

## Know Your Machine From A Programmer's Viewpoint

*To fully master a CNC turning center, you must view it from two distinctly different perspectives. Throughout this text, we will be looking at the machine from a programmer's viewpoint. In a separate text – CNC Turning Center Setup and Operation – we look at the machine from a setup person's or operator's viewpoint.*

**K**ey Concept Number One contains five lessons:

- 1: Machine configurations
- 2: Understanding turning center speeds and feeds
- 3: Visualizing program execution
- 4: Understanding program zero
- 5: Introduction to programming words

A CNC programmer need not be nearly as intimate with a CNC turning center as a setup person or operator – but they must, of course, understand enough about the machine to create programs – to instruct setup people and operators – and to provide the related setup and production run documentation.

First and foremost, a CNC programmer must understand what the CNC turning center is designed to do. That is, they must understand the machining operations a turning center can perform. They must be able to develop a workable process (sequence of machining operations), select appropriate cutting tools for each machining operation, determine cutting conditions for each cutting tool, and design a workholding setup. All of these skills, of course, are related to basic machining practice – which as we state in the Preface – are beyond the scope of this text. For the most part, we'll be assuming you possess these important skills.

Though this is the case, we do include some important information about machining operations that can be performed on CNC turning centers throughout this text. For example, we discuss how to develop tool paths for machining operations in Lessons Six and Seven. We provide a description of rough and finish turning and boring in Lesson Fourteen. Threading is discussed in Lesson Sixteen. And in general, we provide suggestions about how machining operations can be programmed when it is appropriate. This information should be adequate to help you understand enough about machining operations to begin working with CNC turning centers.

### **If you've had experience with conventional (non-CNC) machine tools...**

A CNC turning center can be compared to an engine lathe (or any kind of conventional lathe). Many of the same operations performed on a conventional lathe are performed on a CNC turning center. If you have experience with manually operated lathes, you already have a good foundation on which to build your knowledge of CNC turning centers.

**Lesson 1:**

**Machine Configurations**

*As a programmer, you must understand what makes up a CNC turning 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.*

Most beginners tend to be a little intimidated when they see a turning center in operation for the first time. Admittedly, there will be a number of new functions to learn. The first point to make is that you must not let the machine intimidate you. As you go along in this text, you will find that a turning center is very logical and is almost easy to understand with proper instruction.

You can think of any CNC machine as being little more than the standard type of equipment it is replacing with very sophisticated and automatic motion control added. Instead of activating things manually by hand-wheels and manual labor, you will be preparing a *program* that tells the machine what to do. Virtually anything that needs to be done on a true CNC turning center can be activated through a program – meaning anything you need the machine to do can be commanded in a program.

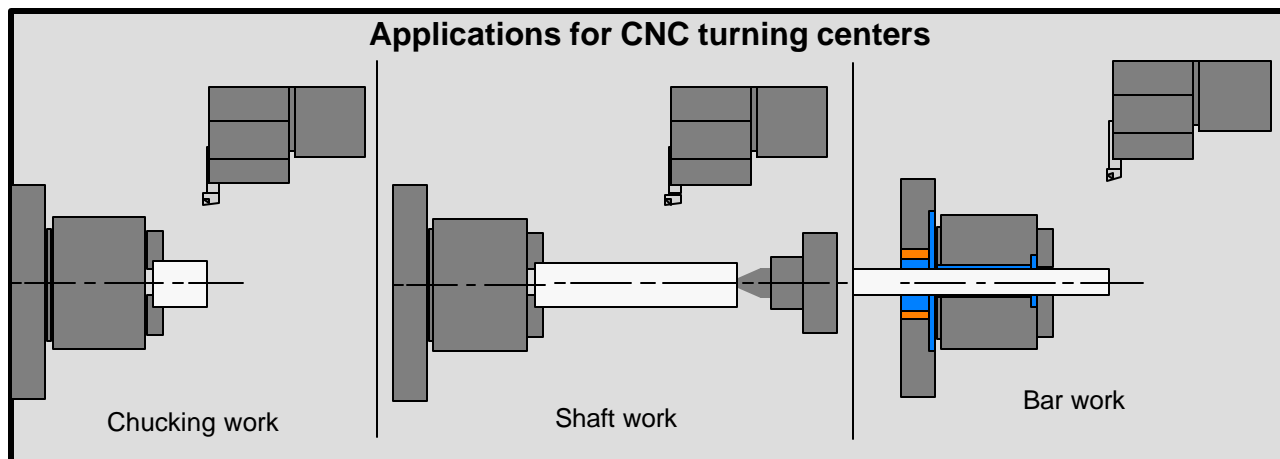
**Special note:** Much of the material in this lesson is also included in the *CNC Turning center Setup and Operation* self-study manual. If you have already read the setup and operation manual, please consider this lesson as a review.

**Types of CNC turning centers**

There are several types of CNC turning centers. While at first glance there may appear to be substantial differences among the various types, all turning centers share several commonalities. We'll begin by describing the most popular type of CNC turning center – the *universal style slant bed turning center*. Because it is so popular, this is the machine type we will use for all examples in this text. We will then introduce several other types of turning centers, comparing them to the universal style slant bed turning center.

**Universal style slant bed turning center**

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.



**Lesson 2****Understanding Turning Center Speeds and Feeds**

*Speed and feed selection is one of the most important basic-machining-practice-skills a programmer must possess. Poor selection of spindle speed and feedrate can result in poor surface finish, scrapped parts, and dangerous situations. Even if speeds and feeds do allow acceptable workpieces to be machined – if they're not efficient – productivity will suffer.*

As you know, spindle speed is the rotation rate of the machine's spindle (and workpiece). Feedrate is the motion rate of the cutting tool as it machines a workpiece. These two *cutting conditions* are extremely important to machining good workpieces. You will select a spindle speed and feedrate for every machining operation that must be performed.

Cutting tool manufacturers supply technical data including the recommendation of a cutting speed and feedrate for the cutting tools they supply. Recommendations are based on three important factors:

- 1) The machining operation to be performed
- 2) The material to be machined
- 3) The material of the cutting tool's cutting edge

***The machining operation to be performed***

This criterion determines the style of cutting tool that must be used to perform the machining operation. As the programmer, *you* will be the person making this decision. It requires that you draw upon your basic machining practice experience.

As stated in the Preface and Lesson One, turning centers can perform a wide variety of machining operations, and specific cutting tools are used to perform each operation. The specified spindle speed and feedrate must be appropriate to the cutting tool selection. Some cutting tools, like rough turning tools, perform very powerful machining operations and can remove a great deal of material from the workpiece per pass – while others, like small boring bars, perform lighter machining operations and can only remove a small amount of workpiece material per pass.

***The material to be machined***

This criterion determines (among other things) how quickly material can be removed from the workpiece. Soft materials, like aluminum can be machined faster than hard materials, like tool steel. So generally speaking, you'll use faster spindle speeds and feedrates for softer materials.

***The material of the cutting tool's cutting edge***

The material used for some cutting tools comprises the entire tool. A standard twist drill or end mill, for example, is made entirely of high speed steel (or cobalt, or some other material).

