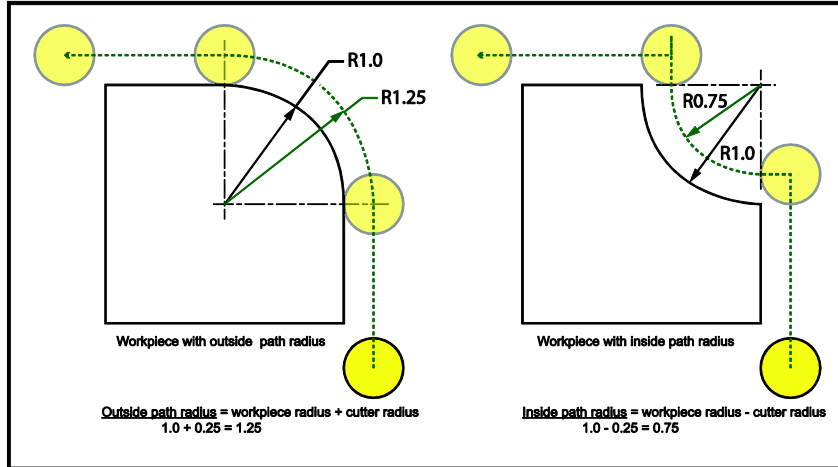


Figure 3.6: Outside radius tool centerline path versus inside radius tool centerline path.

With an outside radius as shown above, you must add the cutter radius to the workpiece radius to come up with the cutter's centerline path radius. If milling an inside radius (as would be the case when milling a circular pocket), you must subtract the milling cutter's radius from the workpiece radius to come up with the cutter's centerline path radius. The figure below shows a full example of circular motion.

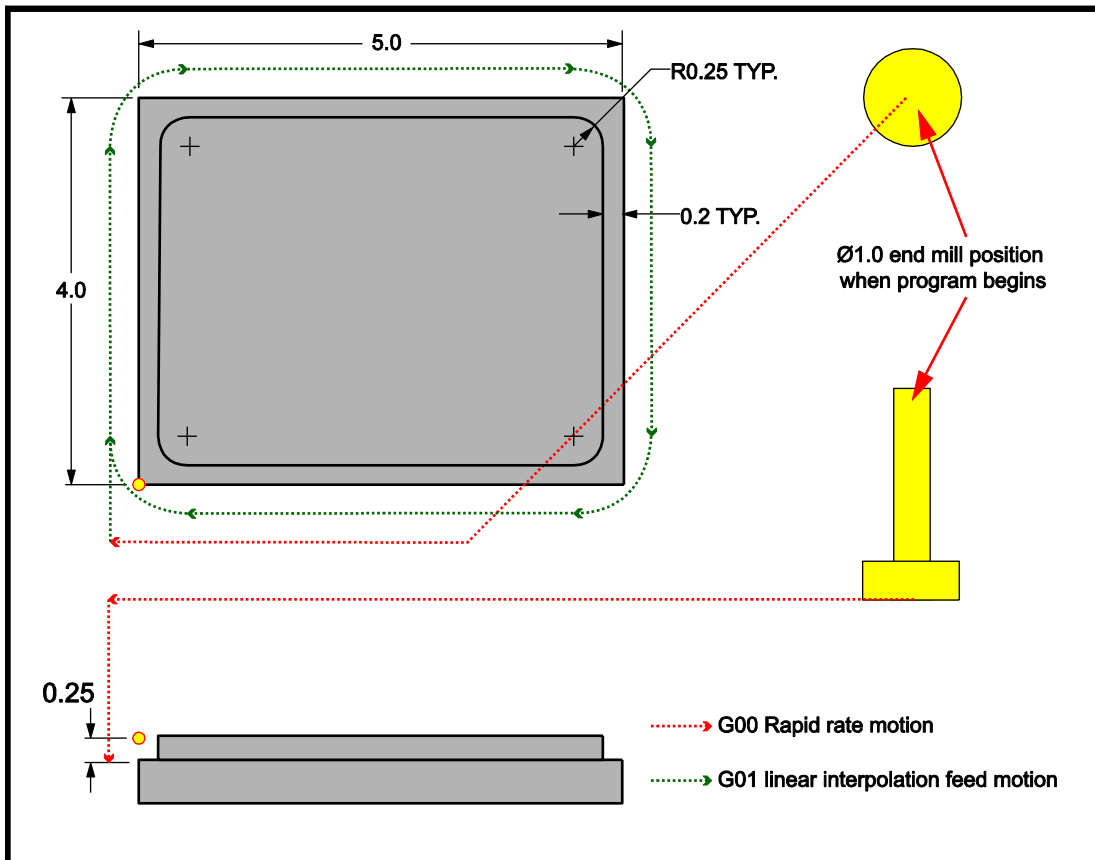


Figure 3.7: Drawing for example program showing circular motion

Program with comments:

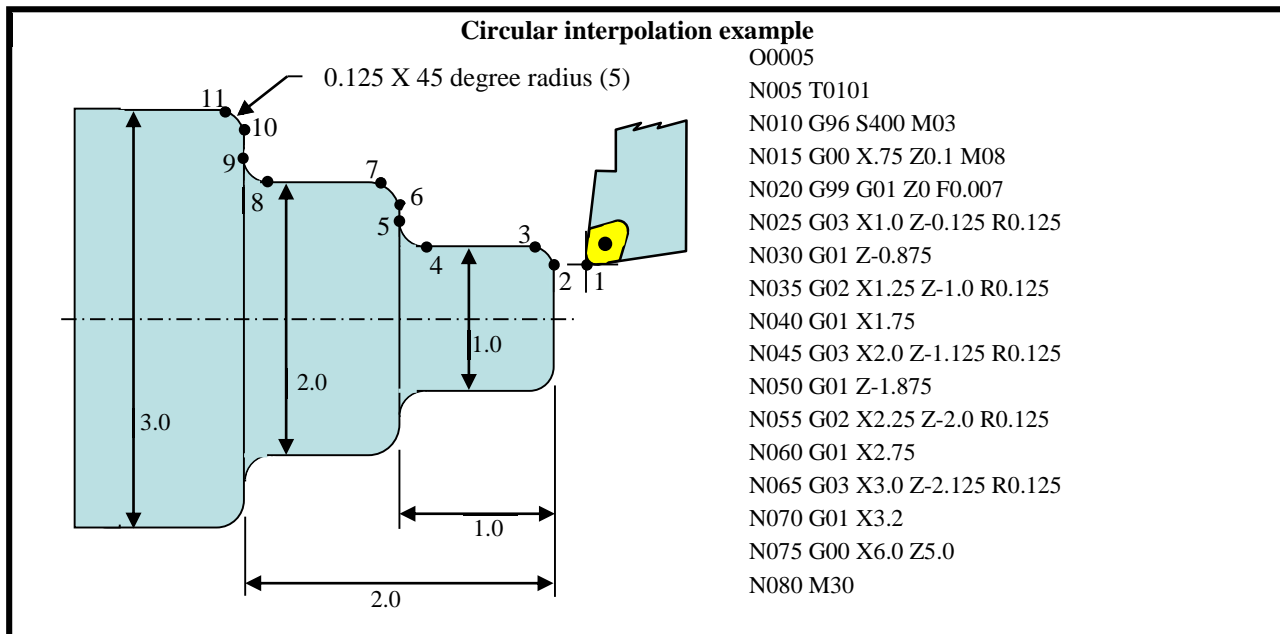
```

O0014 (Program number)
N010 G20 G90 G94 G54 (Inch, absolute ipm modes, work offset #1)
N020 T05 M06 (select tool #5)
N030 S400 M03 (Start spindle fwd at 400 rpm)
N040 G00 X-0.6 Y-0.3 (Rapid to approach position in XY)
N050 G43 H01 Z0.1 (Instate tool length compensation, move to just above workpiece)
N060 G01 Z-0.25 F30.0 (Fast feed to work surface)
N070 X4.55 F5.0 (Mill lower surface at 5.0 ipm)
N080 G03 X5.3 Y0.45 R0.75 (Mill lower-right radius)
N090 G01 Y3.55 (Mill right surface)
N100 G03 X4.55 Y4.3 R0.75 (Mill upper-right radius)
N110 G01 X0.45 (Mill upper surface)
N120 G03 X-0.3 Y3.55 R0.75 (Mill upper-left radius)
N130 G01 Y0.45 (Mill left surface)
N140 G03 X0.45 Y-0.3 R0.75 (Mill lower-left radius)
N150 G00 Z0.1 (Rapid to just above workpiece)
N160 G91 G28 Z0.0 (Rapid to the Z-axis zero return position)
N170 G28 X0.0 Y0.0
N180 M30 (End of program)

```

A turning center example

Turning centers have the same basic motion types, and programming them is much the same as it is for a machining center.



Program with comments:

```

O0005 (Program number)
N005 T0101 M42 (Index to station number one, select high spindle range)
N010 G96 S400 M03 (Start spindle fwd at 400 SFM)
N015 G00 X.75 Z0.1 M08 (Rapid to point 1, turn coolant on)
N020 G99 G01 Z0 F0.007 (Feed to point 2)

```

N025 **G03 X1.0 Z-0.125 R0.125** (CCW circular motion to point 3)
N030 **G01 Z-0.875** (Straight move to point 4)
N035 **G02 X1.25 Z-1.0 R0.125** (CW circular motion to point 5)
N040 **G01 X1.75** (Straight move to point 6)
N045 **G03 X2.0 Z-1.125 R0.125** (CCW circular motion to point 7)
N050 **G01 Z-1.875** (Straight move to point 8)
N055 **G02 X2.25 Z-2.0 R0.125** (CW circular motion to point 9)
N060 **G01 X2.75** (Straight move to point 10)
N065 **G03 X3.0 Z-2.125 R0.125** (CCW circular motion to point 11)
N070 **G01 X3.2** (Straight move off workpiece)
N075 G00 X6.0 Z5.0 (Rapid away from workpiece)
N080 M30 (End of program)

Key Concept 4: You must understand the compensation types

Compensation types vary between machining centers and turning centers, though the most basic reasons for using them remain the same.

Machining center compensation types:

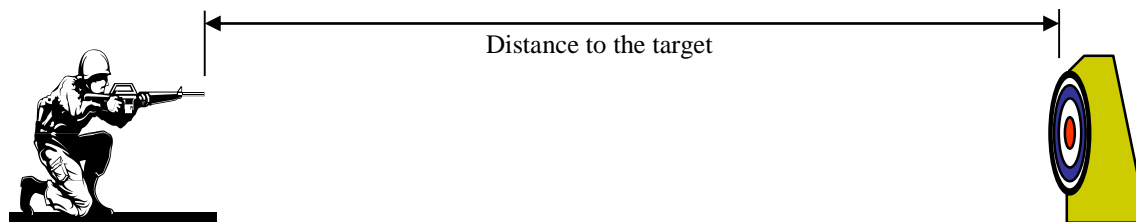
- Tool length compensation
- Cutter radius compensation
- Fixture offsets

Turning center compensation types:

- Geometry offsets
- Wear offsets
- Tool nose radius compensation

What are Compensations and Why are they Needed?

When you compensate for something, you are allowing for some unpredictable (or nearly unpredictable) variation. A race car driver must compensate for the condition of the race track before a curve can be negotiated. In this case, the unpredictable variation is the condition of the track. An airplane pilot must compensate for the wind direction and velocity before a heading can be set. For them, wind direction and velocity are the unpredictable variations. A marksman must compensate for the distance to the target before a shot can be fired—and the distance to the target is the unpredictable variation. The marksman analogy is remarkably similar to what happens with most forms of CNC compensation. Let's take it further...



Before a marksman can fire a rifle, they must judge the distance to the target. If the target is judged to be fifty yards away, the sight on the rifle will be adjusted accordingly. When the marksman adjusts the sight, they are compensating for the distance to the target. But even after this preliminary adjustment and before the first shot is fired, the marksman cannot be absolutely sure that the sight is adjusted perfectly. If they've incorrectly judged the distance—or if some other variation (like wind) affects the sight adjustment—the first shot will not be perfectly in the center of the target.

After the first shot is fired, the marksman will know more. If the shot is not perfectly centered, another adjustment will be needed. And the second shot will be closer to the center of the target than the first. Depending upon the skill of the marksman, it might be necessary to repeat this process until the sight is perfectly adjusted.

With all forms of CNC compensation, the setup person will do their best to determine the compensation values needed to perfectly machine the workpiece. But until machining actually occurs, the setup person cannot be sure that their initial compensation values are correct. After machining, they may find that another variation (like tool pressure) is causing the initial adjustment to be incorrect. Depending upon the tolerances for the surfaces being machined, a second adjustment may be required. After this second adjustment, machining will be more precise.

There is even a way to make an initial adjustment (prior to machining) that ensures excess material will remain on the machined surface after the first machining attempt (this technique is called trial machining). This guarantees that the workpiece will not be scrapped when the cutting tool machines for the first time—and is especially important for very tight (small) tolerances. With tight tolerances, even a small machining imperfection will cause a scrap workpiece.

Once the cutting tool has machined for the first time, the setup person will stop the cycle and measure the surface. If they have used the trial machining technique, there will be more material yet to remove. They will then make the appropriate adjustment and re-run the cutting tool. The second time the cutting tool machines, the surface will be within its tolerance band, probably right at the target dimension.

Tolerances

All dimensions have tolerances. You must program the mean value of the tolerance band for every coordinate you include in your programs. The mean value, of course, is right in the middle of the tolerance band.

Companies vary when it comes to how tight (small) the tolerances are that they machine on their CNC machining centers. Generally speaking, overall tolerances over about 0.010 inch (about 0.25 mm) are considered pretty open (easy to hold with today's CNC machining centers). Tolerances between 0.002 and 0.010 inch (0.050–0.25 mm) are common, and still not considered to be very tight. But tolerances under 0.002 inches can be more difficult to hold. And under 0.0005 inch (0.0013 mm)—which many companies do regularly hold on their CNC machining centers—can be quite challenging—especially when many workpieces must be produced.

The Initial Setting for Compensation

The setup person will do their best to assemble and measure certain cutting tool attributes (like length and diameter). They will then enter their measured values into the CNC (into something called tool offsets). But even if they perfectly measure and enter tooling values, and even if the programmer specifies the mean value for every tolerance in the program, there is no guarantee that every cutting tool will perfectly machine each dimension to the mean value of its tolerance band.

Tool pressure, which is the tendency for a cutting tool to deflect from the workpiece during machining, will always affect the way a cutting tool machines. It usually has the tendency of pushing the cutting tool away from the surface being machined. While the surface being machined should be close to its mean value (again, assuming the setup person perfectly measures and enters tooling information), it may not be perfectly at the mean value.

How tight is the tolerance? The tighter the tolerance, the more likely it will be that the deflection caused by tool pressure will cause the machined surface to be outside the tolerance band when the cutting tool machines for the first time. This could cause a scrap workpiece. And again, this is the reason why trial machining is required—to ensure that the first workpiece gets machined correctly.

And by the way, we've assumed that the setup person has perfectly measured and entered tooling information. Any mistakes will, of course, increase the potential for problems holding size on the first workpiece to be machined.

When is Trial Machining Required?

First let's talk about when trial machining is not required. Trial machining is not required for most roughing operations—like rough milling and rough boring. While a measurement should still be taken right after the roughing operation to confirm that the appropriate amount of finishing stock is being left for the finishing tool (and an adjustment must be made if not), trial machining is not necessary.

There are certain cutting tools and attributes that do not allow adjustments, and trial machining cannot be done for these tools. With drills, for example, you cannot control the diameter that the drill machines. Hole-diameter is based upon the diameter of the drill. The same goes for taps, reamers and counter-boring tools. So for machining operations that cannot be adjusted, you will not be trial machining.

Also, whenever tolerances are greater than about 0.003 inch (0.075 mm) or so, it is not unreasonable to expect the setup person (or tool-setter) to assemble and measure cutting tools accurately enough so that the cutting tool will machine within the tolerance band (even considering tool pressure) on the first try. While an adjustment may be necessary after machining to bring the dimension

precisely to the target value, trial machining should not be necessary. Examples include the depth of many (blind) holes and a certain milling operations.

As stated, trial machining is only required when machining tight tolerances. Though the actual cut-off point for when trial machining is required varies based upon the skill of the setup person and the operation being performed, it is not uncommon to trial machine for dimensions that have tolerances under about 0.002 inch (0.050 mm). This includes finish boring operations and many milling operations.

What Happens as Tools Begin to Wear?

So say the setup person uses trial machining techniques for critical dimensions and the first workpiece passes inspection. Now the production run begins. As cutting tools continue to machine workpieces, of course, they will begin to show signs of wear. Certain tools, like drills, will continue to machine properly for their entire lives without having an impact on the dimensions they machine (hole size and depth).

But as certain tools begin to wear, like milling cutters and boring bars, a small amount of material will wear away from the tool's cutting edge (a boring bar or an end mill will actually get a little smaller in diameter). Depending upon the tolerance band for the dimension being machined by the cutting tool, it may be necessary to make additional sizing adjustments during the cutting tool's life to ensure that the cutting tool continues to machine acceptable workpieces.

What do You Shoot For?

The value for each dimension that you are aiming for as you make adjustments is called the target value. In some companies, CNC setup people and operators are told to always target the mean value of the tolerance band (the same value the programmers includes in the program). But to prolong the time between needed adjustments, some companies ask their CNC people to target a value that is closer to the high or low limit of the tolerance band, whichever will prolong the time between adjustments. In any event, CNC people must know the target value for each dimension they must machine (this information should be in the production run documentation).

