



## LANDMARK UNIVERSITY, OMU-ARAN

**LECTURE NOTE: 1**

**COLLEGE: COLLEGE OF SCIENCE AND ENGINEERING**

**DEPARTMENT: MECHANICAL ENGINEERING**

**PROGRAMME: MECHANICAL ENGINEERING**

**ENGR. ALIYU, S.J**

*Course code: MCE 314*

*Course title: WORKSHOP PRACTICE*

*Credit unit: 2 UNITS.*

*Course status: compulsory*

### **PART 2**

## **The Basics of Computer Numerical Control.**

Today, computer numerical control (CNC) machines are found almost everywhere, from small job shops in rural communities to Fortune 500 companies in large urban areas. Truly, there is hardly a facet of manufacturing that is not in some way touched by what these innovative machine tools can do. Everyone involved in the manufacturing environment should be well aware of what is possible with these sophisticated machine tools. The design engineer, for example, must possess enough knowledge of CNC to perfect dimensioning and tolerancing techniques for work-pieces to be machined on CNC machines. The tool engineer must understand CNC in order to design fixtures and cutting tools for use with CNC machines. Quality control people should understand the CNC machine tools used within their company in order to plan quality control and statistical process control accordingly. Production control personnel should be abreast of their company's CNC technology in order to make realistic production schedules. Managers, foremen, and team leaders should understand CNC well enough to communicate intelligently with fellow workers. And, it goes without saying that CNC programmers, setup people, operators, and others working directly with the CNC equipment must have an extremely good understanding of CNC. In this course, we will explore the basics of CNC, showing you much of what is involved with using these sophisticated machine tools. The primary goal will be to teach you how to learn about CNC. As a student who will eventually be working directly with CNC machine tools, you need to know the basics of each major CNC function. Additionally, more suggestions are made to how you can learn more about each CNC function as it applies to a particular CNC machine/s. At the completion of this course, you should have a good understanding of how and why CNC functions as it does and know those things you must learn more about in order to work with any style of CNC machine tool. For students who are not going to be working directly with CNC equipment in the near future, the secondary goal will be to give you a good working knowledge of CNC technology. At the completion of this

course, you should be quite comfortable with the fundamentals of CNC and be able to communicate intelligently with others in your company about your CNC machine tools. To proceed in an organized manner, a key concepts approach will be used. Five important functions of CNC programming concepts will be discussed. Think of it this way. If you can understand ten basic principles of CNC, you are well on your way to becoming proficient with CNC. While our main focus will be for the two most popular forms of CNC machine tools (machining centers and turning centers), these ten key concepts can be applied to virtually any kind of CNC machine, making it easy to adapt to any form of CNC equipment. With so many types of CNC machine tools in existence, it is next to impossible for this course to be extremely specific about any one particular type. The key concepts allow us to view the main features of CNC in more general terms, stressing why things are handled the way they are even more than the specific techniques used with any one particular CNC machine tool. With the broad background, you should be able to easily zero in on any kind of CNC machine tool you will be working with.

SECTION ONE.

## **Fundamentals of CNC.**

While the specific intention and application for CNC machines vary from one machine type to another, all forms of CNC have common benefits. Though the thrust of this course is to teach you CNC usage, it helps to understand why these sophisticated machines have become so popular. Here are but a few of the more important benefits offered by CNC equipment.

The first benefit offered by all forms of CNC machine tools is improved automation. The operator intervention related to producing work-pieces can be reduced or eliminated. Many CNC machines can run unattended during their entire machining cycle, freeing the operator to do other tasks. This gives the CNC user several side benefits including reduced operator fatigue, fewer mistakes caused by human error, and consistent and predictable machining time for each work-piece. Since the machine will be running under program control, the skill level required of the CNC operator (related to basic machining practice) is also reduced as compared to a machinist producing work-pieces with conventional machine tools. The second major benefit of CNC technology is consistent and accurate work-pieces. Today's CNC machines boast almost unbelievable accuracy and repeatability specifications. This means that once a program is verified, two, ten, or one thousand identical work-pieces can be easily produced with precision and consistency. A third benefit offered by most forms of CNC machine tools is flexibility. Since these machines are run from programs, running a different work-piece is almost as easy as loading a different program. Once a program has been verified and executed for one production run, it can be easily recalled the next time the work-piece is to be run. This leads to yet another benefit, fast change-overs. Since these machines are very easy to setup and run, and since programs can be easily loaded, they allow very short setup time. This is imperative with today's Just-In-Time product requirements.