With other tools, like many turning tools, boring bars, grooving tools, and threading tools, the shank of the tool is made from one material (like steel), while the very cutting edge of the tool is made from another material (like carbide or ceramic). This lowers the cost of the cutting tool. With these kinds of cutting tools, only the cutting edge of the cutting tool is made from the (expensive) cutting tool material. The cutting edge component of the tool is called an insert. Figure 1.15 shows this kind of cutting tool.

**Lesson 3****Visualizing The Execution Of A CNC Program**

*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 turning center will be doing, the easier it will be to prepare a workable CNC program.*

Once again, we stress the importance of basic machining practice as it applies to CNC turning center usage. A machinist that has experience running a conventional lathe will have seen machining operations taking place many times. While this experience by itself does not guarantee the ability to *visualize* a machining operation (seeing the machining operation taking place in your mind), it dramatically simplifies the task of learning how to visualize a CNC program's execution.

When a machinist prepares to machine a workpiece on a conventional lathe, they will have all related components needed for the job right in front of them. The machine, cutting tools, workholding setup (chuck), and part are ready for immediate use. It is highly unlikely that the machinist will make a basic mistake like forgetting to start the spindle before trying to machine the workpiece.

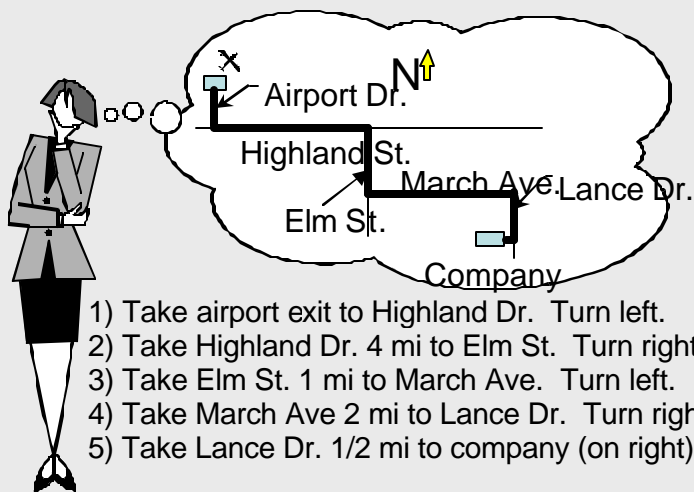
On the other hand, a CNC turning center programmer will be writing the program with only the workpiece drawing to reference. No tooling – no machine – and no workholding setup will be available to them. For this reason, a programmer must be able to visualize just exactly what will happen during the execution of the program – and this can sometimes be difficult, since this visualization must take place in the programmer's mind. A beginning programmer will be prone to forget certain things – sometimes very basic things (like turning the spindle on prior to machining the workpiece).

In this lesson, we will acquaint you with those things a programmer must be able to visualize. We will also show the first (elementary) program example to stress the importance of visualization.

### Visualization is necessary to develop a set 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 be able to visualize the path from the airport to your company. If you cannot visualize the path, you can't write the instructions. Worse, if you think you can visualize the path (but you're wrong), you'll write incorrect instructions and the person following your instructions will get lost!

In similar fashion, if you cannot visualize the path a cutting tool will take as it machines a workpiece, you cannot write the CNC commands that drive the cutting tool through this path.



- 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).

**Lesson 4**

**Program Zero And The Rectangular Coordinate System**

*A programmer must be able to specify positions through 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 program zero*

You know that turning centers have two linear axes – X and Z. You also know these axes move and that they have a polarity (plus versus minus). In order to machine a workpiece in the desired manner, each axis must, of course, be moved in a controlled manner. One of the ways you must be able to control each axis is with precise *positioning control*.

In the early days of NC (before CNC, over forty years ago), a programmer was required to specify drive motor rotation in order to cause axis motion. This meant they had to know how many rotations of an axis drive motor equated to the desired amount of linear motion for the moving component of the axis (turret). As you can imagine, this was extremely difficult – it was not logical. There is no relationship between drive motor rotation and motion of the moving component. Today, thanks to program zero and the rectangular coordinate system, specifying positions through which cutting tools will move is much easier.

The rectangular coordinate system has an origin point that we'll be calling *program zero*. It allows you to specify all positions (we'll be calling *coordinates*) from this central location. As a programmer, *you* will be choosing the location for program zero – and if you choose it wisely, many of the coordinates you will use in the program will come directly from your workpiece drawing, meaning the number and difficulty of calculations required for your program can be reduced.

**Graph analogy**

We use a simple graph to help you understand the rectangular coordinate system as it applies to CNC. Since everyone has had to interpret a graph at one time or another, we should be able to easily relate what you already know to CNC coordinates. Figure 1.21 is a graph showing a company's productivity for last year.

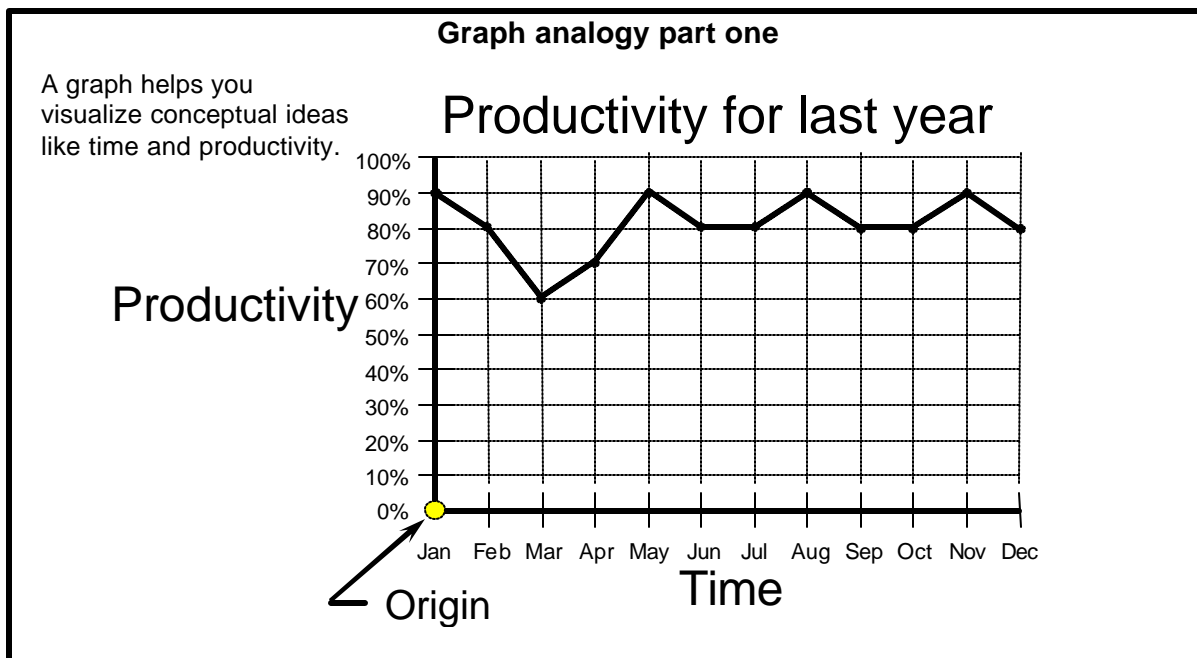


Figure 1.21 Graph example used to illustrate rectangular coordinate system

**Lesson 5****Introduction To Programming Words**

*As you know, all CNC words include a letter address and a numerical value. The letter address identifies the word type. You should be able to quickly recognize the most common CNC word types.*

You know that CNC programs are made up of commands – and that commands are made up of words. Words are categorized into types – and each word type has a special meaning to the machine. Each word type is designated by a *letter address*. You already know a few the letter addresses, like N for sequence number, G for preparatory function, X and Z for axis designations, S for spindle speed, F for feedrate, and M for miscellaneous (or machine) functions. In this lesson, we're going to introduce the rest of the word types.