Why do Programmers have to know about Compensations?

Admittedly, much of this discussion is more related to setup and operation than it is to programming. But again, a programmer must understand enough about machine usage to be able to direct people that run the machine. And these discussions go to the heart of one of the most important reasons why compensation is required on CNC machining centers: to allow sizing adjustments without needing to modify the program.

Understanding Offsets

All three compensation types use offsets. Offsets are storage locations for values. They are very much like memories in an electronic calculator. With a calculator, if a value is needed several times during your calculations, you can store the value in one of the memories. When the value is needed, you simply type one or two keys and the value returns. In similar fashion, the setup person or operator can enter important tooling-related values into offsets. When they are needed by the program, a command within the program will invoke the value of the offset. And by the way, just as a calculator's memory value has no meaning to the calculator until it is invoked during a calculation, neither does a CNC offset have any meaning to the CNC control until it is invoked by a CNC program.

Like the memories of most calculators, offsets are designated with offset numbers. Offset number one may have a value of 6.5439. Offset number two may have a value of 6.2957. For cutting tool related offsets, offset numbers are made to correspond in some way to tool station numbers. For example, the tool length compensation value for the cutting tool placed in station number one is commonly entered in offset number one.

Unlike the memories of an electronic calculator that will be lost when the calculator's power is turned off, CNC offsets are more permanent. They will be retained even after the machine's power is turned off—and until an operator or setup person changes them.

Offsets are used with each compensation type to tell the control important information about tooling. From the marksman analogy, you can think of offset values as being like the amount of sight adjustment a marksman must make prior to firing a shot. Tooling related information entered into offsets includes each cutting tool's length, each milling cutter's radius, and program zero assignment values for work holding devices.

Offset Organization

Machining center controls vary with regard to how many offsets are available and how they are organized. First of all, rest assured that you will always have enough offsets to handle your applications. Most machine tool builders supply many more offsets than are required even in the most elaborate applications (an exception to this statement may sometimes be with regard to workpiece coordinate system setting offsets). Most machining centers have two distinct sets of offsets, one for cutting tools (tool offsets) and another for the workpiece coordinate system zero (work offsets).

Offsets Related to Cutting Tools

All cutting tools will require a tool length compensation offset. Milling cutters that perform side milling operations will additionally require a cutter radius compensation offset. Today’s CNCs also separate the tool offsets into geometry (the value measured at initial setup) and wear (the value determined by trial machining or tool wear). CNC manufacturers vary when it comes to how offsets are organized and displayed.

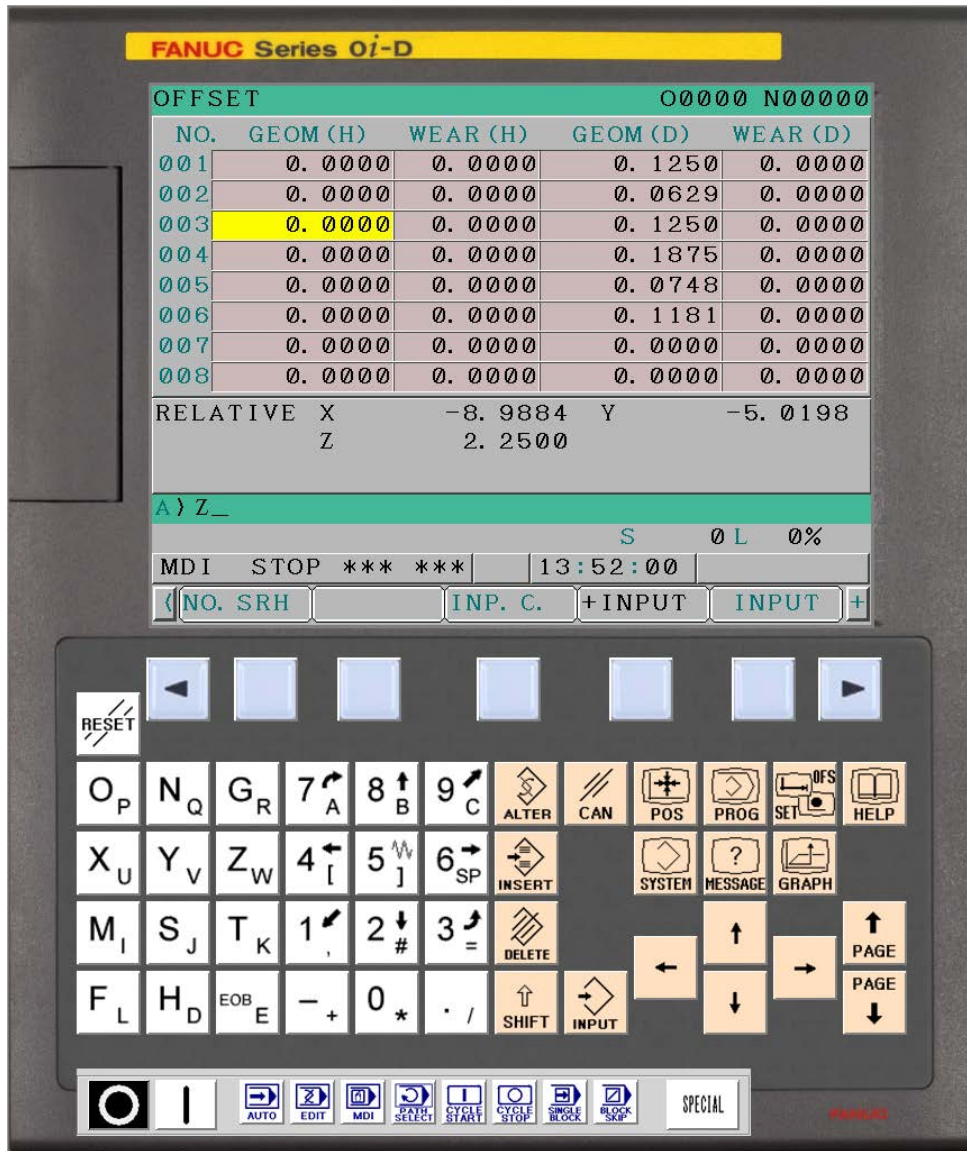


Figure 4.1: Display screen having four values per offset number

Today’s CNCs typically have four fields of data available for each tool pocket in the tool changer magazine. That means the tool length offset (H-word – example, H01) has the same number as the cutter radius offset (D-word, example, D01) and the tool number (T-word – example, T01). The tool length offset has two components, geometry and wear. The tool length geometry offset is the nominal length of the tool measured by the setup person. The tool length wear offset is used for fine tuning due to trial machining or tool wear. The cutter radius offset also has two components, geometry and wear. The tool wear geometry offset is the nominal radius of the cutter measured by the setup person. The cutter radius wear offset is used for fine tuning due to trial machining or tool wear. The CNC adds the respective geometry and wear offsets together to compensate for the tool length or the cutter radius.

Offsets Related to Program Zero

Workpiece coordinate system offset values are placed in work offsets. Each offset has at least three registers, one for X, one for Y, and one for Z. If the machining center has any rotary axes, there will also be a register for each rotary axis. The figure below shows the first page of workpiece coordinate system setting offset display screen for the FANUC 0i-MD with the 8.4-inch display.

WORK COORDINATES				00210 N00000			
[G54]							
NO.		DATA	NO.		DATA		
00	X	0.0000	02	X	0.0000		
EXT	Y	0.0000	G55	Y	0.0000		
	Z	0.0000		Z	0.0000		
01	X	0.0000	03	X	0.0000		
G54	Y	0.0000	G56	Y	0.0000		
	Z	0.0000		Z	0.0000		
A) _							
				S	0 L	0%	
MEM	****	***	***	11:27:30			
OFFSET	SETTING	WORK			{OPRT}	+	

Figure 4.3: Workpiece coordinate system offset display screen

Tool length compensation

Tool length compensation allows a programmer to ignore the precise length of each tool as a program is written. It is used for every tool in every program you write—so you must understand this important CNC feature.

The Reasons Why Tool Length Compensation is Needed

Cutting tools used on machining centers differ from one another. For one thing, there are a variety of cutting tool types that are used on machining centers, including center drills, spot drills, drills, taps, reamers, boring bars, end mills, and face mills (among many others). Each type of tool requires a different way of gripping the actual cutting tool in its holder. Some tools (like some straight shank tools) use a collet system. Others (like end mills) use a set-screw to hold the cutting tool in place. Yet others (like face mills and taps) require a very special style of tool holder—designed especially for the cutting tool.

No Two Tools Will Have Exactly the Same Length

Given the vast assortment of cutting tools available for use on CNC machining centers, it is unlikely that any two tools used in a program will be exactly the same length. And you will not know precisely how long each tool will be when you write the program. The figure below shows five different types of cutting tools to illustrate this point.

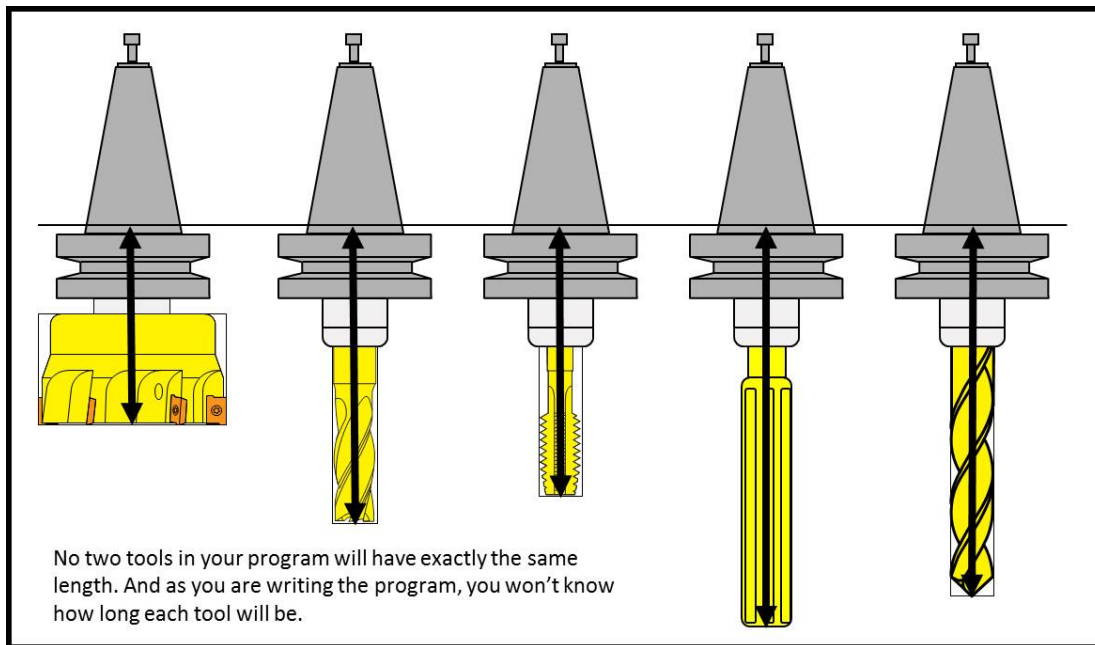


Figure 4.5: Five cutting tools that might be used in by a CNC machining center program

Tool length compensation will allow you to write programs even though you don't know how long the cutting tools will be at production time.

Tool's Length Will Vary Each Time it is Assembled

When a cutting tool is assembled more than once (even with the same components), its length will usually vary. Consider, for examples, straight shank tools that are placed in collet holders. Each time you assemble the tool, it will be of a different length. Tool length compensation will allow you to use the same program over and over again, even though each tool's length changes from one time the job is run to the next.

Tool Data is Entered Separately from the Program

The same program will work regardless of how long each cutting tool is. The program tells the control where to look for the length of each tool. During setup, the setup person (or someone) assembles and measures each cutting tool. The length of each tool is then placed in the appropriate location (a tool offset register).

Sizing and Trial Machining Must Often be Done

During a given tool's life a tool will wear and cause the surface being machined to change. Tool length compensation allows the setup person and operator to easily hold size for Z-axis related dimensions (pocket depths, hole-depths, etc.). The program need not be changed when workpiece dimensions must be adjusted.

Programming Tool Length Compensation

There are two popular methods for measuring and setting tool length compensation. As long as you use workpiece coordinate system offsets to assign program zero, the programming remains exactly the same regardless of which method you choose.

Tool length compensation is instated with a G43 word. Included within the G43 command is an H-word that specifies the offset number in which the tool length compensation value is stored. You must also include a Z-word in the G43 command, telling the machine where you want the tool tip to be positioned. This initial move allows the tool offset value to be instated without an unexpected Z-axis motion.

The G43 command will always be the cutting tool's first Z-axis motion. Said another way, you instate tool length compensation during each tool's first Z-axis motion as the tool approaches the workpiece in the Z-axis.

Once tool length compensation is instated, it remains in effect until the next tool is selected. All Z-axis motions you need the tool to make will be relative to the tool tip.

Since you will instate tool length compensation during the next tool's first Z-axis motion—using the appropriate offset of course—and since offsets are not accumulative—you need not cancel tool length compensation. G49 cancels tool length compensation, but if you the techniques shown in this course, you need not use G49 in your programs. Be warned, if you do program a G49 in a block by itself, it may cause the Z-axis to move by the tool offset amount to uninstall the tool offset.

Choosing the offset number to be used with each tool

Offsets are storage registers for values. Each offset to be used with tool length compensation will contain a tool length compensation value for one tool. To keep from entering the tool length compensation value in the wrong offset, the programmer must use a logical approach for selecting offset numbers.

Keep it simple. Use the offset number that corresponds to the cutting tool's magazine station number. For example, use offset number one for the tool in tool station number one. Use offset number two for the tool in tool station number two. And so on.

This tool length compensation offset will be instated in the program during each tool's first Z-axis motion. An H-word is used to specify the offset number. And again, we recommend that you make the H-word number match the tool station number (the T-word number for each tool).

An example program

The figure below shows the drawing to be used for this example.

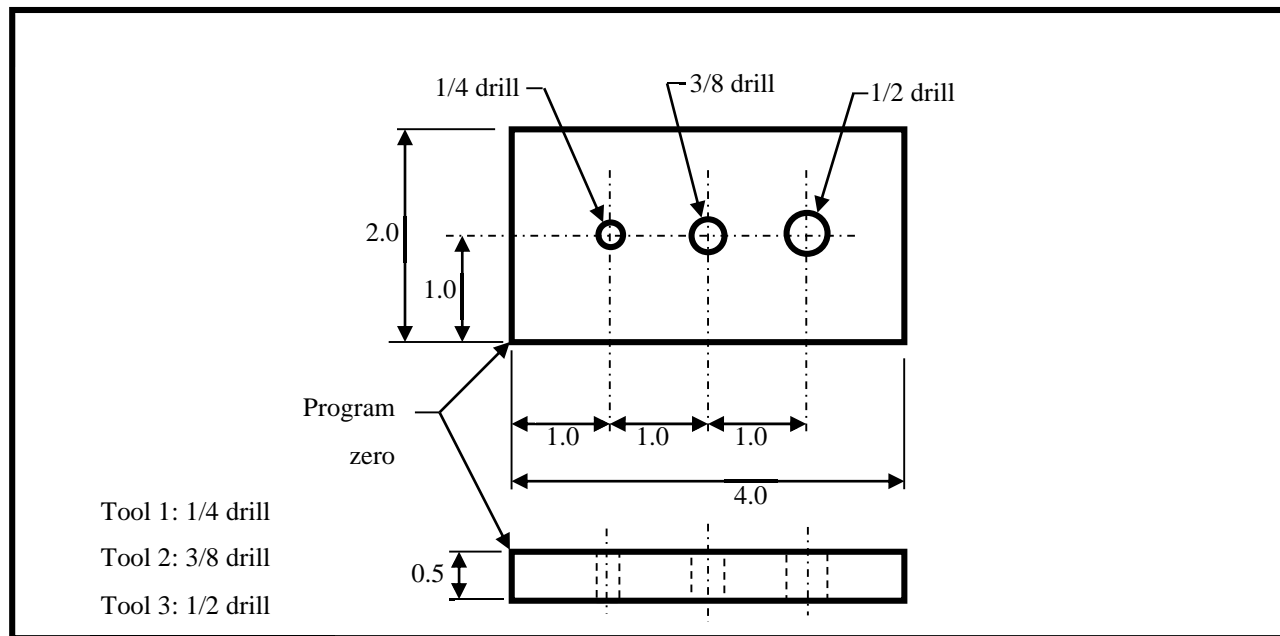


Figure 4.6: Drawing for example program

Program with comments:

```

O0003 (Program number)
N010 G20 G90 G54 (Select inch & absolute modes, workpiece coordinate system setting offset #1)
N020 T01 M06 (Load tool #1 in spindle – 1/4 drill)
N030 S1200 M03 T02 (Start spindle fwd at 1200 rpm, get tool #2 ready)
N040 G00 X1.0 Y1.0 (Rapid to hole location in X and Y)
N050 G43 H01 Z0.1 (Instate tool length compensation for tool one, approach in Z to just above work surface)
N060 M08 (Turn on the coolant)
N070 G01 Z-0.65 F4.0 (Drill hole)
N080 G00 Z0.1 M09 (Rapid out of hole, turn off coolant)
N090 G91 G28 Z0 M19 (Rapid to tool change position, orient spindle)
N100 M01 (Optional stop)

N110 T02 M06 (Load tool #2 in spindle – 3/8 drill)
N120 G54 G90 S1000 M03 T03 (Select workpiece coordinate system setting offset #1, absolute mode, start spindle fwd
at 1000 RPM, get tool number three ready)
N130 G00 X2.0 Y1.0 (Rapid to hole position in X and Y)
N140 G43 H02 Z0.1 (Instate tool length compensation for tool 2, approach in Z to just above work surface)

```

N150 M08 (Turn on coolant)

N160 G01 Z-0.7 F5.0 (Drill hole)

N170 G00 Z0.1 M09 (Rapid out of hole, turn off coolant)

N180 G91 G28 Z0 M19 (Rapid to tool change position, orient spindle)

N200 M01 (Optional stop)

N210 T03 M06 (Place tool number three in spindle)

(1/2 Drill)

N220 G54 G90 S800 M03 T01 (Select workpiece coordinate system setting offset #1, absolute mode, start spindle fwd at 800 RPM, get tool number one ready)

N230 G00 X3.0 Y1.0 (Rapid to hole in X and Y)

N240 G43 H03 Z0.1 (Instate tool length compensation for tool three, approach in Z to just above work surface)

N250 M08 (Turn on coolant)

N260 G01 Z-0.75 F6.0 (Drill hole)

N270 G00 Z0.1 M09 (Rapid out of hole, turn off coolant)

N280 G91 G28 Z0 M19 (Rapid to tool change position, orient spindle)

N290 M30 (End of program)

Blocks N050, N140, and N240 instate tool length compensation for each of the three tools. Notice that each of these commands is the first Z-axis movement for the tool (its approach movement in Z). Each instating command includes the G43 word, the appropriate H-word (that matches the tool station number that is currently in the spindle), and a Z-word. Once tool length compensation is instated, it will remain in effect until the next tool. Again, it never has to be canceled using this style of programming as the G28 automatically suspends the tool length compensation during the move to the zero return position.