Motion control - the heart of CNC

The most basic function of any CNC machine is automatic, precise, and consistent motion control. Rather than applying completely mechanical devices to cause motion as is required on

most conventional machine tools, CNC machines allow motion control in a revolutionary manner. All forms of CNC equipment have two or more directions of motion, called axes. These axes can be precisely and automatically positioned along their lengths of travel. The two most common axis types are linear (driven along a straight path) and rotary (driven along a circular path). Instead of causing motion by turning cranks and hand-wheels as is required on conventional machine tools, CNC machines allow motions to be commanded through programmed commands. Generally speaking, the motion type (rapid, linear, and circular), the axes to move, the amount of motion and the motion rate (feed-rate) are programmable with almost all CNC machine tools. Accurate positioning is accomplished by the operator counting the number of revolutions made on the hand-wheel plus the graduations on the dial. The drive motor is rotated a corresponding amount, which in turn drives the ball screw, causing linear motion of the axis. A feedback device confirms that the proper amount of ball screw revolutions have occurred. A CNC command executed within the control (commonly through a program) tells the drive motor to rotate a precise number of times. The rotation of the drive motor in turn rotates the ball screw. And the ball screw causes drives the linear axis. A feedback device at the opposite end of the ball screw allows the control to confirm that the commanded number of rotations has taken place. Though a rather crude analogy, the same basic linear motion can be found on a common table vise. As you rotate the vise crank, you rotate a lead screw that, in turn, drives the movable jaw on the vise. By comparison, a linear axis on a CNC machine tool is extremely precise. The number of revolutions of the axis drive motor precisely controls the amount of linear motion along the axis. How axis motion is commanded - understanding coordinate systems it would be infeasible for the CNC user to cause axis motion by trying to tell each axis drive motor how many times to rotate in order to command a given linear motion amount. (This would be like having to figure out how many turns of the handle on a table vise will cause the movable jaw to move exactly one inch!) Instead, all CNC controls allow axis motion to be commanded in a much simpler and more logical way by utilizing some form of coordinate system. The two most popular coordinate systems used with CNC machines are the rectangular coordinate system and the polar coordinate system. By far, the most popular of these two is the rectangular coordinate system, and we'll use it for all discussions made during this course.

One very common application for the rectangular coordinate system is graphing. Almost everyone has had to make or interpret a graph. Since the need to utilize graphs is so commonplace, and since it closely resembles what is required to cause axis motion on a CNC machine, let's review the basics of graphing. As with any two dimensional graph, this graph has two base lines. Each base line is used to represent something. What the base line represents is broken into increments. Also, each base line has limits. In our productivity example, the horizontal base line is being used to represent time. For this base line, the time increment is in months. Remember this base line has limits - it starts at January and end with December. The vertical base line is representing productivity. Productivity is broken into ten percent increments and starts at zero percent productivity and ends with one hundred percent productivity. The person making the graph would look up the company's productivity for January of last year and at the productivity position on the graph for January, a point is plotted. This would then be repeated for February, March, and each month of the year. Once all points are plotted, a line or curve can be drawn through each of the points to make it clearer as to how the company did last year. Let's take what we now know about graphs and relate it to CNC axis motion. Instead of

plotting theoretical points to represent conceptual ideas, the CNC programmer is going to be plotting physical end points for axis motions. Each linear axis of the machine tool can be thought of as like a base line of the graph. Like graph base lines, axes are broken into increments. But instead of being broken into increments of conceptual ideas like time and productivity, each linear axis of a CNC machine's rectangular coordinate system is broken into increments of measurement. In the inch mode, the smallest increment is usually 0.0001 inch. In the metric mode, the smallest increment is 0.001 millimeter. (By the way, for rotary axes the increment is 0.001 degrees.) Just like the graph, each axis within the CNC machine's coordinate system must start somewhere. With the graph, the horizontal baseline started at January and the vertical base line started at zero percent productivity. This place where the vertical and horizontal base lines come together is called the origin point of the graph. For CNC purposes, this origin point is commonly called the program zero point (also called work zero, part zero, and program origin). For this example, the two axes we happen to be showing are labeled as X and Y but keep in mind that program zero can be applied to any axis. Though the names of each axes will change from one CNC machine type to another (other common names include Z, A, B, C, U, V, and W), this example should work nicely to show you how axis motion can be commanded. The program zero point establishes the point of reference for motion commands in a CNC program. This allows the programmer to specify movements from a common location. If program zero is chosen wisely, usually coordinates needed for the program can be taken directly from the print. With this technique, if the programmer wishes the tool to be sent to a position one inch to the right of the program zero point, X1.0 is commanded. If the programmer wishes the tool to move to a position one inch above the program zero point, Y1.0 is commanded. The control will automatically determine how many times to rotate each axis drive motor and ball screw to make the axis reach the commanded destination point. This lets the programmer command axis motion in a very logical manner. With the examples given so far, all points happened to be up and to the right of the program zero point. This area up and to the right of the program zero point is called a quadrant (in this case, quadrant number one). It is not uncommon on CNC machines that end points needed within the program fall in other quadrants. When this happens, at least one of the coordinates must be specified as minus.