If you are a beginner looking at the word types for the first time, you may want to read this section a few times to get better acquainted with these word types. Note that we are *not* asking you to memorize the word types – just to get familiar with them. In Key Concept Number Five – program formatting – we will provide you with a way to remember each word's function.

Also, this lesson is only intended to *introduce* each word, not to give you an in-depth description. When appropriate, we'll point you to the lessons that discuss the word type in more detail.

You will find that certain words are seldom used, meaning you will have little or no need for them. Other words are constantly used, and you will soon have them memorized.

Some CNC words have more than one function, depending on commanded format. We will be showing you the primary (most common) function of the word next to the "A" description and the secondary uses for the word next to the "B" and "C" descriptions.

Once you've seen a word type a few times, it should not be too difficult to remember its function. Again, most word types are aptly named with an easy-to-remember letter address. Additionally, only about fifty words are used consistently, so try to look at learning to program a CNC turning center as like learning a foreign language that contains only fifty words.

As you continue with this text, *use this lesson as a reference*. If you come across a word or word type you don't recognize, remember to come back to this lesson. You've probably already noticed that we provide a quick reference for CNC words on the inside front and back cover of this text, but information in this lesson is a little more detailed.

**Words allowing a decimal point**

Current CNC controls allow you to include a decimal point in those words that are used to specify *real numbers* (values that require a portion of a whole number). You must remember to include a decimal point with these words or the control will revert to the fixed format for the word. Word types that allow a decimal point include:

X, Y, Z, U, W, C, I, K, F, E, and R

Certain CNC words are used to specify *integers* (whole numbers). These word types do *not* allow a decimal point:

O, N, G, P, L, S, T, and M

## Key Concept

## 2

## You Must Prepare To Write 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.*

Key Concept Number Two is made up of but one lesson:

### 6: Preparation for programming

Frankly speaking, this Key Concept can be applied to any CNC related task, including setup and operation. Truly, the better prepared you are to perform a task, the easier it will be to correctly complete the task. When it comes to making setups, for example, if you have all of the needed components to make the setup at hand (jaws, cutting tools, cutting tool components, program, documentation, etc.), making the setup will be much easier and go faster. So *gathering needed components* is always a great preparation step to perform.

The same goes for completing a production run. If the CNC operator has all needed components (raw material, inserts for dull tool replacement, a place to store completed workpieces, gauging tools, etc.), they will be able to smoothly complete the production run.

In this Key Concept, however, we're going to limit our discussion to preparation steps that you can perform to get ready to write a CNC program. Remember, the actual task of *writing* a CNC program is but part of what you must do. Certain things must be done *before* you're adequately prepared to write the 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 ready 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 worker makes a guess and attempts to fit the pieces together. Since the worker 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 jig saw 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 with the process. Worse, the program may be completed and being verified on the CNC turning center before some critical error is found.

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

**Lesson 6****Preparation Steps For Programming**

*Any complex project can be simplified by breaking it down into small pieces. This can make seemingly insurmountable tasks much easier to handle. CNC turning center programming is no exception. Learning how to break up this complex task will be the primary focus of this lesson.*

As you now know, preparation will make programming easier, safer, and less error-prone. Now let's look at some specific steps you can perform to prepare to write CNC programs.

**Prepare the machining process**

*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, of course, 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.

In most manufacturing companies, this involves *routing* the workpiece through a series of different machine tools and processes. Each machine tool along the way will perform only those operations the process planner intends, as specified on the routing sheet. This commonly means that some non-CNC machine tools are needed to complete a given workpiece. For example, the workpiece shown in practice exercise at the end of Lesson Six might have the following routing sheet.

<b>Op. #:</b>	<b>Operation:</b>	<b>Machine:</b>
10	Procure material	Vender ID #12322
20	Cut bar stock to 5.7" long	Cut off saw
30	Clean and de-burr	Cleaning tanks
40	Face, turn, drill, and bore	CNC turning center
50	Clean and de-burr	Cleaning tanks
60	Plate with nickel	Finishing tanks

Note that only operation 40 of this process requires a CNC machine tool. When a CNC machine is involved in the process plan, often the CNC machine will be required to perform several machining operations on the workpiece (as is the case in our example). As you know, CNC turning centers are multi-tool machines, meaning several tools can be used by one program. In some companies, the process sheet will clearly specify the order of machining operations that must be performed by the CNC machine. However, the vast majority of process planners do not get so specific with their routing sheets. Instead, the *sequence of machining operations to be performed on the CNC machine is left completely to the CNC programmer*. If the programmer must develop the machining order, of course, they must possess a good understanding of basic machining practice.

Regardless of *who* does the planning, the step-by-step machining order required to machine a workpiece on the CNC machine must be developed before the CNC program can be written. With a simple process, an experienced programmer may elect to develop the process as the program is being written. While some experienced programmers have the ability to do this, beginning programmers will find it necessary to plan *and document* the machining process first.



## Key Concept

## 3

## Understand The Motion Types

*Motion control is at the heart of any CNC machine tool. CNC turning 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 Number Three.*

Key Concept Number Three is another one-lesson Key Concept:

### 7: Motion types

You know that all CNC turning centers have at least two axes (X and Z). You also know that you can specify positions (coordinates) along each axis relative to the program zero point. These coordinates will be positions through which your cutting tools will move. In Key Concept Number Three, we're going to discuss the ways to specify *how* cutting tools move from one position to another.

### What is interpolation?

If a single linear axis is moving (X or Z), the motion will of course, be along a perfectly straight line. For example, look at Figure 3.1.