Cutter radius compensation

Milling cutters (like end mills) can be used for side milling operations. Motions can be linear (straight-line) or circular. To this point, we have shown the side milling tool path based upon a milling cutter's centerline path. As you know, calculating coordinates for a milling cutter's centerline path requires that you consider the milling cutter's radius for every coordinate—which can make calculating coordinates quite difficult.

Reasons Why Cutter Radius Compensation is Required

Let's begin by discussing why you must master cutter radius compensation. Some of the reasons are quite similar to the ones given for tool length compensation.

Calculations are Simplified for Manual Programmers

When performing side milling operations without cutter radius compensation, you must specify coordinates for the milling cutter's centerline path. This requires that you consider the size of the milling cutter in every calculation—complicating each calculation. With cutter radius compensation, and as a manual programmer (not using a computer aided manufacturing [CAM] system), you will specify coordinates that are right on the work surface, ignoring the size of the milling cutter.

Range of Cutter Sizes

Just as tool length compensation allows the length of the cutting tool to vary without requiring a program change, so does cutter radius compensation allow the milling cutter's diameter (and of course, its radius) to vary without requiring a program change. This is valuable to manual and CAM system programmers, setup people, and operators (maybe more so to setup people and operators).

Do you use re-sharpened (re-ground) cutters?

Many companies re-sharpen their dull milling cutters and use them again. When a milling cutter is re-sharpened, its diameter will become smaller by about 0.020 inch or so (0.50 mm). The figure below shows some re-sharpened milling cutters. The milling cutter's tool path must, of course, be modified whenever a re-sharpened cutter is used—due to its smaller diameter. Again, cutter radius compensation will automatically modify the milling cutter's tool path based upon the current size of the re-sharpened cutter. If you do not use cutter radius compensation, you cannot use sharpened cutters without changing all of the cutter's motions in the program.

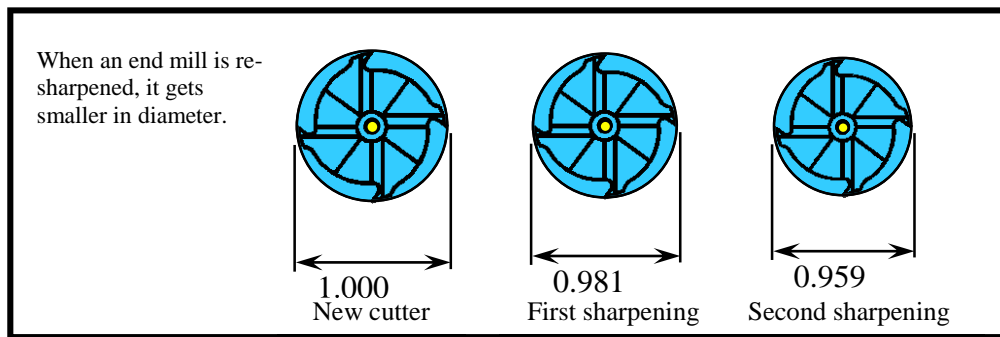


Figure 4.7: Many companies use re-sharpened cutters

Trial Machining and Sizing

Just as tool length compensation allows trial machining and sizing for depth surfaces, so does cutter radius compensation allow trial machining and sizing for XY surfaces (surfaces milled with the periphery of the milling cutter). Like depth surfaces, many XY surfaces have close tolerances. If you do not use cutter radius compensation, there will be no way to make sizing adjustments without actually changing the tool path coordinates in the program (which is usually difficult to the point of being infeasible—and we never recommend changing programmed coordinates to make sizing adjustments).

Also as with machining depth dimensions, tool pressure will affect how XY surfaces are milled. Again, just because a milling cutter's coordinates have been programmed perfectly (to the mean value for each tolerance)—and just because a milling cutter of exactly the intended size is being used—it is no guarantee that the milling cutter will machine the XY surface/s perfectly to size. The tighter the tolerance/s to be held, the more likely the milling cutter will not correctly machine the surfaces.

There is also a tooling-related problem that affects the accuracy of side milling operations. Milling cutters have a tendency to run-out in their holders. That is, the milling cutter will not be perfectly concentric with its tool holder. Run-out will cause the milling cutter to machine more material than it should—having the same effect as using a slightly larger milling cutter.

Milling cutters also have a tendency to wear during production. As the cutter wears, a small amount of material will be removed from its outside diameter. In effect, the milling cutter becomes smaller in diameter. With tight tolerances, this small amount of variance will cause problems. Sizing must be done during the tool's life to keep the milling cutter machining surfaces to size.

For these reasons, you must have the ability to perform trial machining and sizing with XY milling operations. Whether you are simply milling one straight surface (like the right end of the workpiece), or machining an elaborate contour, cutter radius compensation will allow you to do so.

Steps to Programming Cutter Radius Compensation

As with tool length compensation, there are two ways to use cutter radius compensation. But unlike tool length compensation, how programs are written is directly related to the method you choose. With the method we recommend, which is especially intended for manual programmers, you program the work surface path and the cutter radius compensation offset value is the milling cutter's radius. With the other method, preferred by many computer aided manufacturing (CAM) system programmers, you program the milling cutter's centerline path and the cutter radius compensation offset value is only the difference (in radius) between the planned milling cutter size and the actual size of the milling cutter being used.

Here are the three basic steps you must master in order to program cutter radius compensation:

1. Instate cutter radius compensation
2. Program the tool path to be machined
3. Cancel cutter radius compensation

Admittedly, if you are studying cutter radius compensation for the first time, this is going to get a little complicated. Cutter radius compensation tends to be the most difficult CNC feature to fully understand and master, but stick with it. Based upon the reasons just shown, it will be well worth the time it takes to master.

Step 1: Instate Cutter Radius Compensation

Machining centers allow a great deal of flexibility when it comes to instating and using cutter radius compensation. We're going to be showing a method that will work in almost all cases.

Instating cutter radius compensation involves at least three positioning commands that engage the milling cutter to the first surface to be milled. These motions include:

1. An XY motion to the approach position, called the prior position
2. One or more Z motions to the Z-axis work surface

3. A command instating cutter radius compensation that brings the milling cutter (in X and/or Y) to the first surface to mill. Let's discuss these commands one at a time.

The XY motion to the prior position

This positioning movement brings the center of the milling cutter to a position in X and Y that makes it possible to instate the cutter radius compensation. It is usually done at rapid—with the milling cutter is well clear of the workpiece. Indeed, this motion is usually done while the milling cutter is above the workpiece in the Z-axis.

We call the end-point for this motion the prior position because it is the XY position just prior to the instating command. The next XY motion will instate cutter radius compensation.

The prior position must be at least the milling cutter's radius (the value stored in the cutter radius compensation offset) away from the surface to mill. And it is important to choose a prior position that allows the milling cutter to form a ninety degree angle (right angle) with the first surface to be milled.

That is a lot to remember, so let's look at an example. The figure below shows the prior position.

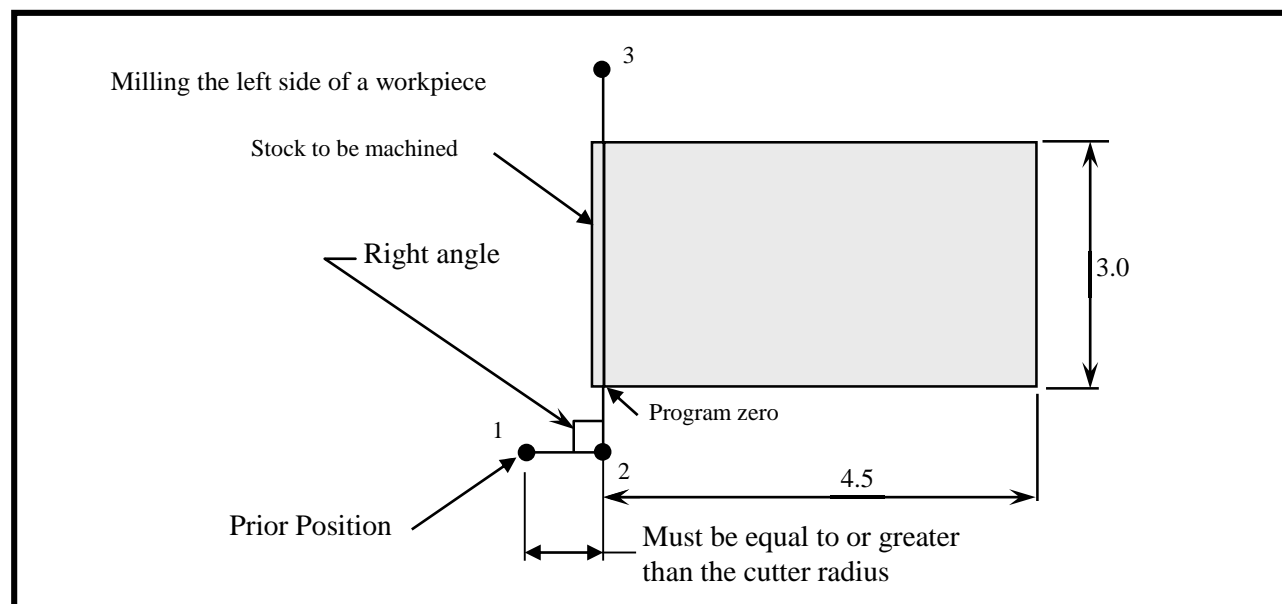


Figure 4.8: The prior position for cutter radius compensation

You program a rapid motion to the prior position with:

```
O0001 (Program number)
N010 G20 G90 G94 G54
N020 T01 M06 (1.0 end mill)
N030 S350 M03 T02 (( start spindle fwd at 350 rpm, and get tool station one ready)
N040 G00 X-0.5 Y-0.6 (Rapid to prior position)
```

When a 1.000 inch cutter moves to the position specified in block N040, it will be perfectly flush with the first surface to mill. While this will work as a prior position for a 1.000 end mill, it does not allow for any variation (like trial machining and sizing) during the production run. You have set the maximum cutter size to a 1.000 diameter end mill. If the setup person enters anything larger than 0.500 inch in the cutter radius compensation offset, the machine will generate an alarm. (The machine will think the cutter is already violating the first surface to mill.)

Let's revise the prior positioning movement to

```
N015 G00 X-0.6 Y-0.6
```

Now there is plenty of clearance between our planned 1.000 end mill and the first surface to machine. In fact, an end mill up to 1.200 inches in diameter can be used without generating an alarm. So, one way to determine maximum cutter size is to double the distance from the prior position to the first surface to mill. (Note that there are some other limiters for maximum cutter size. You'll see them as we continue.)

The Z motions to the Z-axis work surface

With the milling cutter at the prior position in the X and Y axes, it's time to move it to the Z-axis work surface. In most cases, when the milling cutter is at the prior position, it will be well clear of the workpiece in the X and Y axes, meaning you can rapid the tool to an approach distance above the Z-axis work surface. For our example, we'll say the workpiece is 1.00 thick. We'll want to move the tip of the end mill a little further below the bottom surface of the workpiece (say be 0.1-inch). We'll do so by first positioning the milling cutter just above the workpiece, then sending it to the Z-axis machining level.

```
O0001 (Program number)
N010 G20 G90 G94 G54
N020 T01 M06 (1.0 end mill)
N030 S350 M03 T02 (( start spindle fwd at 350 rpm, and get tool station one ready)
N040 G00 X-0.6 Y-0.6 (Rapid to prior position)
N050 G43 H01 Z0.1 (Instate tool length compensation)
N060 Z-1.1 M08 (End mill is now at Z-axis work surface and coolant is on)
```

Block N040 is the movement to the prior position in the X and Y axes. Block N050 instates tool length compensation and moves the end mill to just above the work surface. Block N060 moves it to the Z-axis work surface (still at rapid). Note that you could include Z-1.1 in block N050 instead of Z0.1 to bring the end mill directly to the Z-axis work surface. This is a matter of programming style.

Instate cutter radius compensation and position the cutting tool to the first surface to mill

At this point, the tool is positioned ready to instate cutter radius compensation. The instating command will include three things: 1: a G41 or G42 specifying the relationship between the milling cutter and the tool path, 2: a D-word specifying the offset number containing the cutter radius compensation value, 3: a motion to the first work surface to be machined.

G41 or G42?

One of two G-codes will be used in the instating command.

- G41—milling cutter is to the left of its path
- G42—milling cutter is to the right of its path

One way to determine which of G41 or G42 you must use is to look in the direction the milling cutter will be moving as it machines the work surface and ask “Which side of the work surface is the milling cutter on?” If the milling cutter is on the left side of the work surface, you will use G41 in the instating command. If the cutter is on the right side of the work surface, you will use G42. Selecting the G41 or G42 tells the CNC on which side to instate the value stored in the cutter radius offset. The figure below shows some examples.

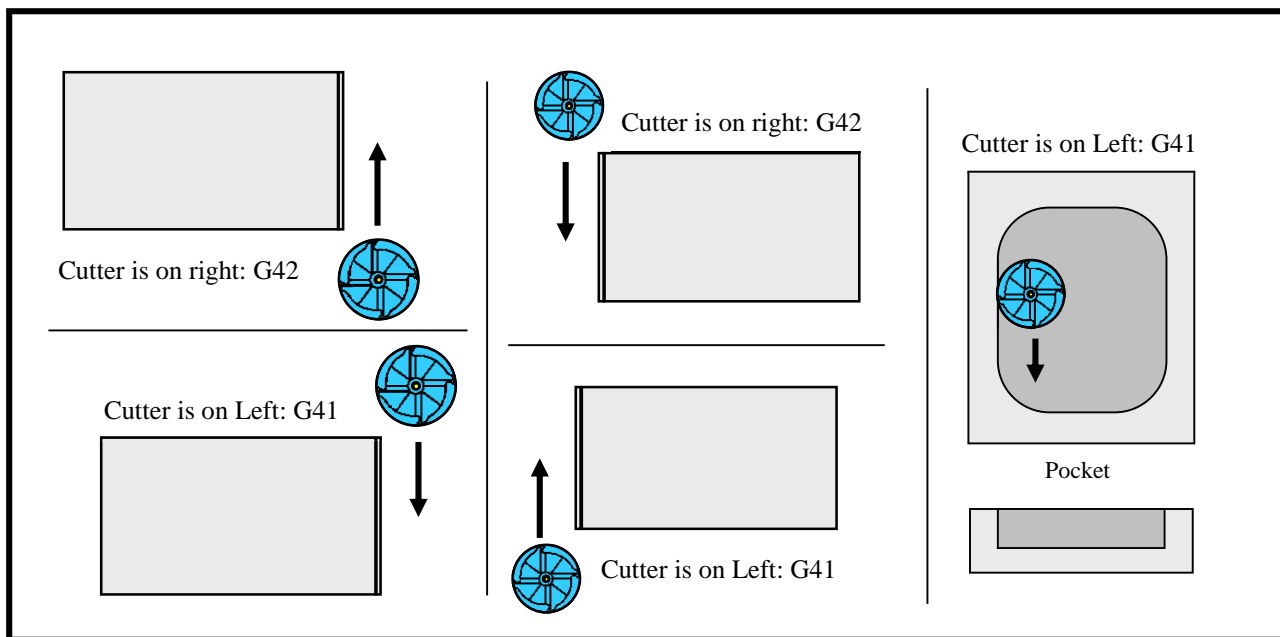


Figure 4.9: Deciding whether to use G41 or G42

The offset used with cutter radius compensation

The command that instates cutter radius compensation also requires that you specify the offset number in which the setup person will place the cutter radius compensation value. While instating, the machine will use this value to bring the cutter flush with the first surface to machine. It will also use this value to keep the cutter flush with all surfaces as the contour is being machined.

In the program, a D-word is used to specify the offset number used with cutter radius compensation. The D-word must be included in the instating command.

Machining centers vary when it comes to offset organization. Most of today's CNCs have two or four register fields per offset allowing tool length compensation offset (H-word) and the cutter radius compensation offset (D-word) to have the same number. However, older CNCs may only have one register per offset, they must be shared between tool length and cutter radius compensation. If you use the offset number that corresponds to the tool station number in which to store the tool length compensation value, it cannot be used again for cutter radius compensation—you must choose another.

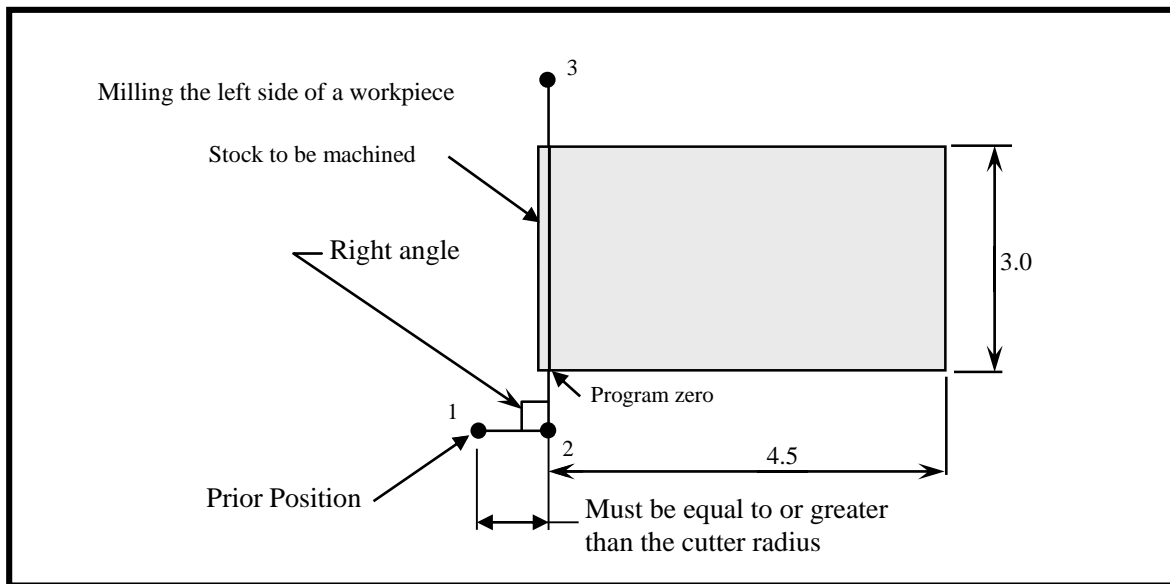
For CNCs that have only one register per offset, we recommend adding a constant number to the tool station number to come up with the offset number to use with cutter radius compensation. This constant number must be equal to or larger than the number of cutting tools the machine's automatic tool changer can hold.

For example, if your machining center's automatic tool changer magazine can hold thirty tools, add thirty to the tool station number to come up with the cutter radius compensation offset number. For this machine, if tool number 5 is a milling cutter, you will use offset number 5 in which to store its tool length compensation value (H5) and offset number 35 in which to store its cutter radius compensation value (D35).

If you have a machining center that has two or four registers per offset, choosing the cutter radius compensation offset number is much easier. Simply make it correspond to the milling cutter's tool station number. For this kind of machine, if station 5 is a milling cutter (T5 or T05), you will use offset number 5 in which to store its tool length compensation value (H5 or H05) and its cutter radius compensation value (D5 or D05).

The motion to the first work surface

Also included in the instating command is a motion that brings the milling cutter to the first work surface to be machined. With this motion, you are no longer programming the milling cutter's centerline position. The end point for this position is right on the work surface.



The instating command moves the milling cutter from the prior position (point number 1) to the first work surface to machine (point number 2). Again, notice that point number 2 is right on the work surface. Also, the instating command can be done at rapid (G00) if the milling cutter clears the workpiece during the motion—or in a straight line motion (G01) with a feedrate if it does not. You are not allowed to instate cutter radius compensation during a circular motion command (G02 or G03).

Since points one and two in our example are well clear of the workpiece in the Y-axis (with our 1.00 planned cutter size), we'll make the instating command at rapid. Here are the commands so far:

```
O0001 (Program number)
N010 G20 G90 G94 G54
N020 T01 M06 (1.0 end mill)
```

N030 S350 M03 T02 ((start spindle fwd at 350 rpm, and get tool station one ready)

N040 G00 X-0.6 Y-0.6 (Rapid to prior position)

N050 G43 H01 Z0.1 (Instate tool length compensation)

N060 Z-1.1 M08 (End mill is now at Z-axis work surface and coolant is on)

N070 G41 D01 X0.0 (Instate cutter radius compensation)

Step 2: Program the Tool Path to be Machined

Once cutter radius compensation is instated, you must program the work surface path for the contour to be machined. It is usually important that you end the contour with the milling cutter well clear of the last machined surface. If you leave the milling cutter in contact with the last milled surface, it's likely that the milling cutter will leave a witness mark on the surface when it is retracted in the Z-axis.

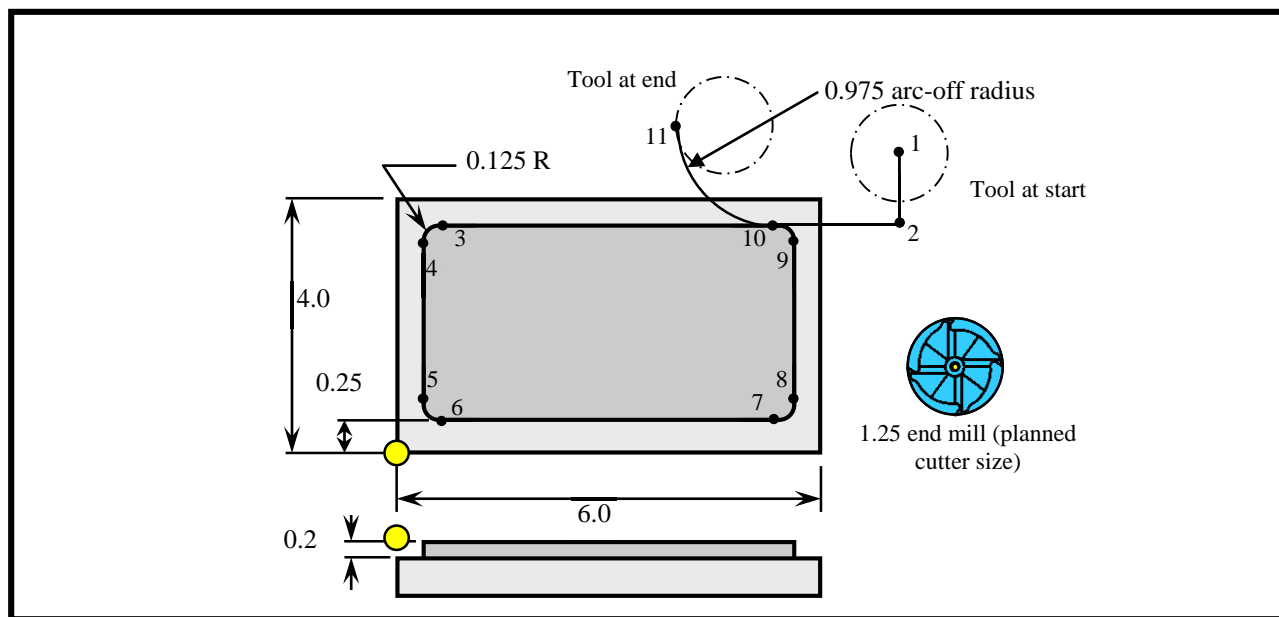


Figure 4.10: More elaborate example of a work surface tool path using cutter radius compensation

The work surface tool path takes the milling cutter from point 2 to point 3 (linear interpolation motion)—from point 3 to point 4 (circular interpolation motion)—from point 4 to point 5 (linear interpolation motion)—and so on through point number 10. Remember that since you are programming the work surface tool path, the radius of all circular motions (R-word) will be the actual workpiece radius, R0.125 for each circular motion in our example.

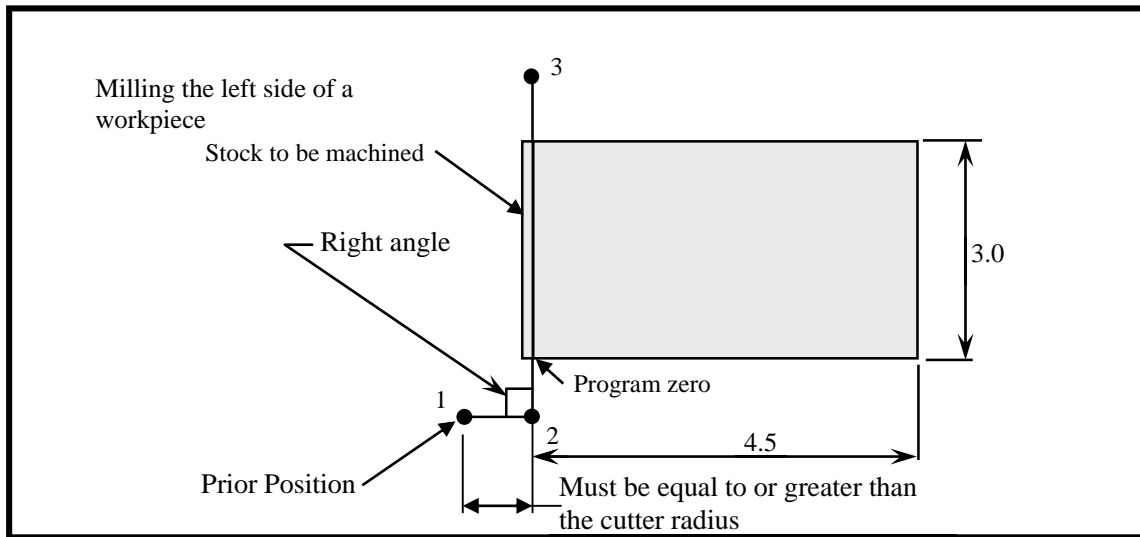
When the milling cutter reaches point number 10, it is finished milling the contour, but if you allow it to retract in Z at point 10, it will leave a nasty witness mark on the workpiece (a gouge-line on the side of the milled surface). Most programmers will come up with one more work surface tool path motion to bring the milling cutter away from the last surface to mill. This is best done in a circular motion (commonly called an arc-out motion). The size of this arc must, of course, be larger than the radius of the largest planned cutter. It must also include enough clearance to allow the largest planned cutter to move away from the last surface being milled.

For our example, when using the planned cutter size (1.25 inch in diameter), the 0.975 arc-out radius will provide 0.350 inch motion away from the last surface milled (0.975 arc-out radius minus 0.625 milling cutter radius). This distance provides clearance for the 0.25 inch step around the contour. When the cutter reaches point number 11, it will be completely clear of the workpiece (by 0.1 inch) when using a 1.25 diameter cutter.

Step 3: Cancel Cutter Radius Compensation

Like many CNC features, cutter radius compensation is modal. The machine will continue to keep the milling cutter on the right or left side of all programmed motions until cutter radius compensation is cancelled. A G40 word is used to cancel cutter radius compensation. The very next X and/or Y coordinate (either within the G40 command or after it) will be taken as a centerline position.

Though it is not always possible to do so, we recommend retracting the milling cutter in the Z-axis prior to canceling cutter radius compensation. This way, the cutter will be clear of the workpiece should an unexpected X and/or Y movement take place during cancellation.



Here is the entire milling operation, including the cancellation of cutter radius compensation:

```

O0001 (Program number)
N010 G20 G90 G94 G54
N020 T01 M06 (1.0 end mill)
N030 S350 M03 T02 (( start spindle fwd at 350 rpm, and get tool station one ready)
N040 G00 X-0.6 Y-0.6 (Rapid to prior position)
N050 G43 H01 Z0.1 (Instate tool length compensation)
N060 Z-1.1 M08 (End mill is now at Z-axis work surface and coolant is on)
N070 G41 D01 X0.0 (Instate cutter radius compensation)
N080 G01 Y3.6 F5.0 (Mill left side)
N090 G00 Z.01 M09 (Retract cutter above work surface, turn off coolant)
N100 G40 (Cancel cutter radius compensation)
N100 G91 G28 Z0 M19 (Move to tool change position, orient spindle)
N110 M01 (Optional stop)

```

.
.

.

In block N090, we retract the tool to above the workpiece—then in block N100, we cancel cutter radius compensation. From this point on, any XY position specified in the program will be to the spindle centerline.

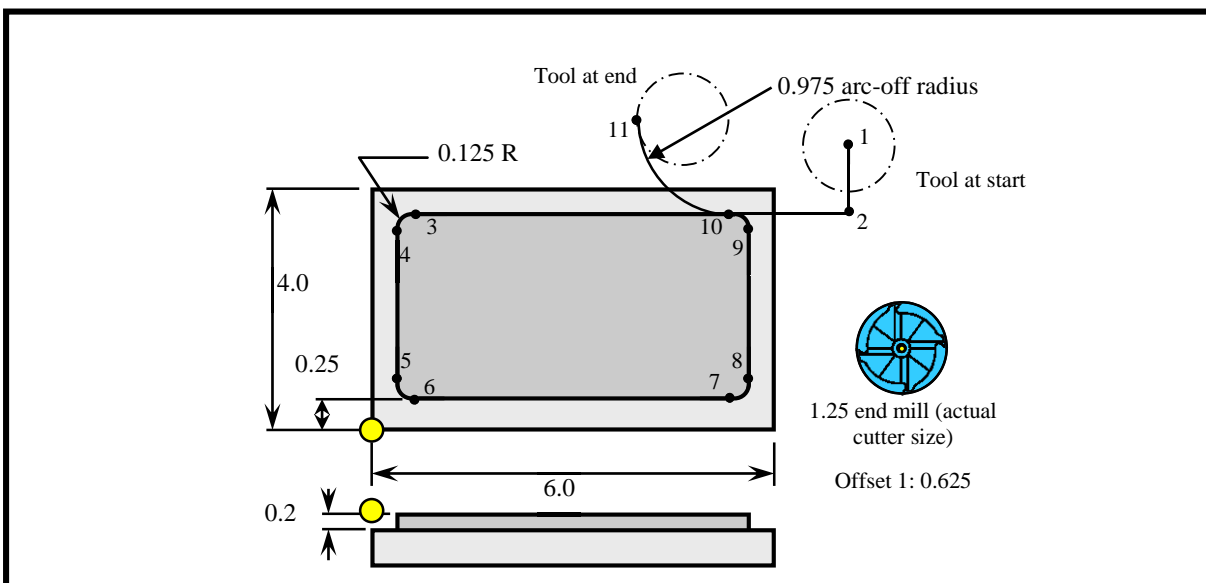
Example

Figure 4.11: Complete example 1 using cutter radius compensation

The milling cutter will be milling the 0.25 inch step around the outside of this workpiece. The planned cutter size is a 1.25 diameter end mill (having a 0.625 radius). Based upon the distance from point one to point two in the program, the maximum cutter size for this program is 1.45 inches in diameter.

Program with comments:

```

O0026 (Program number)
N010 G20 G90 G94 G54
(1.25 end mill in station 1)
N020 S400 M03 (Start spindle fwd at 400 rpm)
N030 G00 X6.725 Y4.475 (Rapid to prior position pt 1)
N040 G43 H01 Z0.1 (Instate tool length compensation)
N050 G01 Z-0.2 F40.0 (Fast feed to work surface)
N060 G42 D01 Y3.75 (Instate cutter radius compensation to pt 2)
N070 X0.375 F5.0 (Mill to point 3)
N080 G03 X.25 Y3.625 R.125 (Circular mill to point 4)
N090 G01 Y0.375 (Mill to point 5)
N100 G03 X.375 Y0.25 R.125 (Circular mill to point 6)
N110 G01 X5.625 (Mill to point 7)
N120 G03 X5.75 Y.375 R.125 (Circular mill to point 8)
N130 G01 Y3.625 (Mill to point 9)
N140 G03 X5.625 Y3.75 R.125 (Circular mill to point 10)
N150 G02 X4.65 Y4.725 R.975 (Arc-off the work surface to point 11)
N160 G00 Z0.1 (Retract to above the work surface in Z)
N170 G40 (Cancel cutter radius compensation)
N180 G91 G28 Z0.0 (Rapid to Z-axis zero return position)
N190 G28 X0.0 Y0.0 (Rapid XY to zero return position)
N200 M30 (End of program)

```

Since the cutter is 0.725 away from the right side of the workpiece in block N030, we're allowing the fast-feed approach in line N050 (at 40.0 ipm). In block N070, when milling begins, we include the cutting feedrate.

Geometry offsets (turning centers)

Geometry offsets are used to assign program zero, which is the operator's responsibility (not the programmer), so we minimize our discussion here.

How geometry offsets work

The control must be told the distances in X and Z from the cutting tool tip while the machine is resting at its zero return position to the program zero point. We call these distances the *program zero assignment values*. There are several ways to actually determine and enter these values.

While it may not be of the utmost importance, you may find it helpful to understand what actually happens when geometry offsets are used. (Of course, what is of utmost importance is that you know how to use geometry offsets.)