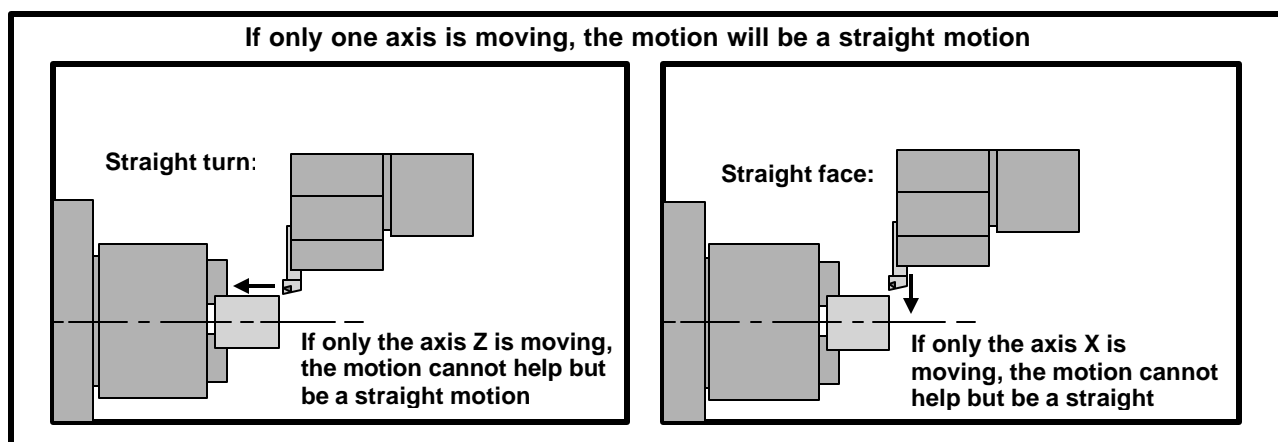


Figure 3.1 – A perfectly straight motion will occur if only one axis is moving

When turning a straight diameter on the workpiece (left view in Figure 3.1), only the Z axis is moving. And since the Z axis is a *linear* axis, this will force the motion to be perfectly straight – and parallel to the Z axis. In like fashion, when facing a workpiece (right view), only the X axis is moving – and again – this motion will be perfectly straight – parallel to the X axis.

But sometimes a straight surface on a workpiece is *tapered*. Additionally, there will be times when a cutting tool must machine a radius. This will require that *both* the X and Z axes move in a controlled manner, as shown in Figure 3.2.

**Lesson 7****Programming The Three Most Basic Motion Types**

*There are only three motion types used in CNC turning center programs on a regular basis – rapid, straight-line, and circular motion. You must understand how they are commanded.*

While it helps to understand how the turning center will interpolate motion, it is not as important as *knowing how to specify motion commands in a program*. This is the focus of Lesson Seven. Let's begin our discussion by showing those things that all motion types share in common.

**Motion commonalities**

All motion types share five things in common:

First, they are all modal, meaning a motion type will remain in effect until it is changed. If more than one consecutive movement of the same type must be made, you need only include the motion type G code in the *first* command of the series of movements.

Second, each motion type command requires the *end point* of the motion. The control will assume the tool is positioned at the starting point of the motion prior to the motion command. Think of motion commands that form a tool path as being like a series of connect-the-dots.

Third, all motion commands are affected by whether or not you specify coordinates with absolute or incremental positioning. With most turning center controls, X and Z are used to specify absolute positioning. U and W are used to specify incremental positioning. As stated in Lesson Four, you should concentrate on specifying coordinates with absolute positioning (X and Z).

Fourth, each motion command requires *only* the moving axes. If specifying a motion in only one axis, only one axis specification (X or Z) need be included in the motion command. *Axes that are not moving can be (and should be) left out of the command.*

Fifth, leading zeros can be left out of the G codes related to motion types. 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. All examples in this text do include the leading zero.

**Understanding the programmed point of each cutting tool**

In order to generate appropriate motions for cutting tools, you must know the location on each cutting tool that is being programmed – which we're calling the *programmed point*. In some cases, you will have to modify program coordinates to compensate for certain attributes of cutting tools.

For example, hole-machining tools like drills, taps, and reamers will have a certain amount of *lead* that must be compensated when Z axis (depth) coordinates are calculated. You can calculate the lead for a twist drill (118 degree point angle) by multiplying 0.3 times the drill diameter. This lead must be added to the hole's depth when plunging through holes. See Figure 3.5.

## Key Concept

## 4

## Know The Compensation Types

*CNC turning centers provide three kinds of compensation to help you deal with tooling related problems. In essence, each compensation type allows you to create your CNC program without having to know every detail about your tooling. The setup person will enter certain tooling information into the machine separately from the program.*

Key Concept Four is made up of four lessons:

- 8: Introduction to compensation
- 9: Geometry offsets and wear offsets
- 10: Tool nose radius compensation

The fourth Key Concept is that you must know the three kinds of compensation designed to let you ignore certain tooling problems as you develop CNC programs. In Lesson Eight, we'll introduce you to compensation, showing the reasons why compensation is needed on CNC turning centers.

Two of the compensation types are related to cutting tools: wear offsets and tool nose radius compensation. We discuss them in Lessons Nine and Ten. The third compensation type, geometry offsets, is more related to work holding devices – and we'll also discuss geometry offsets in Lesson Nine. While we do introduce geometry offsets in Key Concept Number One, we present more about them in Lesson Nine.

### More on interpreting tolerances

We cannot overstress the importance of being able to interpret tolerances. Every CNC person must possess this ability. Many of the discussions made during Key Concept Four assume you have this ability. Let's begin by reviewing some important points that are related to interpreting tolerances. If you've had machine shop experience, consider this to be a review of what you already know. But if you haven't had shop experience, be sure you understand the material presented here before moving on to Lesson Nine.

As mentioned previously, *all* dimensions have tolerances. And from Lesson Four, you know that 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. One way to calculate the mean value for any dimension is to divide the overall tolerance for the dimension by two, and add the result to the low limit for the tolerance.

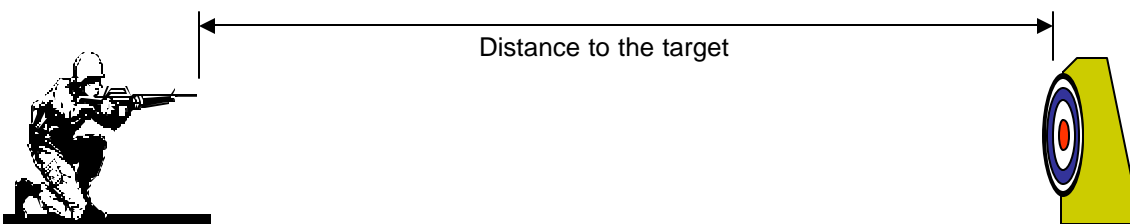
Companies vary when it comes to how tight (small) the tolerances are that they machine on their CNC turning centers. Generally speaking, overall tolerances over about 0.010 inch (about 0.25 mm) are considered pretty open (easy to hold with today's CNC turning centers). Tolerances between 0.002 and 0.010 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 turning centers – can be quite challenging – especially when a large number of workpieces must be produced. While there are exceptions, about the smallest overall tolerance most companies regularly hold on their CNC turning centers is 0.0002 inch (0.0051 mm).

**Lesson 8****Introduction To Compensation**

*An airplane pilot must compensate for wind direction and velocity when setting a heading. A race-car driver must compensate for track conditions as they negotiate a turn. A marksman must compensate for the distance to the target when firing a rifle. And a CNC programmer must compensate for certain tooling-related problems as programs are written.*

**What is compensation and why is it 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 (just as the marksman does their best in judging the distance to the target and adjusting the sight). 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 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 when very tight (small) tolerances must be held. With tight tolerances, even a small machining imperfection will cause a scrap workpiece.

**Lesson 9****Geometry Offsets And Wear Offsets**

*Most of what you need to know about geometry offsets and wear offsets is presented in a separate manual entitled *CNC Turning Center Setup and Operation*. There are just a few more points we want to make about these important CNC features – taken from a programmer’s viewpoint.*

You know from the CNC Turning Center Setup and Operation manual that current model Fanuc controls are equipped with geometry offsets. You also know that geometry offsets will be your method of choice for assigning program zero. In this lesson, we’re assuming that your turning center has geometry offsets, and that you’re using them to assign program zero.

You know that wear offsets allow you to make sizing adjustments – during the setup to allow for tool positioning imperfections and during the production run to allow for tool wear. The machine will simply add the geometry and wear offset values together to come up with the total amount of offset.

**Review of reasons for using geometry offsets to assign program zero**

Again, there are several advantages to using geometry offsets. Here is a quick list of the major advantages.

- ✍ Program zero assignment is separated from the program (improved ease of use)
- ✍ Only two values must be entered for each tool (improved ease of use)
- ✍ If the control is equipped with the measure function, error prone calculations and offset entries are eliminated (safety is improved)
- ✍ With the feature *work shift*, redundant Z measurements for cutting tools left in the turret from job to job can be eliminated (improved efficiency)
- ✍ The turret can be in any location at the beginning of the program (improved efficiency and ease of use)
- ✍ The turret can be in any location when re-running tools (improved efficiency and ease of use)
- ✍ For tool changing, the turret can be sent to any safe index point (improved efficiency)

**How geometry offsets work**

If you have read the setup and operation text, you know 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*. You know 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.

**Lesson 10****Tool Nose Radius Compensation**

*This compensation type allows you to deal with problems caused by the small radius that is on the cutting edge of all single point turning tools and boring bars. With the addition of two G codes per tool in your program – and with the entry of two simple values in the tool's offset – the machine will automatically keep the cutting edge of cutting tools flush with the surfaces they machine.*

You know that you must sometimes compensate for tooling attributes when you calculate the coordinates for your programs. With a twist drill, for example, you must compensate for the lead of the drill when calculating hole depth. In similar fashion, you must compensate for the lead of a reamer – or the number of imperfect threads on a tap.

There is one other 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.

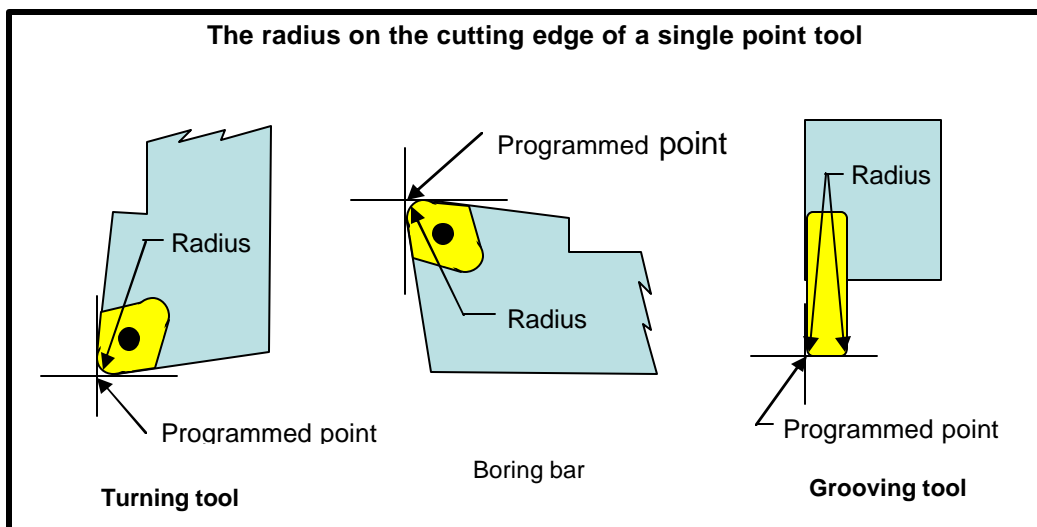


Figure 4.12 – All single point tools have a small radius on the cutting edge

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.

## Key Concept

## 5

## You Must Provide Structure To Your CNC Programs

*While there are many ways to write programs, you must ensure that your programs are safe and easy to use, yet as efficient as possible. This can be a real challenge since safety and efficiency usually conflict with one another.*

Key Concept Number Five contains two lessons:

11: Introduction to program structure

12: Four types of program format

The fifth Key Concept is you must structure your CNC programs using a strict format – while incorporating a design that accomplishes the objectives you intend. Though it is important to create *efficient* programs, *safety and ease-of-use* must take priority – at least until you gain proficiency. In Key Concept Number Five, we'll be showing techniques that stress *safety as the top priority*.

To this point in the text, we have been presenting the building blocks of CNC programming – providing the needed individual tools. Machine components, axes of motion, program zero, and programmable functions are presented in Key Concept One. Preparation steps in Key Concept Two. Motion types in Key Concept Three. Compensation types in Key Concept Four. In Key Concept Five, we're going to draw all of these topics together, showing you what it takes to write CNC turning center programs completely on your own – making you a *self-sufficient programmer*.

Lesson Eleven will introduce you to a CNC program's structure, showing you the reasons *why* programs must be strictly formatted. We'll also review some the program-structure-related points we've made to this point – and present a few new ones. And we'll address some variations related to how certain machine functions are handled.

In Lesson Twelve, we'll show the four types of program format as they are applied to CNC turning centers. These formats can be used as a crutch to help you write your first few programs.

You may be surprised at how much you already know about a program's structure, especially if you have been doing the exercises included in this text and/or those in the workbook. We have introduced many of the CNC words used in programming, and we have been following the structure-related suggestions that we will be recommending here in Key Concept Five.

For example, do you recognize any of these words and commands? If so, write their meanings in the space provided.

O0001: \_\_\_\_\_

T0101 M41: \_\_\_\_\_

G96 S500 M03: \_\_\_\_\_

**Lesson 11****Introduction To Program Structure**

*Structuring a CNC program is the act of writing a program in a way that the CNC machine can recognize and execute safely, efficiently, and with a high degree of operator-friendliness.*

You know that CNC programs are made up of commands, that each command is made up of words, and that each word is made up of a letter address and a numerical value.

You also know that programs are executed sequentially – command by command. The machine will read, interpret, and execute the first command in the program. Then it will move on to the next command. Read – interpret – execute. It will continue doing so until the entire program has been executed.

You have seen several complete programs so far in this text – you have even worked on a few if you have done the exercises in this text and in the workbook. You have probably noticed that there is quite bit of consistency and *structure* in the CNC programs we have shown.

Our focus in Lesson Twelve will be to help you understand more about the structure that is used in CNC programming.

**Objectives of your program structure**

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 will go into an alarm state – 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 the three most 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.