You know that each cutting tool will have a different set of program zero assignment values. When geometry offsets are used, program zero assignment values (one set per tool) are placed in each tool's geometry offset. The geometry offset number will match the tool station number, so the program zero assignment values for tool number six will be placed in geometry offset number six.

You can get a better understanding of how geometry offsets actually work by looking at the absolute position display page (*not* the relative page) right after power up and before a program is executed. While monitoring this page, send the machine to its zero return position in the X and Z axes.

With the machine is resting at its zero return position right after powering up, most machines will set the absolute position display page to X00.0000 and Z00.0000. Think about this. The absolute position display page always shows the machine's position relative to the program zero point. At this point in time, the control actually thinks the zero return position is the program zero point!

Whenever a geometry offset is instated (with the first two digits of the T-word) the distances from the tool tip at the zero return position to the program zero point will be transferred to the absolute position display screen registers. If the machine is not currently at the zero return position in one axis of the other, it will even take this into consideration when it sets the absolute position display values.

For example, say the current X and Z registers of geometry offset number one are set to:

X-12.2437 Z-11.8476

The machine is currently at the zero return position when the T-word T0101 is executed. At this point the absolute position display registers will show X12.2437 and Z11.8476 (again *if* the machine is currently at the zero return position in X and Z – and if the work shift offset is currently zero). The machine now knows how far it is from the tool tip to the program zero point, and can correctly make the motions commanded by the program to machine the workpiece.

Again, if the machine is not at the zero return position when a geometry offset is instated, it will consider this when it sets the absolute position display. For example, say the machine is precisely one inch in each axis from the zero return position (closer to the workpiece – on the negative side of the zero return position). In this case, when the T-word T0101 is executed, the absolute position displays will show X11.2437 and Z10.8476. The machine has correctly compensated for the turret's position, and still knows the correct distance between the program zero point and the tool tip.

This is why the machine's starting position is not critical when geometry offsets are used. The turret can be in any location and the machine will still correctly determine the cutting tool's position – and set the absolute position displays accordingly.

The total program zero assignment value

As you now know, the absolute position displays will be correctly updated whenever a T-word is executed. And the machine will take into consideration the geometry offset value for the tool, the work shift value (for the Z-axis), and the machine's current position. But there's one more thing that affects the values being placed in the absolute position displays when a T-word is executed – the *wear offset values* for the tool.

A wear offset will also be instated when a T-word is executed. The machine will total the geometry offset, the wear offset, the work shift value, and the machine's current position relative to the zero return position. It will use the result (in each axis) as the total offset for the tool. *These are the true program zero assignment values for the tool* – they are the distances from the program zero point to the tool tip (in each axis) at the very moment the T-word is executed. It will place these values on the absolute position display page. Figure 4.4 shows an example.

Figure 4.4 – How total offset is calculated

Admittedly, you don't have to know how geometry offsets work to use them. But it's important to know that nothing magical is happening. Additionally, this presentation should help you understand the function of each value involved with program zero assignment.

Wear offsets

Like geometry offsets, wear offsets are more related to CNC operators.

You know that the tolerances commonly held on CNC turning centers are quite small. It is not unusual to see at least one overall tolerance of under 0.001 inch (0.254 mm) on turned workpieces.

You also know that each cutting tool has its own program zero assignment – and that there are several ways to assign program zero. And you know that unless you are using a properly calibrated tool touch-off probe, mistakes with program zero assignment – even minor ones – as well as the effects of tool pressure, make it difficult to *perfectly* assign program zero. That is, even after program zero is assigned, there is no guarantee that every tool will machine the workpiece perfectly – or even within specified tolerances. The tighter the tolerances you must hold, the greater the potential there will be for sizing problems.

Wear offsets provide a way to make minor adjustments when machined surfaces are not within their tolerance bands – or when they're close to a tolerance limit.

There are at least four times when a typical CNC setup person or operator will use wear offsets:

- **During setup and after mounting a cutting tool in the turret** - After machining with the new tool, if the setup person discovers that the cutting tool has not machined a surface within the tolerance band, or if the surface is close to a tolerance limit, they can change a wear offset to make the needed adjustment.
- **When trial machining** – Trial machining is done when the setup person or operator is worried that a cutting tool (that has just been placed in the turret) will not machine within the tolerance band. Wear offsets are commonly used with which to make trial machining adjustments (Remember, if you are using a properly calibrated tool touch-off probe, you shouldn't need to trial machine.)
- **When compensating for tool wear** – As a cutting tool wears, it will cause the surfaces it machines to grow or shrink in size. Wear offsets are used to keep cutting tools machining on-size for their entire lives.

-After a dull tool is replaced – Again, during a cutting tool's life, tool wear commonly causes the need for sizing adjustments in wear offsets. When a dull tool is replaced with a new one, the wear offset must be set back to its initial value – otherwise the new tool will machine too much material from the workpiece.

Every dimension has a tolerance. You know that each tolerance will have a high limit (largest acceptable dimension), a low limit (smallest acceptable dimension), and a mean value (the dimension right in the middle of the tolerance band).

You also know that each dimension to be machined on a workpiece will have a *target value* – this is the dimension you shoot for when an adjustment must be made. Many CNC people use the mean value of the tolerance band as the target value. That is, when an adjustment must be made, they target the mean value. While there are times when this may not be appropriate (large lot sizes with small tolerances). That is, for each surface needing an adjustment, we'll be targeting the mean value.

You know that the *deviation* is the amount of needed adjustment. It is the difference between the measured value and the target value. In all cases, there will be a polarity to the deviation (plus or minus). The current wear offset value must be either increased or decreased by the amount of the deviation. The polarity is determined by judging which way the cutting tool must move (plus or minus) in order to bring the dimension back to the target value.

Which dimension do you choose for sizing?

It is very common for a finishing tool to machine several surfaces. One finish turning tool may, for example, finish three external diameters and faces. Each surface may have its own tolerance specification. And of course, you program the mean value for each tolerance. In almost all cases, only one wear offset will be used to control *all* surfaces machined by each tool, meaning one adjustment will handle all surfaces machined by the tool.

For example, say a finish turning tool machines three external diameters: a 1.5 inch diameter, a 2.0 inch diameter, and a 3.0 inch diameter. Unless there is a substantial tool pressure problem, when the 1.5 inch diameter is coming out correctly (based on a wear offset adjustment in X), so will the 2.0 and 3.0 inch diameters.

When it comes to making sizing adjustments, you should use the two surfaces (one for diameters and one for lengths) that have the smallest tolerances. Use them to make sizing adjustment decisions.

How wear offsets are programmed

As you know, wear offsets are invoked by the second two digits of the T-word. The command

```
N140 T0303
```

for example, indexes the turret to station number three, instates geometry offset number three (the first two digits) and instates wear offset number three (the second two digits). In almost all applications, you'll be making the wear offset number the same as the tool station number (tool one: wear offset one, tool two: wear offset two, and so on).

What actually happens when a wear offset is instated will vary based upon the CNC control (model and age). With newer controls (and especially if geometry offsets are used to assign program zero), the T-word appears to simply index the turret. The geometry and wear offsets will not actually be instated (made active) *until the next motion command*. But with some older controls – and especially when program zero must be assigned in the program – the T-word indexes the turret and the wear offset will be immediately instated. This will cause the turret to actually move by the amount of the wear offset.

Tool nose radius compensation

Earlier, we mention one important time when you must compensate for attributes of cutting tools. It has to do with the small radius that is on the cutting edge of any single point cutting tool – like a turning tool or a boring bar. Figure 4.12 shows this radius.

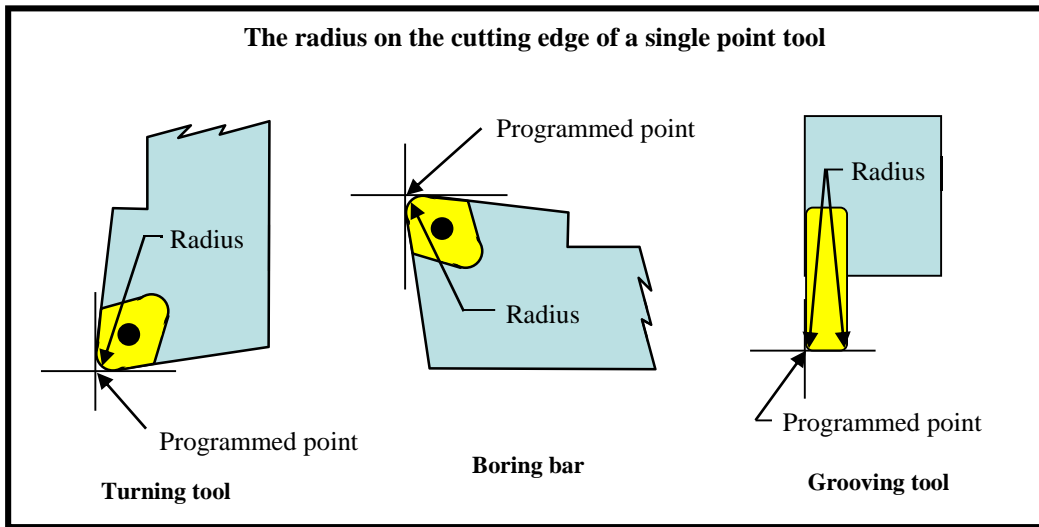


Figure 4.12 – All single point tools have a small radius on the cutting edge

For cutting tools used in the United States, the actual size of the radius will be specified in inches – and there are four standard tool nose radius sizes for turning and boring inserts:

- 1/64 inch (0.0156)
- 1/32 inch (0.0316)
- 3/64 inch (0.0468)
- 1/16 inch (0.0625)

Though you may consider these radii to be quite small, this small nose radius on the edge of the cutting tool will be sufficient to cause a small deviation from the programmed shape of your workpiece – at least when angular and circular surfaces must be machined.

Remember that you are programming the extreme edges of the cutting tool in each axis. This is illustrated in Figure 4.12 (specified as *programmed point*). Notice the small gap between the programmed point and the actual cutting edge.

This small gap will not affect the turning of diameters (parallel to the Z-axis) and the machining of faces (parallel to the X-axis). Figure 4.13 shows this.

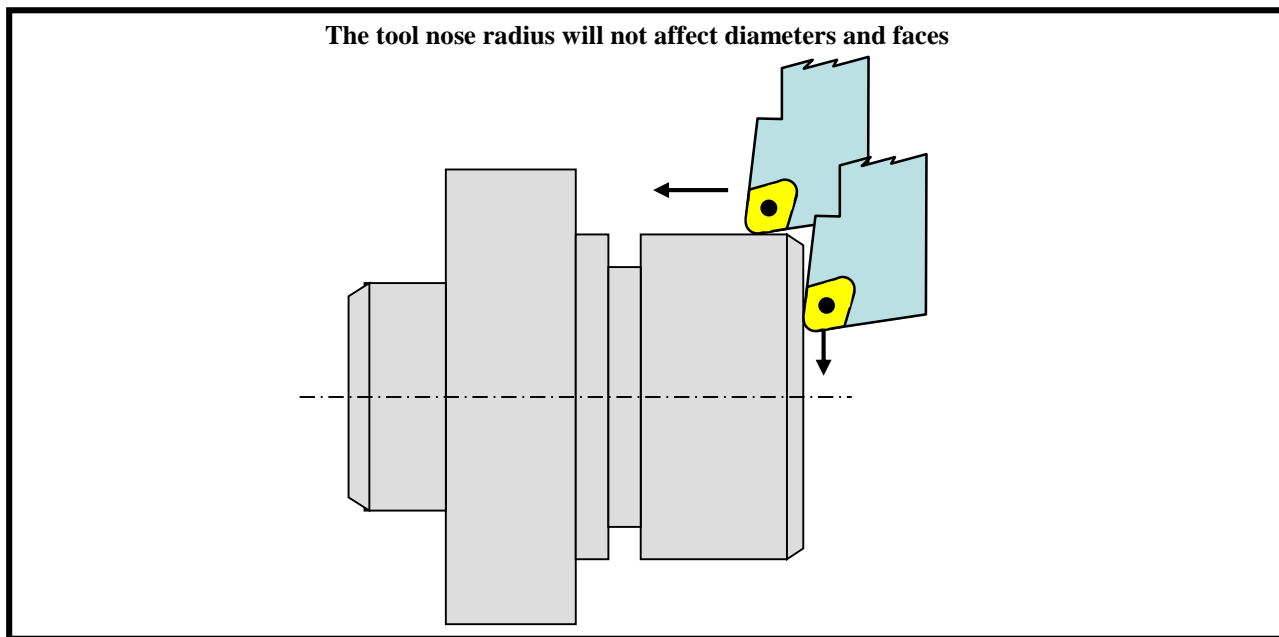


Figure 4.13 – The tool nose radius will not affect the machining of straight turns and faces

When a cutting tool is turning diameters and machining faces, the point programmed and cutting edge are in direct contact with the workpiece surface being machined.

But when angular (tapered) surfaces and circular surfaces must be machined, the gap between the programmed point and the cutting edge will affect machining – as Figure 4.14 shows.

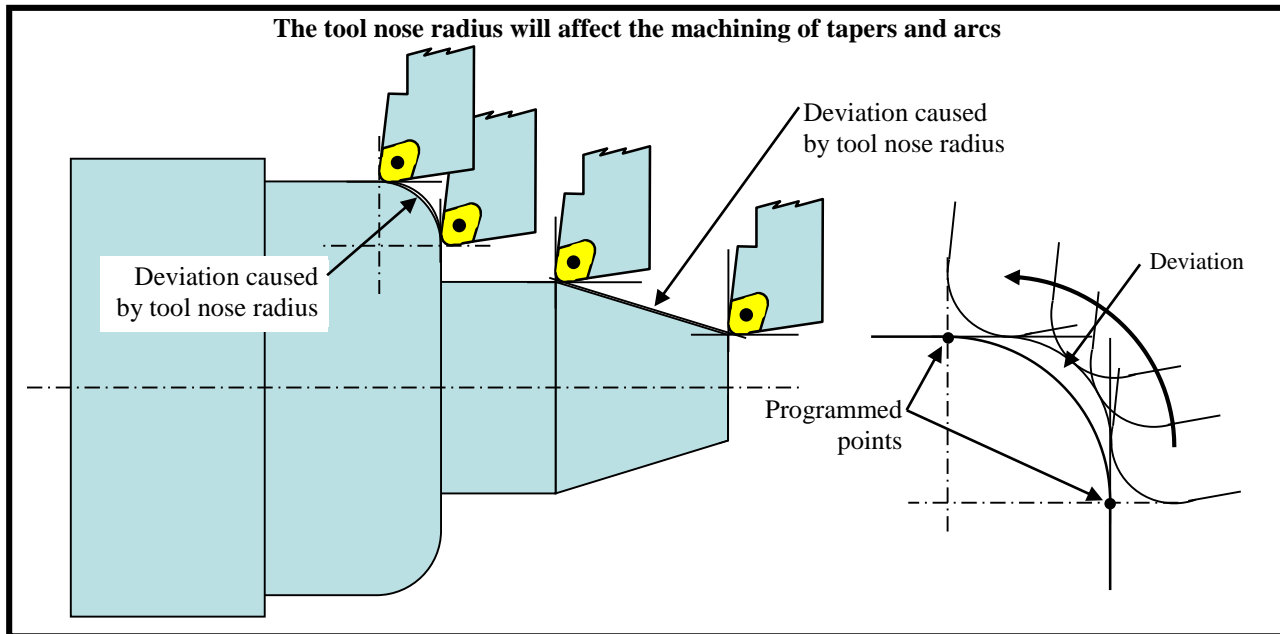


Figure 4.14 – Tool nose radius affects the machining of angular and circular surfaces

Steps to programming tool nose radius compensation

Programming tool nose radius compensation is relatively easy. Here are the three programming steps:

- Instate tool nose radius compensation
- Program the motions to machine the workpiece
- Cancel tool nose radius compensation

Instating tool nose radius compensation

To instate tool nose radius compensation, you must first be able to determine how the tool will be related to the work surface during the machining operation. Look in the direction the tool will be moving during the operation (rotate the print if it's necessary). Looking in this direction, ask the question, "What side of the work surface is the cutting tool on, the left side or the right side?" If the cutting tool is on the left side of the surface to be machined, you will use a G41 to instate tool nose radius compensation. If the cutting tool is on the right side of the surface to be machined, you will use a G42 to instate tool nose radius compensation. Look at Figure 4.17 to see some examples of how this is done.