We’re going to be assuming that *you have control* of the structure you use to write programs. Your company may, however, already have a programmer that is writing programs with a different structure. As long as these programs are working – and satisfying the company’s objectives – you’re going to have to adapt to the established structure. If you understand the reasons for formatting, and if you understand one successful method for structuring programs, it shouldn’t be too difficult to adapt.

**Reasons for structuring programs with a strict and 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 in Lesson Twelve will provide you with this help. You’ll be able to use our given formats as a crutch until you (eventually) have them memorized.



**Lesson 12****Four Types Of Program Format**

*The formats shown in this lesson 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 structure.*

You know the reasons why programs must be formatted using a strict structure. In Lesson Twelve, we're going to show the actual formats. We will show two sets of format, one for use when geometry offsets are used to assign program zero – the other for use when program zero is assigned in the program. We will also explain every word in each format in detail. And we'll show example programs that stress the use of our given formats. Finally, we'll show some efficiency-related limitations of our given formats.

There are four types of program format:

- Program startup format

- Tool ending format

- Tool startup format

- Program ending format

Any time you begin writing a new program, follow the program startup format. 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 format, you write the motions for the first tool's machining operations. When finished with the first tool motions, you follow the tool ending format. You then follow the tool startup format for the second tool and write the second tool's machining motion commands. From this point, you toggle among tool ending format, tool startup format, and machining motion commands until you are finished machining with the last tool. You then follow the program ending format.

One of the most important benefits of using these formats is that you *will not have to memorize anything*. You simply copy the structure of the format.

Remember that you should use the formats related to assigning program zero with geometry offsets unless your machine does not have geometry offsets.

## Key Concept

## 6

## Special Features That Help With Programming

*You now have the basic tools you need to write CNC programs. However, writing programs with only the CNC features we have shown will be very tedious. In Key Concept Number Six, we'll be showing features that make programming easier, shorten the program's length, and in general, facilitate your ability to write programs.*

Key Concept Number Six contains seven lessons:

- 13: One-pass canned cycles
- 14: G71 and G70 – multiple repetitive cycles: rough turning and boring followed by finishing
- 15: G72-G75 – other multiple repetitive cycles
- 16: G76 – threading multiple repetitive cycle
- 17: Working with subprograms
- 18: Other special features of programming
- 19: Special machine types and accessories

This key concept should be of great interest to you. While you have learned how to write simple CNC programs, you may be wondering if there are easier ways to get things done. As you'll see in this key concept, there are.

You know, for example, that programming any kind of roughing operation (rough turning, rough facing, or rough boring) can be quite tedious. With only G00, G01, G02, and G03 to work with, every roughing pass will require at least four commands. Consider rough turning a workpiece down from 8.0 inches to 2.0 inches in diameter, taking a depth-of-cut of 0.125 inches. This will require twenty-four passes – at least ninety-six commands. And if a roughing pass intersects a chamfer, taper, or radius, it will be difficult to calculate the end point for the roughing pass. You will learn in Lesson Fourteen that as long as you can program the finish pass, the machine will completely rough turn or bore the workpiece for you – based on one simple command – regardless of how many passes are required. In Lesson Fifteen, you'll see that there is a similar command for rough facing.

In Lesson Sixteen, we'll present a simple way to machine threads – one command will machine the entire thread regardless of how many passes it takes. In Lesson Seventeen, you'll learn how to use a program-shortening feature called *sub-programming*. In Lesson Eighteen, we'll present several special features that can help you create programs. While they won't all be of immediate use, you'll surely find at least some of them to be very helpful.

We also include how to handle special machines and machine accessories in Lesson Nineteen. For instance, live tooling, bar feeders, and tailstocks will be presented.

Almost every CNC function presented thus far is *essential* to your ability to program CNC turning centers. You will be using the features and techniques presented in Key Concepts One through Five in every program you write. Think of what you have learned so far as the rudimentary tools of CNC programming.

**Lesson 13****One-Pass Canned Cycles**

*These cycles have been replaced with multiple repetitive cycles. While they are still available – and they can sometimes be helpful – you'll probably find little use for them. But for the sake of completeness, we include them in this text.*

There are two types of canned cycles equipped on Fanuc and Fanuc-compatible turning centers. The first, one-pass canned cycles, are quite limited. As the name implies, they will simply *make one pass* (turning, boring, facing, or threading). While there are some good applications for these cycles and you should understand their use, there is a much more helpful series of canned cycles called *multiple repetitive cycles*. These cycles are much better than one-pass canned cycles, especially when it comes to roughing operations (rough turning, rough facing, or rough boring) and threading. Some multiple repetitive cycles actually rival what a good computer aided manufacturing (CAM) system can do when it comes to simplifying a programmer's work.

**Cycle consistencies**

There are three things that *all* canned cycles have in common (including one-pass canned cycles and multiple repetitive cycles).

- 1) Every canned cycle requires you to first send the tool to a *convenient starting position*. The convenient starting position will vary from one cycle to the next, but it will always be the tool's current position in X and Z just prior to the canned cycle command. It is the tool's rapid approach position for the canned cycle.
- 2) At the completion of any canned cycle, *the tool will be left back at the convenient starting position*.
- 3) All canned cycles can be used for both internal diameter machining (boring) as well as external diameter work (turning). Though the convenient starting position will change from external to internal work, all points about the canned cycle will apply to either.

One-pass canned cycles allow you to program one pass of a machining operation, including turning or boring, facing, and threading. This will minimize the number of commands required in the program since they cause the tool to make four motions from one command. One-pass canned cycles are all *modal*, remaining in effect until another motion type is programmed (like G00 or G01). Additionally, each one-pass canned cycle requires a feedrate, specifying the machining rate to be used for machining during the cycle.

**Lesson 14****G71 And G70 – Rough Turning And Boring Followed By Finishing**

*These cycles are extremely helpful. They allow you to completely rough turn or rough bore a workpiece based upon a finish pass definition and one special command. In this command you specify important information like depth-of-cut and the amount of stock you want to leave for finishing. Based upon this simple information, the machine will completely rough the workpiece—regardless of the number of passes required.*

You know that one-pass canned cycles can be used to minimize the number of commands needed to machine a workpiece. But as you have seen, they are somewhat cumbersome to use. By comparison, the multiple repetitive cycles we discuss in this lesson are *much* more helpful. They make it relatively simple to program complex rough turning and rough boring operations for even the most difficult workpieces. As you'll see, they rival what can be done with a computer aided manufacturing (CAM) system.

As mentioned earlier, first we will show how these multiple repetitive cycles are programmed for the majority of Fanuc turning center controls (with one-line multiple repetitive cycles). Then we will discuss the slight differences required for OTA, OTB, 3T, and 18T Fanuc controls (with two-line multiple repetitive cycles).

**G71 - Rough turning and boring**

If you've been doing the programming exercises in this text and in the workbook, you've seen that rough turning and rough boring operations can be very tedious and error prone to program with only the three basic motion types (rapid, straight line, and circular). Every rough turning or boring pass requires at least four commands. And in our programming exercises so far, we've kept things as simple as possible. In reality, roughing a workpiece is usually much more difficult.

For example, all of the roughing passes you've programmed so far end at a known position (at a face or diameter). 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.

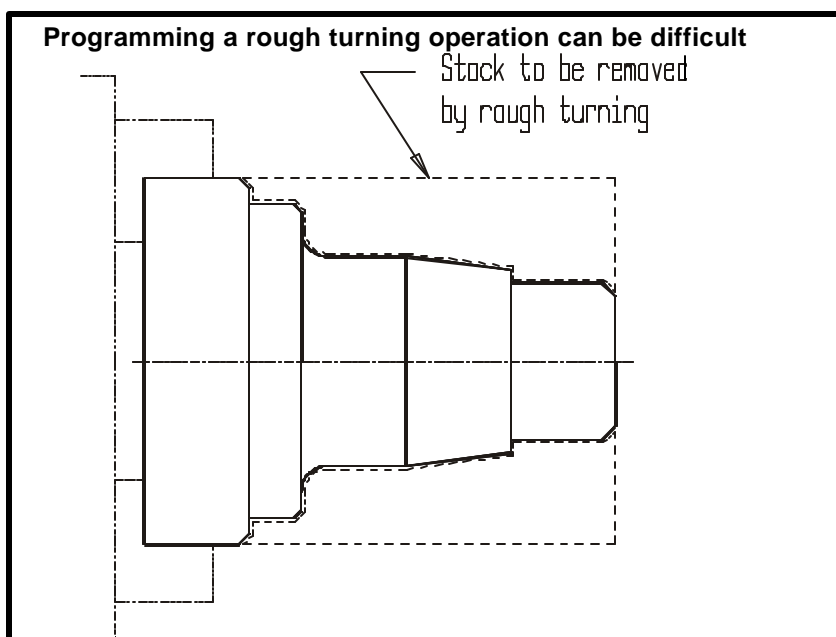


Figure 6.6 – Drawing that stresses the difficulty of programming roughing passes long-hand

**Lesson 15****G72-G75 – Other Multiple Repetitive Cycles**

*The rough facing cycle (G72) works very much like G71 – but roughing is done by facing instead of turning. If your workpiece is large in diameter and short in length, it makes more sense to rough face than to rough turn. The pattern repeating cycle (G73), the peck drilling cycle (G74) and the grooving cycle (G75) are not so helpful. We'll explain why in this lesson.*

In this lesson, we cover four more multiple repetitive cycles. But frankly speaking, you'll probably find that – of these four – only the G72 rough facing multiple repetitive cycle is very helpful.

**G72 - Rough facing**

There are times when it is not efficient to rough machine in a turning manner. When a workpiece is large in diameter and relatively short in length, it is usually better to rough machine in a facing manner since fewer passes will be required. G72 can be used for this purpose.

The G72 command is very similar to the G71 command. Here we will discuss *only the differences*. See Lesson Fourteen for more about how G71 works.

First, the convenient starting position is in a different location with G72. When using G72, the convenient starting position must be a position in X that clears the outside diameter of the raw material. In Z, it will be flush with the face of the raw material. (Point one in the figure 6.12 shows the convenient starting position for G72.) Second, the command specified by the P word must now include only a Z departure (no X word) along with a G00 or G01 (normally G00). Third, the finish pass definition must be done in a facing manner.

That's it. All words in the G72 mean exactly the same thing as they do in G71. And G70 can still be used for finishing. Knowing these differences, you should be able to understand this example program. Figure 6.12 shows the workpiece drawing along with the points needed for programming.

**Lesson 16****G76 Threading Command**

*This command is extremely helpful for machining threads. It machines external threads and internal threads with equal ease. Only one command is needed to machine the thread – regardless of how many passes are required.*

Threading involves *chasing* a thread on the workpiece with a single point threading tool. By chasing, we mean the threading tool will make a series of synchronized passes to machine the thread to its required depth. Thread depth will increase with each pass. The threading tool has the form of the thread ground into its insert.

G76 is an extremely helpful multiple repetitive cycle that will machine an entire thread in one command. Like the other multiple repetitive cycles, the G76 command allows you to specify the criteria for the thread to be machined with a series of *variables* that are included right in the G76 command.

Like the G71 command, G76 can be used for outside diameters as well as inside diameters. Also like the G71, the programmer will rapid the threading tool to a *convenient starting position*. This position for threading will be clear of the surface to be threaded in the X axis (by at least 0.03 inch or so), and well off the face of the diameter to be threaded in the Z axis (we recommend a distance of four times the pitch or 0.2 inch, *whichever is smaller*). This large approach distance in Z is required to allow the threading tool to accelerate to the required motion rate (feedrate) before contacting the workpiece. If it does not, the thread's pitch will not be correct at the beginning of the thread.

Figure 6.17 shows the words related to the G76 threading cycle. Note that an external thread is being shown.

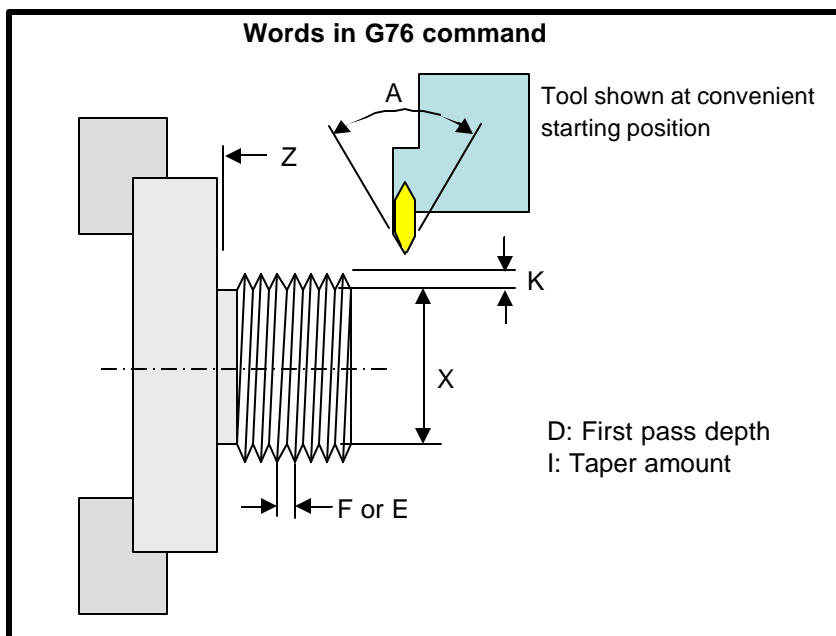


Figure 6.17 – Words related to G76 command

**X word**

The X word in a G76 command specifies the *final threading diameter*. For external threads, the X word is the minor diameter of the thread. For internal threads, the X word is the major diameter of the thread.

**Lesson 17****G76 Working With Subprograms**

*There are times when a series of CNC commands must be repeated – within one program – and sometimes among several programs. Whenever you find yourself writing a series of commands a second time – you should consider using a subprogram. The longer the series of commands and the more often they must be repeated, the more a subprogram can help.*

You know that a CNC turning center will execute a program in sequential order. It will start with the first command in the program: read it – interpret it – and execute it. Then it will move on to the next command – read, interpret, execute. It will continue this process for the entire program.