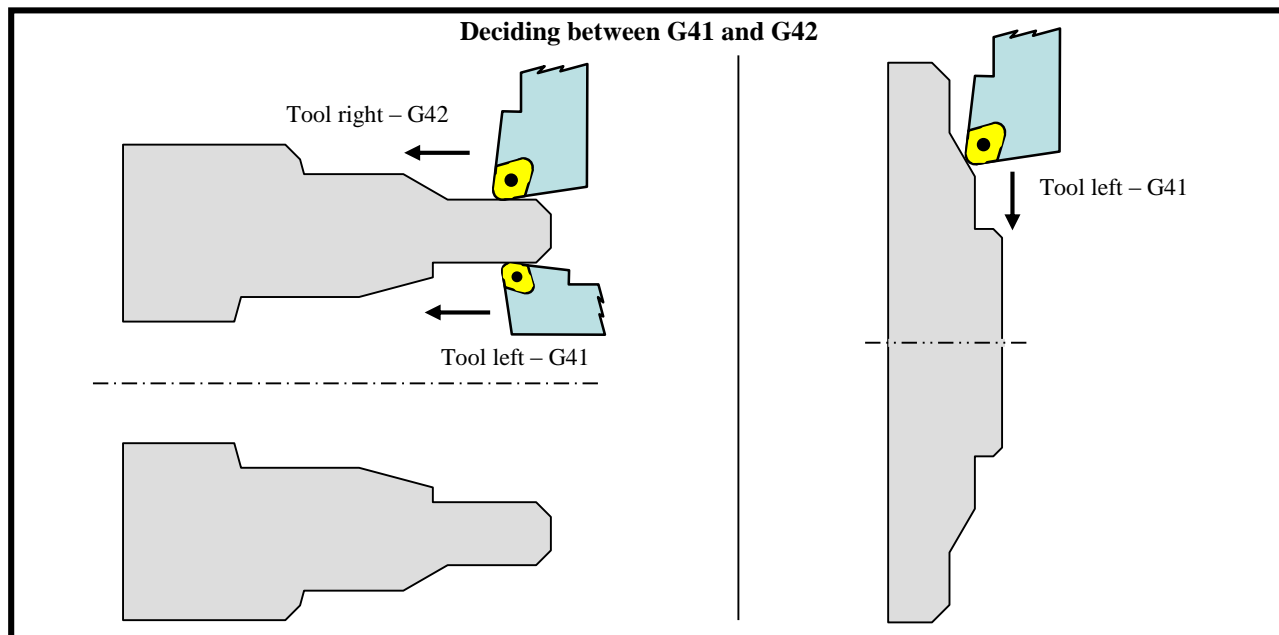


Figure 4.17 – Drawing illustrates how to decide between G41 and G42

Notice that G42 is always used for turning toward the chuck and G41 is always used for facing (toward the spindle center) and boring operations.

Once you know which side of the surface to be machined the tool is on (left or right), you simply include the appropriate G-code (G41 or G42) in the cutting tool's approach movement.

Programming motion commands to machine the workpiece

Once you've instated tool nose radius compensation, you simply program the movements to finish turn or bore the workpiece using work surface coordinates. Again, these programmed coordinates are right on the work surface. The control will automatically keep the cutting edge radius on the specified side of the workpiece (left or right), and tangent to surfaces being machined.

Once tool nose radius compensation is instated, the machine will simply keep the tool on the left or right side of all programmed surfaces until tool nose radius compensation is canceled.

Canceling tool nose radius compensation

You must remember to cancel tool nose radius compensation. If it is not canceled, the machine will remain under its influence even with subsequent tools.

Canceling tool nose radius compensation is easy. Just include a G40 word in the command that sends the tool back to the tool change position.

An example program

Figure 4.19 shows the workpiece to be used for our first example program. Again, we're only finishing this workpiece – the roughing has already been done by another tool.

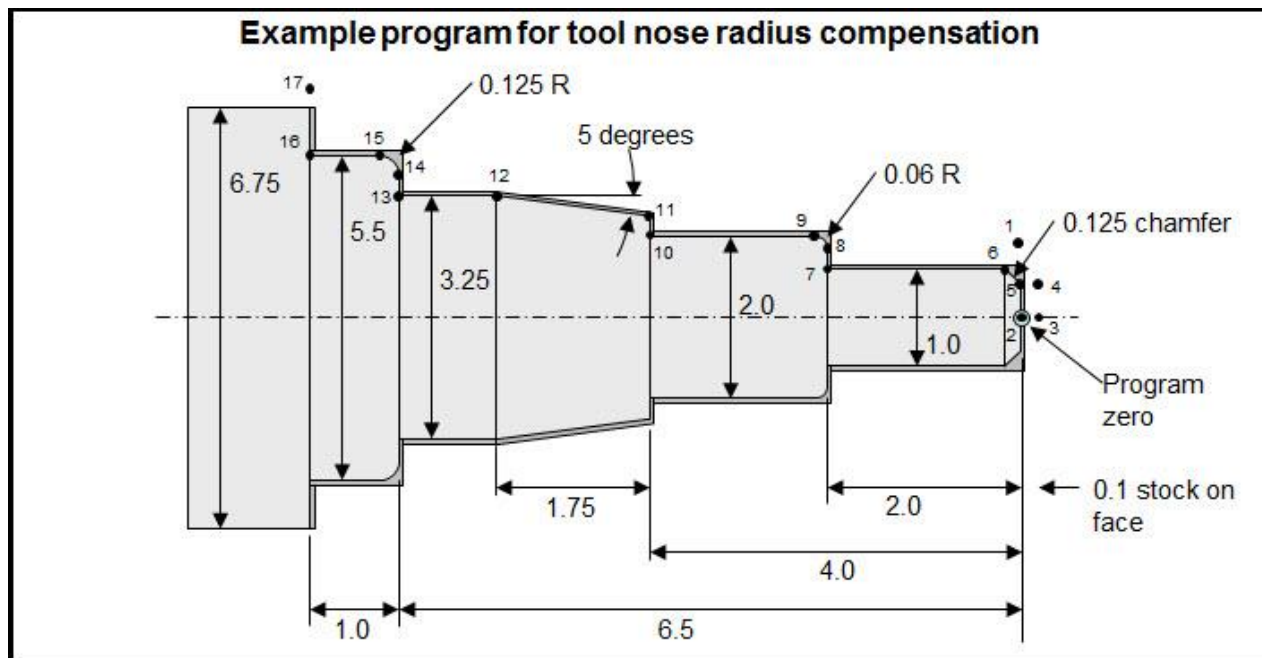


Figure 4.19 – Drawing for tool nose radius compensation example program

Here is the program.

```

O0001
N005 T0202 M42 (Index turret, select high spindle range)
N010 G96 S500 M03 (Start spindle fwd at 500 sfm)
N015 G00 X1.2 Z0 M08 (Rapid to point 1, turn on coolant)
N020 G01 X-0.062 F0.007 (Face to point 2)
N025 G00 Z0.1 (Rapid away to point 3)
N030 G42 X0.75 (Instate tool nose radius compensation, rapid up to point 4)
N035 G01 Z0 (Feed to point 5)
N040 X1.0 Z-0.125 (Chamfer to point 6)
N045 Z-2.0 (Turn 1.0 diameter to point 7)
N050 X1.875 (Feed up face to point 8)
N055 G03 X2.0 Z-2.0625 R0.0625 (Turn radius to point 9)
N060 G01 Z-4.0 (Turn 2.0 diameter to point 10)
N065 X2.9438 (Feed up face to point 11)
N070 X3.25 Z-5.75 (Turn taper to point 12)
N075 Z-6.5 (Turn 3.25 diameter to point 13)
N080 X5.25 (Feed up face to point 14)
N085 G03 X5.5 Z-6.625 R0.125 (Turn radius to point 15)
N090 G01 Z-7.5 (Turn 5.5 diameter to point 16)
N095 X6.95 (Feed up face to point 17)
N100 G00 G40 X8.0 Z6.0 (Rapid to tool change position, cancel tool nose radius compensation)
N105 M30 (End of program)

```

Key Concept 5: Provide structure to your CNC programs

CNC machines have come a long way. In the early days of NC (before computers), a program had to be written just so. If anything was out-of-place, the machine would generate an alarm—failing to execute the program. While today's CNC machines are much more forgiving, you must still write CNC programs in a rather strict manner.

There are many ways to write a workable program—and the methods you use in structuring your programs will have an impact on three important objectives:

- Safety
- Efficiency
- Ease-of-use (operator friendliness)

It may be impossible to come up with a perfect balance among these objectives. Generally speaking, what you do to improve one objective will negatively affect the other two. When faced with a choice, a beginning programmer's priorities should always lean toward safety and ease-of-use. Our recommended programming structure stresses these two objectives. We will, however, show some of the efficiency-related short-comings of our recommended methods—so you can improve efficiency as you gain proficiency.

Reasons for Structuring Programs with Consistent Format

Let's begin by discussing the reasons why you must write your programs using a strict structure.

Familiarization

You must have some way to get familiar with CNC programming. You'll need some help writing your first few programs. The formats we show will provide you with this help. You'll be able to use our given formats as a crutch until you (eventually) have them memorized.

Consistency

If you have been doing the exercises in this text, you've already worked on a few actual programs, filling in the blanks with needed CNC words. You have also seen several complete example programs in this course. You probably noticed that these programs are written in a very consistent manner. And the commands within each tool of each program are consistent with the other tools in the program.

Re-running Tools in the Program

This is the most important reason for structuring your programs using a strict format.

Say you're verifying a long program. You are fifteen tools into a twenty-tool program when you find a mistake. You must stop the cycle to correct the mistake. With the mistake corrected, you'll want to pick up where you left off—at the beginning of the fifteenth tool. You wouldn't want to re-run the entire program (from the beginning) just to get to tool number fifteen (doing so would be a waste of time).

Structured program format

The program formats will keep you from having to memorize most of the words and commands needed in CNC programming. As you'll see, a large percentage of most programs is related to a common structure.

We will show one program format for vertical machining centers and another for universal slant bed turning centers. There are four structure elements used in the program format:

1. Program-startup structure
2. Tool-startup structure
3. Tool-end structure
4. Program-end structure

Any time you begin writing a new program, start with the program-startup structure. You can copy this structure to begin your program. The actual values of some words will change based on what you wish to do in your own program, but the structure will remain the same every time you begin writing a new program.

After writing the program-startup structure comes the first tool-startup structure. This must be customized for the specific tool details. The tool-startup structure is followed by the motions for the first machining operation. Obviously, the motions to complete a particular machining operation will be somewhat unique, and you are on your own to write this section. When finished with the motions for a machining operation with a tool, you complete that section with the tool-end structure.

The pattern of tool-startup structure → machining operation motions → tool-end structure is repeated once for each tool operation required to machine the workpiece. Again, a tool-startup structure, followed by the motions for the machining operation and terminated by the tool-end structure.

When all the machining operations are programmed, you finish the program with the program-end structure.

One of the most important benefits of using the structured program format is that you do not have to memorize anything. You simply copy the structures and edit the appropriate words.

The structured program formats assume that you are using workpiece coordinate system offsets to assign program zero. If you come across a very old machine that does not support workpiece coordinate offsets.

Structured Program Format for Vertical Machining Centers

This structured program format is used for vertical machining centers that have workpiece coordinate system offsets. This particular format assumes that the machine has a double-arm automatic tool changer, that the tool change position is the Z-axis zero return position, and that the machine is resting at the tool change position in Z when the program begins. When this program ends, the machine is left at the X, Y and Z-axis zero return position. This makes a convenient workpiece loading position (with the table out toward the operator in the Y-axis).

For now you are to use the strict structure of these given formats. But the values that are shown in **bold** will change from program to program and from tool to tool.

Program Start-Up Structure:

O0001 (PROGRAM DESCRIPTION HERE)
 N010 G17 G20 G23 (SELECT XY-PLANE, INCH MODE, CANCEL STORED STROKE LIMIT)
 N020 G40 G50 G64 (CANCEL CUTTER COMP., SCALING, AND SELECT NORMAL CUT. MODE)
 N030 G67 G69 G80 (CANCEL CUSTOM MACRO, ROTATION, AND CANNED CYCLES)

Tool Start-Up Structure:

(DESCRIPTION/NAME FOR THIS TOOL)
 N100 **T01** M06 (LOAD TOOL IN SPINDLE)
 N110 G**54** G90 S**300** M03 T**02** (SELECT WORK OFFSET/ABSOLUTE MODE, START SPINDLE, READY NEXT TOOL)
 N120 G00 X**5.0** Y**5.0** (MOVE TO XY APPROACH POSITION)
 N130 G43 H**01** Z**0.1** (INSTALL TOOL LEN. COMP., MOVE TO Z APPROACH PSN)
 N140 M08 (COOLANT ON)
 N150 **G01** **F3.0** (CUTTING MOVES WITH FEEDRATE)

Tool Ending Structure:

N200 M09 (COOLANT OFF)
 N210 G91 G28 Z0 M19 (TOOL CHG. PSN., PRE-ORIENT SPINDLE)
 N220 M01 (OPTIONAL STOP)

Program Ending Structure:

N300 G91 G28 Z0 M19 (RETURN TO Z ZERO RETURN PSN, PRE-ORIENT SPINDLE)
 N310 G28 X0 Y0 (RETURN TO XY ZERO RETURN – LOAD/UNLOAD PSN)
 N320 M30 (END OF PROGRAM)

Example program:

Here is an example program that stresses the use of program structure . Although the program is quite simple (it is actually the same program shown earlier during our discussion of tool length compensation), it shows all of the principles of program formatting. Pay primary attention to the strict structure followed for each tool. The program can be broken into mini-programs, each making up one tool. Again, each tool is independent of the rest of the program.

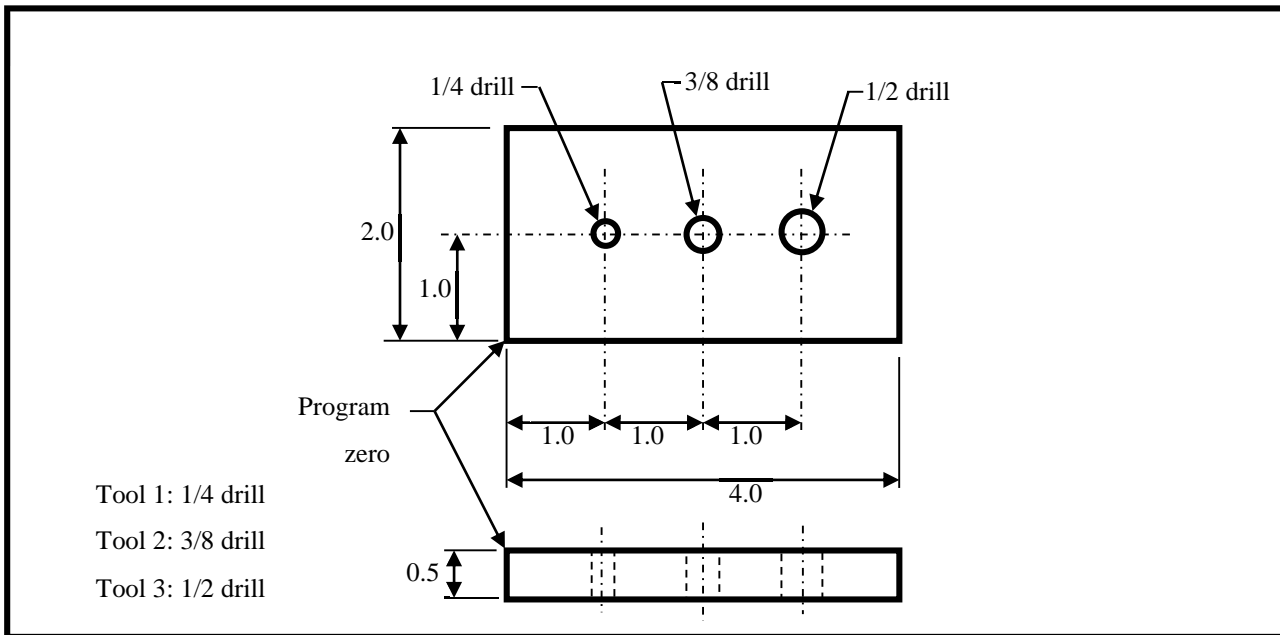


Figure 5.3: Example program to demonstrate the used of structured program format

Program-startup structure	<p>O0003 (EXAMPLE PROGRAM FORMAT) N010 G17 G20 G23 (SELECT XY-PLANE, INCH MODE, CANCEL STORED STROKE LIMIT) N020 G40 G50 G64 (CANCEL CUTTER COMP., SCALING, AND SELECT NORMAL CUT. MODE) N030 G67 G69 G80 (CANCEL CUSTOM MACRO, ROTATION, AND CANNED CYCLES)</p>
Tool-startup structure	<p>(1/4 DRILL) N100 T01 M06 (LOAD TOOL IN SPINDLE) N110 G54 G90 S1200 M03 T02 (START SPINDLE, READY NEXT TOOL) N120 G00 X1.0 Y1.0 (MOVE TO XY APPROACH POSITION) N130 G43 H01 Z0.1 (INSTALL TOOL LEN. COMP., MOVE TO Z APPROACH PSN) N140 M08 (COOLANT ON) N150 G01 Z-0.65 F4.0 (CUTTING MOVES WITH FEEDRATE) N160 G00 Z0.1 (RETRACT FROM HOLE)</p>
Tool-end structure	<p>N170 M09 (COOLANT OFF) N180 G91 G28 Z0 M19 (TOOL CHG. PSN., PRE-ORIENT SPINDLE) N190 M01 (OPTIONAL STOP)</p>
Tool-startup structure	<p>(3/8 DRILL) N200 T02 M06 (LOAD TOOL IN SPINDLE) N210 G54 G90 S1000 M03 T03 (START SPINDLE, READY NEXT TOOL) N220 G00 X2.0 Y1.0 (MOVE TO XY APPROACH POSITION) N230 G43 H02 Z0.1 (INSTALL TOOL LEN. COMP., MOVE TO Z APPROACH PSN) N240 M08 (COOLANT ON) N250 G01 Z-0.7 F5.0 (CUTTING MOVES WITH FEEDRATE) N260 G00 Z0.1 (RETRACT FROM HOLE)</p>
Tool-end structure	<p>N270 M09 (COOLANT OFF) N280 G91 G28 Z0 M19 (TOOL CHG. PSN., PRE-ORIENT SPINDLE) N290 M01 (OPTIONAL STOP)</p>
Tool-startup structure	<p>(1/2 DRILL) N300 T03 M06 (LOAD TOOL IN SPINDLE)</p>

Six Key Concepts Needed to Master CNC Programming

Tool-end structure	N310 G54 G90 S800 M03 T01 (START SPINDLE, READY NEXT TOOL)
	N320 G00 X3.0 Y1.0 (MOVE TO XY APPROACH POSITION)
	N330 G43 H03 Z0.1 (INSTALL TOOL LEN. COMP., MOVE TO Z APPROACH PSN)
	N340 M08 (COOLANT ON)
	N350 G01 Z-0.75 F6.0 (CUTTING MOVES WITH FEEDRATE)
	N360 G00 Z0.1 (RETRACT FROM HOLE)
Program-end structure	N370 M09 (COOLANT OFF)
	N380 G91 G28 Z0 M19 (TOOL CHG. PSN., PRE-ORIENT SPINDLE)
	N390 M01 (OPTIONAL STOP)
Program-end structure	N400 G91 G28 Z0 M19 (RETURN TO Z ZERO RETURN PSN, PRE-ORIENT SPINDLE)
	N410 G28 X0 Y0 (RETURN TO XY ZERO RETURN – LOAD/UNLOAD PSN)
	N420 M30 (END OF PROGRAM)

Structured Program Format for universal slant bed turning centers

Program startup structure

Follow these commands to begin a program.