You also know that there are times when a series of commands in your program must be repeated. We've introduced one time so far – when multiple identical grooves must be machined.

In this lesson, you will learn about a way to change the order of program execution to some extent – and this will be especially helpful when commands must be repeated.

**The difference between main- and sub- programs**

A *main program* is the program that a setup person or operator will execute when they activate a cycle. *Every program shown to this point in the text has been a main program.* Main programs almost always end with M30 (or M02 with some machines).

A *subprogram* is a program that is invoked by a main program or by another subprogram. A subprogram ends with M99.

In a main program, when you reach a series of commands that you know will be repeated, you can invoke a subprogram that contains the repeated commands (you place the repeated commands in the subprogram instead of in the main program). Each time the commands must be executed, you will invoke the subprogram.

M98 is the word used to invoke a subprogram. A *P word* in the M98 command specifies the program number of the subprogram to be executed. When the machine is finished executing the subprogram, it will come back to the main program to the command after the calling M98.

Subprograms provide two important benefits – one rather obvious one and the other not so obvious. The obvious benefit is that subprograms will minimize the number of commands that must be specified in order to machine a workpiece. So programming becomes simpler and programs become shorter.

A second, less obvious, benefit of using a subprogram has to do with program verification. If the subprogram is correct the first time it is executed, it will be correct *every time* it is executed. Consider machining multiple identical grooves. When you independently program each groove (without using a subprogram), you can make a mistake at any time – when either writing or typing the program. When verifying the program, you must be careful with every command in every groove – again, a mistake can be made at any time.

But when you use a subprogram, the commands to machine one of the grooves will be included in the subprogram. If the first groove comes out correctly, so will all the others. *The same commands are being executed for every groove.*

**Loading multiple programs**

If you're not careful, using subprograms will require more time and effort on the setup person's part. Any programming time gained by using subprograms may be lost when setups are made. And of course, setup

**Lesson 18****Other Special Programming Features**

*Current model CNC turning centers come with many features to help with special applications. While some of these features will be of little need to you in the immediate future, it is important to know they exist. You cannot begin to apply any feature that you don't know about.*

CNC control manufacturers strive to equip their controls with as many helpful programming features as possible. Those mentioned so far (multiple repetitive cycles and subprograms) are used on a very regular basis — and you should strive to master them. However, there are some special programming features that are not used nearly as regularly. Indeed, there are some features that are extremely important to one company but of no value to another.

As you study this lesson, you need to consider your own company's CNC applications. If you are in doubt about the value of a given feature, ask an experienced person in your company about its value to your company. You can minimize your studies about those features your company does not currently need. You can always come back and study this lesson in greater detail should the need arise.

As you study this lesson, remember that *your ingenuity is based predominantly upon your knowledge of what is possible*. You cannot apply a feature of which you are unaware. At the very least, this lesson will acquaint you with what is possible with special CNC programming features.

The organization of this lesson is not as tutorial as previous lessons. While we will explain each feature in detail and in tutorial format, we don't present them in a special order. Here are the topics contained in this lesson:

- ✍ Block delete (also called optional block skip)
- ✍ Special techniques with sequence numbers
- ✍ G codes that have not yet been introduced (in numerical order)

**Block delete (also called optional block skip)**

The block delete function is used to give the CNC operator a choice between one of two possibilities. An on/off switch on the control panel (commonly labeled block delete or optional block skip) is used to actually make the choice. Since applications for block delete vary, the programmer must make each use of block delete *very clear* to the operator. This should be done in the setup- and production-run-documentation.

A slash code (/) in the program tells the control to look to the position of the block delete switch. If the switch is on, the control will skip any words to the right of the slash code. If the switch is off, the control will execute these words.

Here is a simple example. Say your turning center does not have an adequate coolant switch. Your setup person has no way to turn off the coolant when programs are being verified (whenever an M08 is executed, coolant will come on). To solve this problem, you can place a slash code at the beginning of every coolant command. Here is one way to do so.

/N015 M08 (If the block delete switch is on, the M08 will be skipped and coolant will stay off)

During program verification, the setup person can turn *on* the block delete switch. This will force the control to *skip* commands that turn coolant on (leaving the coolant off).

The slash code does not have to be placed at the very beginning of a command. If, for example, you have a totally enclosed work area (as most turning centers do), you may want the coolant to come on during each cutting tool's approach to the workpiece. Consider this command.



**Lesson 19****Special Machine Types And Accessories**

*To this point, we've limited our presentations to programming methods for basic two axis turning centers. In this lesson, we'll be showing how to handle the programming of special machines as well as special accessory devices.*

As you know, turning centers come in many configurations – and they can be equipped with many special accessories that change the way programs must be written. If you'll be working with a basic two axis turning center, you may be able to skip much of this appendix. Just skim this material to see what's here. You can always come back and read more thoroughly should the need arise. Be sure to check with an experienced person in your company to find out which accessories and machine types you must study.

All accessories and machine types discussed in this appendix require that you have a firm understanding of what has been presented thus far. In all cases, we'll be building upon what you already know.

Unfortunately, the actual programming and usage of certain accessories is highly machine-builder-dependent. While we can easily explain the application for a given device, it will be impossible to show specific usage techniques for all turning centers. In these cases, you'll at least be exposed to one builder's way of handling the accessory's usage.

**Work holding and work support devices**

How a workpiece is held and supported during its turning center machining operations is one of the most critical factors contributing to the success of the turning center application. If a workpiece is held and supported properly, all other facets of the machining operation will be easy to control. But an improperly held and/or supported workpiece will, at best, make the machining operation difficult to perform. In worse cases, an inadequate work holding and/or support device will make it impossible to hold size on the workpiece. That is, you won't be able to make good workpieces. And at worst, a poor selection of workholding and support devices can make for a very dangerous working environment.

In this section we will take a close look at the most common work holding and support devices available for use with CNC turning centers. While we cannot possibly discuss every detail and implication of each device, this presentation should give you a reasonably well rounded-view of the devices available.

**Work holding devices**

During any turning operation, the workpiece will be under two kinds of stress. One cause of workpiece stress is the centrifugal force generated with workpiece rotation. At the high spindle speeds available from today's turning centers, an out-of-round workpiece will have the tendency to shake itself apart. The second cause of workpiece stress is caused by the machining operation itself. As any cutting tool comes into contact with the workpiece, it can have the tendency to push the workpiece out of the workholding device.

During the typical purchase of a turning center, often too little consideration is given to possible work holding and work support alternatives. And, as stated, a poorly chosen work holding and/or work support device will make for a poor machining environment. In this section, you will be exposed to the most common work holding alternatives, how they are used, and for what kinds of work they are best.

Try not to view a turning center's work holding device as an integral part of the machine tool itself. Rather, try to view it as only part of the machine's setup, just as a common table vise is only a part of a vertical machining center's setup. Though a table vise is a very common work holding tool, it will not properly hold all workpieces to be machined by a vertical machining center. Just as a table vise will not suffice for all machining center setups, no single turning center work holding device will suffice for all kinds of workpieces that can be machined on a turning center.