O0001 (Program number)

N005 G99 G20 G23 (Select feed per revolution feedrate mode, select inch mode, cancel stored stroke limit)

N010 G50 S**4000** (Limit spindle speed to 4,000 rpm)

Oxxxx specifies the program number (O0001 in our case), and will change from program to program. All programs begin with a program number.

N005 is a safety command that ensures that important initialized modes are still active.

N010 specifies spindle limiting. Since most workpieces don't require spindle limiting, you normally specify the machine's maximum rpm (4,000 rpm in our case).

Tool startup structure

Follow these commands to begin each tool.

N015 T**0101** M**41** (Index to first tool, instate geometry and wear offset number one, select spindle speed range)

N020 G**96** S**350** M**03** (Select spindle mode, speed, and activate spindle in the forward direction)

N025 G00 X**3.0** Z**0.1** M08 (Move to approach position, and turn on the coolant)

N030 G01 X... Z... F**0.015** (The first cutting motion must include feedrate)

N015 specifies the turret index and selects the desired spindle range. The T-word selects the appropriate turret station (again, your first tool may not always be station number one). It also instates the geometry and wear offsets. We've assumed this tool will be run in the low spindle range, but you may want to begin with the high spindle range (M42 with most turning centers). We're also assuming that your machine has more than one spindle range. If it does not, you must omit the M41 (or M42) from this command.

N020 selects the spindle mode (css or rpm – G96 or G97), the appropriate speed with the S-word, and turns the spindle on in the appropriate direction. We're assuming that you're using right hand tools. If you use a left hand tool, the spindle must be started in the reverse direction (M04).

N025 makes the rapid approach movement to within a small distance (usually 0.1 inch) from the surface being machined. We're assuming you want to run the tool with flood coolant. If you do not, leave the M08 word out of this command.

N030 performs the first machining command. The feedrate word (F) is part of the program startup structure and must be included in the tool's first machining command.

Tool ending structure

Follow these commands to end every tool in your program.

N075 G00 X**8.0** Z**6.0** (Move to turret index position)

N080 M01 (Optional stop)

N075 rapids the tool back to a safe turret index position.

N080 is the optional stop (M01) that gives the setup person or operator the ability to stop the machine (with the optional stop on/off switch) to see what this tool has done. This is very helpful during the program's verification – and whenever trial machining must be done.

Program ending structure

Follow these commands to end a program.

N210 G00 X8.0 Z6.0 (Move to tool change position)

N215 M30 (End of program)

Program ending structure is almost identical to tool ending structure. The only difference is that you end with M30 (end of program) instead of M01. M30 will turn off anything that's still running (spindle & coolant), rewind the program to the beginning for the next workpiece, and stop the cycle.

With some machines, there may be additional commands required to end the program. If the machine has automatic doors, for example, you will want to include a command (usually an M-code) to open the door at the end of the program.

Machines with bar feeders will require commands to advance the bar.

Example program showing structure for turning centers

Figure 5.2 is the drawing we use to stress turning center structure.

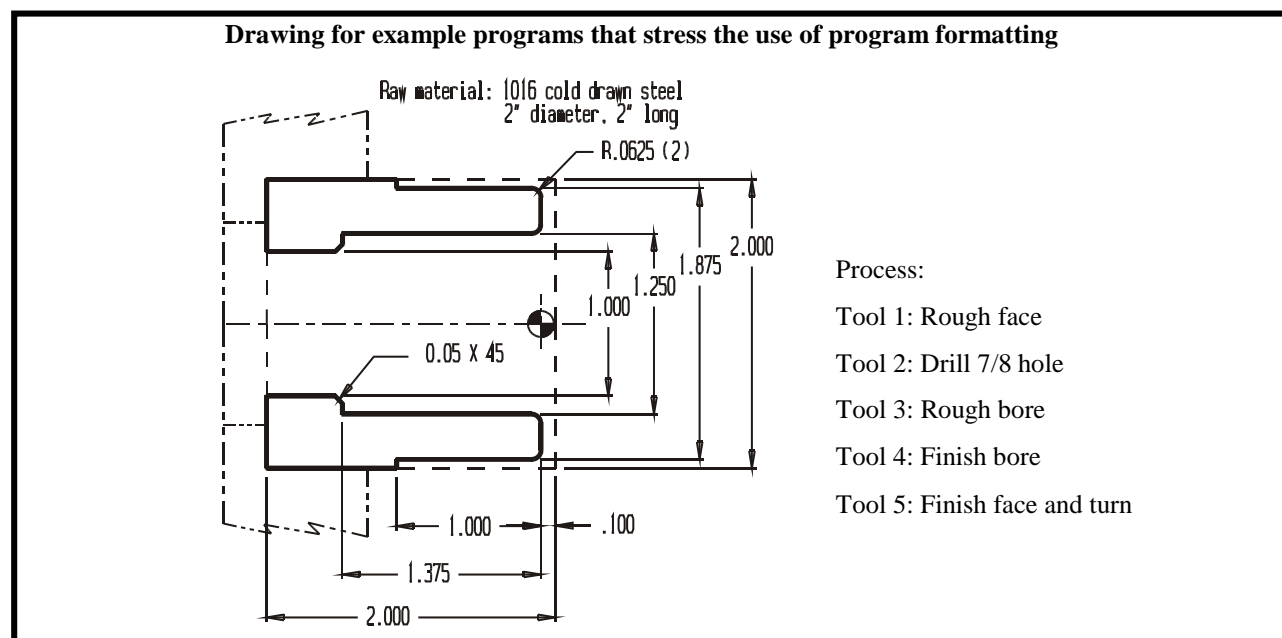


Figure 5.2 – Drawing to be used for example programs

Program:

O0002 (Program number)

(ROUGH FACING TOOL)

N002 G99 G20 G23 (Ensure that initialized modes are still in effect)

N004 G50 S5000 (Limit spindle speed to machine's maximum)

N005 T0101 M41 (Index turret, select spindle range)

N010 G96 S400 M03 (Start spindle in forward direction at 400 sfm)

N015 G00 X2.2 Z0.005 M08 (1) (Rapid to starting position, start coolant)

N020 G01 X-0.062 **F0.012** (2) (Select per revolution feedrate mode, face workpiece at 0.012 ipr)

N025 G00 Z0.1 (3) (Rapid away)

N030 X6.0 Z5.0 (Rapid to tool change position)

N035 M01 (Optional stop)

(7/8 DRILL)

N040 T0202 M41 (Index turret, select spindle range)

N045 G97 S354 M03 (Start spindle forward at 354 rpm)

N050 G00 X0 Z0.1 M08 (4) (Rapid into position, start coolant)

N055 G01 Z-2.2 **F0.008** (5) (Drill hole at 0.008 ipr)

N060 G00 Z0.1 (4) (Rapid out of hole)

N065 X6.0 Z5.0 (Rapid to tool change position)

N070 M01 (Optional stop)

(3/4 ROUGH BORING BAR)

N075 T0303 M42 (Index turret, select spindle range)

N080 G96 S350 M03 (Start spindle forward at 350 sfm)

N085 G00 X1.19 Z0.1 M08 (6) (Rapid into position, start coolant)

N090 G01 Z-1.37 **F0.007** (7) (Begin boring operation at 0.007 ipr)

N095 X0.875 (8)

N100 G00 Z0.1 (9) (Rapid out of hole)

N105 X5.0 Z6.0 (Rapid to tool change position)

N110 M01 (Optional stop)

(3/4 FINISH BORING BAR)

N115 T0404 M42 (Index turret, select spindle range)

N120 G96 S400 M03 (Start spindle forward at 400 sfm)

N125 G00 X1.375 Z0.1 M08 (10) (Rapid into position, start coolant)

N130 G01 Z0 **F0.005** (11) (Begin finish boring at 0.005 ipr)

N135 G02 X1.25 Z-0.0625 R0.0625 (12)

N140 G01 Z-1.375 (13)

N145 X1.1 (14)

N150 X1.0 Z-1.425 (15)

N155 Z-2.0 (16)

N160 G00 X0.8 (17)

N165 Z0.1 (18) (Rapid out of hole)

N170 X6.0 Z5.0 (Rapid to tool change position)

N175 M01 (Optional stop)

(FINISH FACE AND TURN TOOL)

N180 T0505 M42 (Index turret, select spindle range)

N185 G96 S450 M03 (Start spindle forward at 450 sfm)

N190 G00 X2.075 Z0 M08 (19) (Rapid into position, start coolant)

N195 G01 X1.05 **F0.006** (20) (Start finish facing and turning at 0.006 ipr)

N200 G00 Z0.1 (21)

N205 X1.75 (22)

N210 G01 Z0 (23)

N215 G03 X1.875 Z-0.0625 R0.0625 (24)

N220 G01 Z-1.0 (25)

N225 X2.2 (26)

N230 G00 X6.0 Z5.0 (Rapid back to tool change position)

N235 M30 (End of program)

Key Concept 6: Take advantage of features that simplify programming

Here is a list of common special programming features for machining centers:

- ✓ G73 - G89 Canned Cycles for Drilling - To simplify hole-machining operations
- ✓ M98 - M99 Sub-programming - To minimize redundant commands
- ✓ / - Block delete techniques - To give your operator a choice
- ✓ Sequence number techniques

- ✓ G02-G03 helical interpolation - For thread milling, machining external threads and when holes are too large to tap
- ✓ Other G-codes not discussed to this point
- ✓ M-codes not addressed to this point
- ✓ Rotary devices

Here is a list of common special programming features for turning centers:

- ✓ One pass canned cycles
- ✓ Multiple repetitive cycles
- ✓ Sub-programming

Example of machining center program simplification feature: Hole machining canned cycles

Almost all programs have at least some hole-machining operations. If you have been doing the exercises in this text, you have seen how tedious, time consuming, and error-prone it can be to program hole-machining operations with G00 and G01. You know that with G00 and G01, each hole will require at least three blocks, making the program quite long. And we have only performed basic drilling operations. Peck drilling, tapping, boring, and counter-boring operations will require even more blocks per hole.

Canned Cycles for Drilling will simplify the programming of hole-machining operations. Only one block is required per hole, regardless of the machining style (drill, peck drill, tap, ream, bore, counter-bore, etc.). Additionally, canned cycles are modal, meaning once you instate a canned cycle, you can continue machining additional holes with the same geometry by simply programming the coordinates of the hole. This will dramatically shorten the program's length, make programming easier, less time-consuming, and less error-prone.

Here is a list of the most common hole-machining canned cycles as FANUC names them in approximate order of popularity:

- G80 – Cancel any of the canned cycles
- G81 – Drilling cycle
- G73 – High-speed peck drilling cycle (breaks chips as the hole is machined)
- G83 – Peck drilling cycle (causes the drill to retract between pecks)
- G82 – Counter-boring cycle
- G84 – Right hand tapping cycle (also used for rigid tapping)
- G74 – Left hand tapping cycle (also used for rigid tapping)
- G86 – Boring cycle (rapids out of the hole)
- G89 – Boring cycle with dwell (pauses at hole bottom, rapids out)
- G76 – Fine boring cycle (leaves no witness mark in hole)
- G85 – Boring cycle (retracts from hole at the programmed feedrate)

Words used in canned cycles

As you have probably noticed, canned cycles share many words in common. These words will have exactly the same meaning in all canned cycles. Although some canned cycles do not use some of these words, when you understand how the shared words work in one canned cycle, you will know how they work in all. Here is a list of all words used in canned cycles, which cycles they apply to, and a brief description of their use:

WORD:	STATUS:	DESCRIPTION:
N	All cycles	Sequence number
G73-G89	All cycles	Canned cycle type
X	All cycles	X coordinate of the hole-center
Y	All cycles	Y coordinate of the hole-center
R	All cycles	Rapid plane position above work surface (specified from program zero in Z)
Z	All cycles	Z position of hole-bottom (specified from program zero in the Z-axis)

F	All cycles	Feedrate for the machining operation
L	All cycles	Number of holes to be machined in the command (used with incremental mode only)
G98	All cycles	Retract to the initial plane (initialized)
G99	All cycles	Retract to the R-plane
P	G82, G89	Pause time at hole bottom (P500 = .5 second)
Q	G73, G83	Peck drill amount per peck
I	G76	Amount and direction of move-over in X at hole bottom
J	G76	Amount and direction of move-over in Y at hole bottom

Example

Here is a simple example program that stresses the points made so far. Figure 6.6 shows the workpiece. We're simply drilling four holes using the drilling cycle, G81.

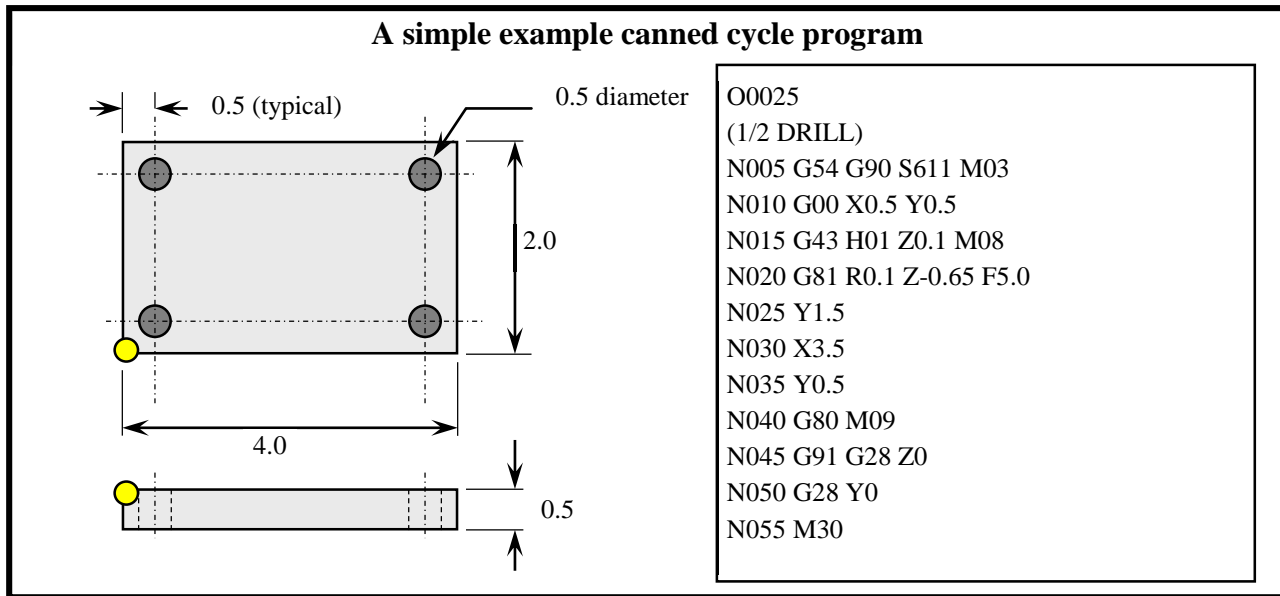


Figure 6.6 – A simple example

Program with comments:

```

O0025 (Program number)
(1/2 DRILL)
N005 G54 G90 S611 M03 (Select workpiece coordinate system setting offset #1, absolute mode, start spindle fwd at 611 rpm)
N010 G00 X0.5 Y0.5 (Rapid to first hole-location)
N015 G43 H01 Z0.1 M08 (Instate tool length compensation, position tool to just above work surface, start coolant)
N020 G81 R0.1 Z-0.65 F5.0 (Drill lower-left hole)
N025 Y1.5 (Drill upper-left hole)
N030 X3.5 (Drill upper-right hole)
N035 Y0.5 (Drill lower-right hole)
N040 G80 M09 (Cancel canned cycle, turn off coolant)
N045 G91 G28 Z0 (Move to Z-axis zero return position)
N050 G28 Y0 (Move to Y-axis zero return position)
N055 M30 (End of program)

```

Example of turning center program simplification feature: Rough turning and boring multiple repetitive cycle

If a roughing pass ends at a chamfer, taper, or radius, calculating the end point for the pass will require more difficult math calculations (including trigonometry), even for relatively simple workpieces. Figure 6.6 shows an example workpiece for which programming rough turning will be more difficult.

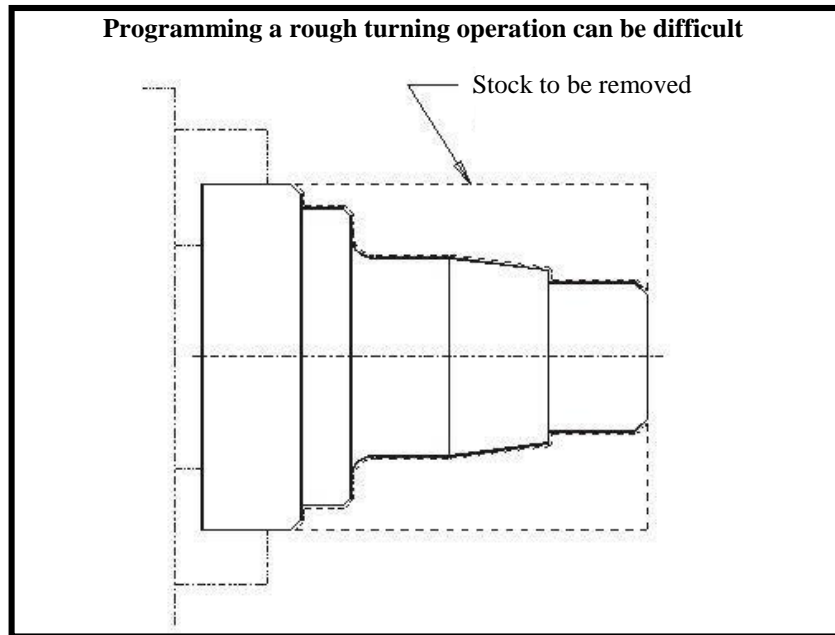


Figure 6.6 – Drawing that stresses the difficulty of programming roughing passes long-hand

Though this is a relatively simple workpiece, notice how much stock must be removed by the rough turning operation. To program this long hand (without any special cycles), a programmer will have to plan and program several roughing passes based on a previously determined *depth-of-cut*. This creates a real problem for manual programmers. And consider the additional difficulty if one or more of the roughing passes ends in the middle of the large taper or fillet radius. Trigonometry will be required in both cases.

Knowing how difficult it can be to program individual turning or boring passes, FANUC has designed three very helpful multiple repetitive cycles for roughing:

- G71 - Rough turning and rough boring
- G72 - Rough facing
- G73 - Pattern repeating

With roughing multiple repetitive cycles, you only need to describe the finish pass of the workpiece surface being roughed. Based on this *finish pass definition* and one very simple command, the control will completely rough the entire workpiece.

The two phases of G71

G71 and G72 commands will be completed in two *phases*. In the phase one, the machine will make a series of roughing passes based on a specified depth-of-cut. But as soon as the tool comes close to an end point for a roughing pass, the machine will immediately retract the tool for another pass. Figure 6.7 shows the motions generated during the first phase of the G71 command.

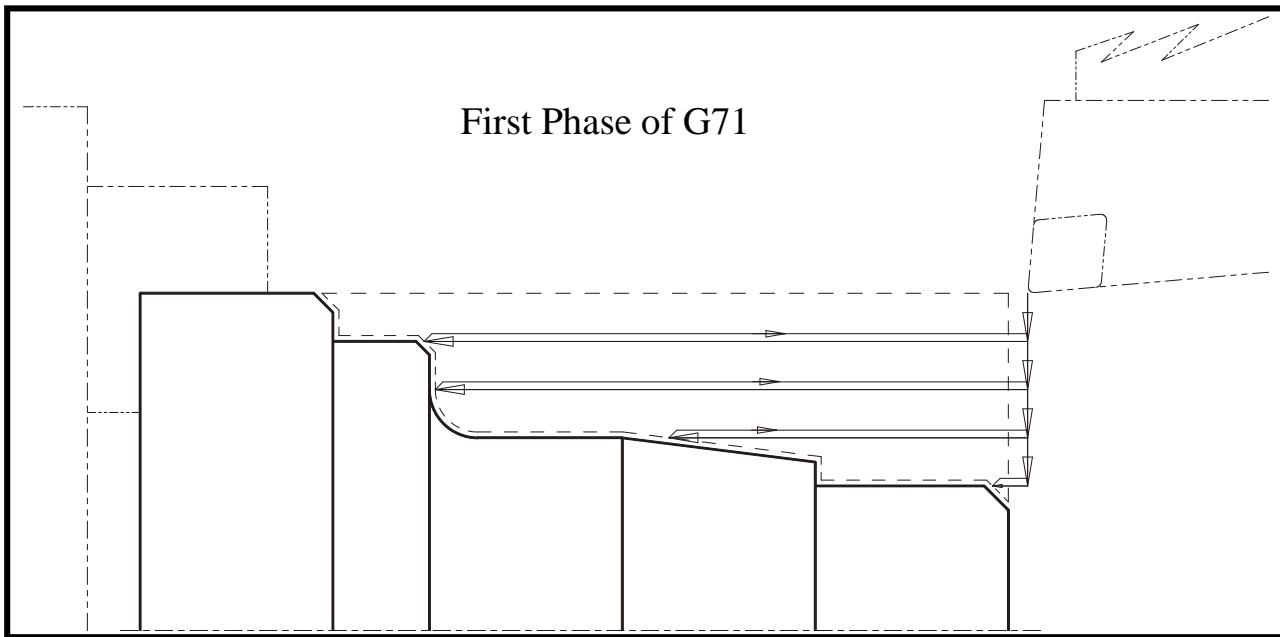


Figure 6.7 – Motions during phase one of a G71 command

After the control finishes the first phase of G71, the workpiece will have a series of steps (like a staircase) and it will not be close to its proper size (it will not have the appropriate amount of finishing stock on all surfaces). None of the steps left by the first phase of G71 will be larger than the depth-of-cut specified in the G71 command. Figure 6.8 shows what the workpiece will look like *after* the first phase of the G71 command.

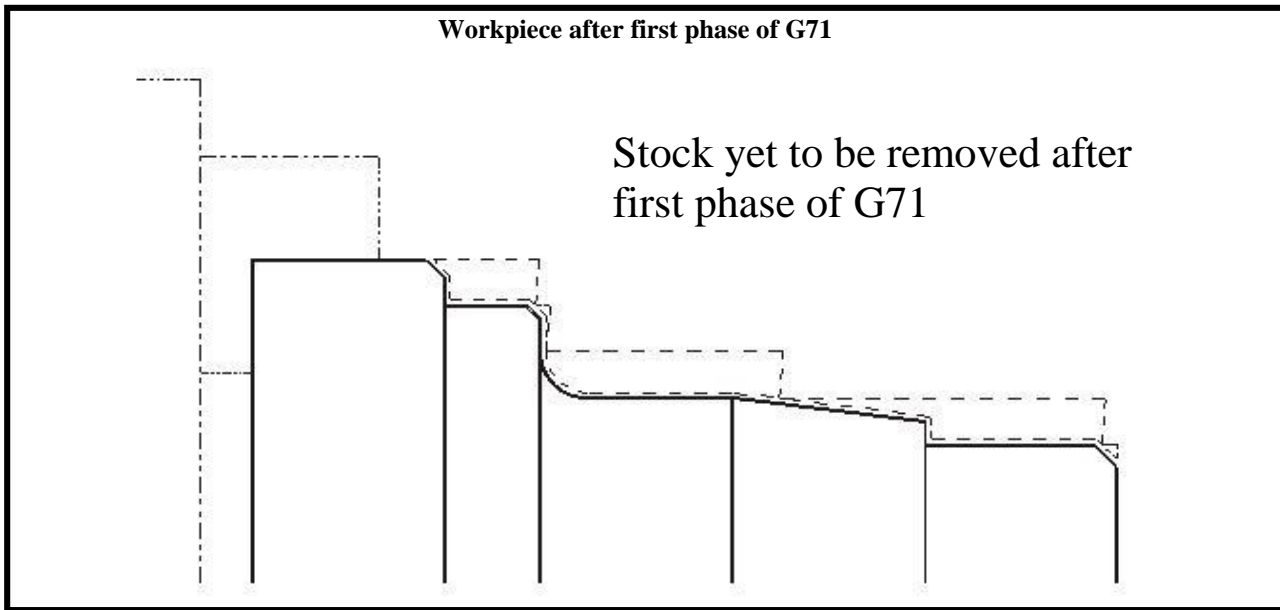


Figure 6.8 – What workpiece will look like after phase one of G71 command

In the second phase of G71, the machine will return to the starting point of the finish pass definition and make one sweeping semi-finish pass over the entire workpiece. It will stay away from the finished surface by the finishing stock values specified in the G71 command. Figure 6.9 shows the motions generated during the second phase of the G71 command.

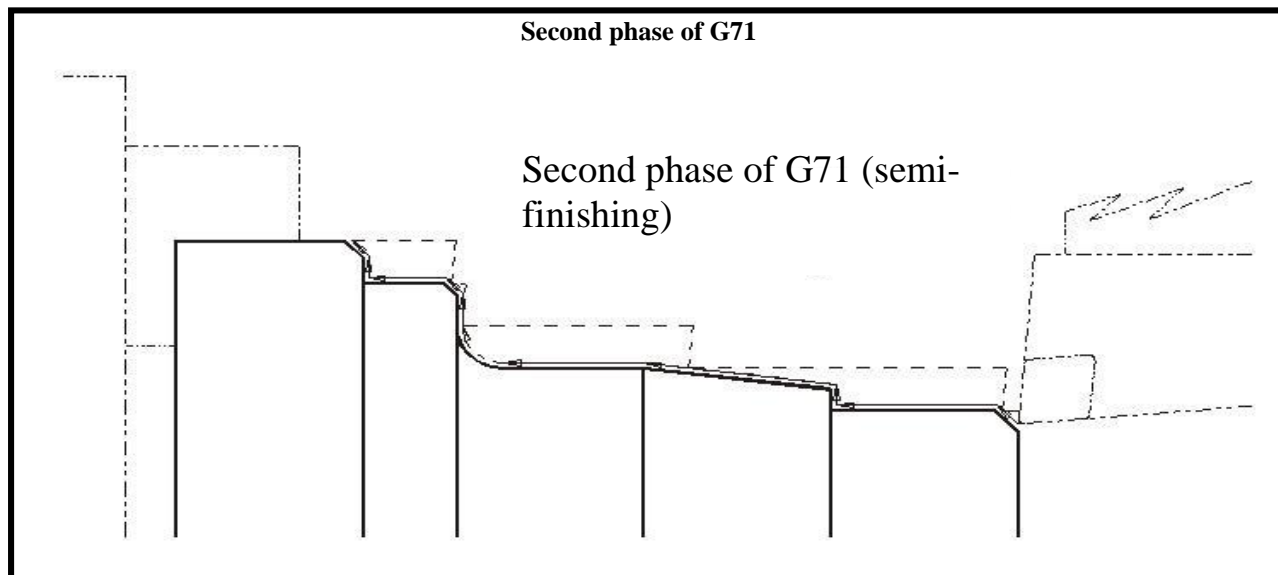


Figure 6.9 – Motions during phase two of G71 command

This is all accomplished by one command (the G71 command) together with the finish pass definition. G71 makes it very easy to program even very lengthy and complex rough turning and boring operations. In essence, the machine will figure out how to make all of the roughing passes for you, in much the same way a computer aided manufacturing (CAM) system does.

Example showing G71 for rough turning and G70 for finish turning

Admittedly, the G71 command probably sounds a little complicated at this point. When you see an example, things should clear up a bit. And regardless of how difficult you find this to be, it is well worth your time to study until you thoroughly understand the G71 and G70 commands. You will save countless programming hours when you master them.

Figure 6.10 shows the workpiece to be used for our first example program. Tool number one is the rough turning tool and tool two is the finish turning tool. The end of this workpiece has been previously faced to size.

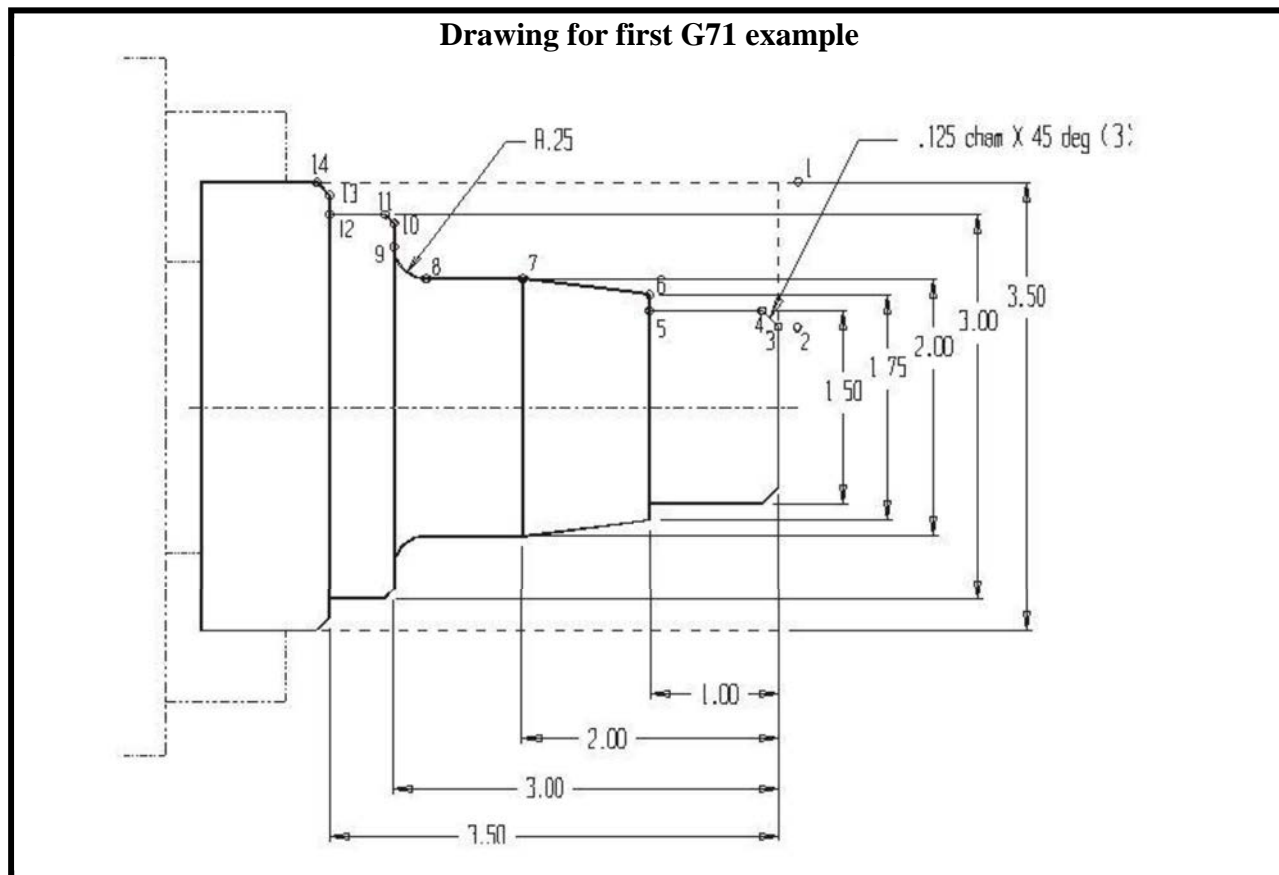


Figure 6.10 – Drawing for first G71 example program

Program:

```

O0010 (Program number)
(ROUGH TURNING TOOL)
N003 G99 G20 G23 (Ensure that initialized states are still in effect)
N004 G50 S4000 (No need for limiting, limit to machine's maximum speed)
N005 T0101 M41 (Select roughing tool and low spindle range)
N010 G99 G96 S400 M03 (Turn spindle on fwd at 400 sfm)
N015 G00 X3.5 Z0.1 M08 (Rapid to convenient starting position, point 1, start coolant)
N020 G71 P025 Q085 U0.040 W0.005 D1250 F0.015 (Rough turn entire workpiece based on what is between lines N025
and N085)
N025 G00 X1.25 (First block of finish pass definition, rapid to point 2)
N030 G01 Z0 F0.008 (Feed to point 3)
N035 X1.5 Z-0.125 (Feed to point 4)
N040 Z-1.0 (Feed to point 5)
N045 X1.75 (Feed to point 6)
N050 X2.0 Z-2.0 F0.005 (Feed to point 7)
N055 Z-2.75 F.008 (Feed to point 8)
N060 G02 X2.5 Z-3.0 R.25 (Circular move to point 9)
N065 G01 X2.75 (Feed to point 10)
N070 X3.0 Z-3.125 (Feed to point 11)
N075 Z-3.5 (Feed to point 12)
N080 X3.25 (Feed to point 13)

```

Six Key Concepts Needed to Master CNC Programming

N085 X3.5 Z-3.625 (Feed to point 14)

N090 G00 X6.0 Z6.0 (Rapid to safe index position)

N095 M01 (Optional stop)

(FINISH TURNING TOOL)

N100 T0202 M42 (Select finish turning tool and high spindle speed range)

N105 G96 S600 M03 (Turn spindle on fwd at 600 sfm)

N110 G00 **G42** X3.5 Z0.1 M08 (Rapid to the same convenient starting position as used for the rough turning operation, start coolant)

N115 **G70 P025 Q085 F0.008** (Go back to block N025 and do through block N085, finish turning workpiece)

N120 G00 **G40** X6.0 Z6.0 (Rapid to safe index point)

N125 M30 (End of program)