## **Understanding absolute versus incremental motion.**

All discussions to this point assume that the absolute mode of programming is used. The most common CNC word used to designate the absolute mode is G90. In the absolute mode, the end points for all motions will be specified from the program zero point. For beginners, this is usually the best and easiest method of specifying end points for motion commands. However, there is another way of specifying end points for axis motion. In the incremental mode (commonly specified by G91), end points for motions are specified from the tool's current position, not from program zero. With this method of commanding motion, the programmer must always be asking "How far should I move the tool?" While there are times when the incremental mode can be very helpful, generally speaking, this is the more cumbersome and difficult method of specifying motion and beginners should concentrate on using the absolute mode. Be careful when making motion commands. Beginners have the tendency to think incrementally. If working in the absolute mode (as beginners should), the programmer should always be asking "To what position should the tool be moved?" This position is relative to program zero, NOT from the tool's current position. Aside from making it very easy to determine the current position for any

command, another benefit of working in the absolute mode has to do with mistakes made during motion commands. In the absolute mode, if a motion mistake is made in one command of the program, only one movement will be incorrect. On the other hand, if a mistake is made during incremental movements, all motions from the point of the mistake will also be incorrect.

## **Assigning program zero**

Keep in mind that the CNC control must be told the location of the program zero point by one means or another. How this is done varies dramatically from one CNC machine and control to another. One (older) method is to assign program zero in the program. With this method, the programmer tells the control how far it is from the program zero point to the starting position of the machine. This is commonly done with a G92 (or G50) command at least at the beginning of the program and possibly at the beginning of each tool. Another, newer and better way to assign program zero is through some form of offset. Commonly machining center control manufacturers call offsets used to assign program zero fixture offsets. Turning center manufacturers commonly call offsets used to assign program zero for each tool geometry offsets. More on how program zero can be assigned will be presented during key concept number four.

Other points about axis motion.

To this point, our primary concern has been to show you how to determine the end point of each motion command. As you have seen, doing this requires an understanding of the rectangular coordinate system. However, there are other concerns about how a motion will take place. For example, the type of motion (rapid, straight line, circular, etc.), and motion rate (feed-rate), will also be of concern to the programmer. We'll discuss these other considerations later.

Telling the machine what to do - the CNC program.

Almost all current CNC controls use a word address format for programming. (The only exceptions to this are certain conversational controls.) By word address format, we mean that the CNC program is made up of sentence-like commands. Each command is made up of CNC words. Each CNC word has a letter address and a numerical value. The letter address (X, Y, Z, etc.) tells the control the kind of word and the numerical value tells the control the value of the word. Used like words and sentences in the English language, words in a CNC command tell the CNC machine what it is we wish to do at the present time. One very good analogy to what happens in a CNC program is found in any set of step by step instructions. Say for example, you have some visitors coming in from out of town to visit your company. You need to write down instructions to get from the local motor-park or airport to your company. To do so, you must first be able to visualize the path from the arrive park or airport to your company. You will then, in sequential order, write down one instruction at a time. The person following your instructions will perform the first step and then go on to the next until he or she reaches your facility. In similar manner, a manual CNC programmer must be able to visualize the machining operations that are to be performed during the execution of the program. Then, in step by step order, the programmer will give a set of commands that makes the machine behave accordingly. Though

slightly off the subject at hand, we wish to make a strong point about visualization. Just as the person developing travel directions **MUST** be able to visualize the path taken, so **MUST** the CNC programmer be able to visualize the movements the CNC machine will be making **BEFORE** a program can be successfully developed. Without this visualization ability, the programmer will not be able to develop the movements in the program correctly. This is one reason why machinists make the best CNC users. An experienced machinist should be able to easily visualize any machining operation taking place. Just as each concise travel instruction will be made up of one sentence, so will each instruction given within a CNC program be made up of one command. Just as the travel instruction sentence is made up of words (in English), so is the CNC command made up of CNC words (in CNC language). The person following your set of travel instructions will execute them explicitly. If you make a mistake with your set of instructions, the person will get lost on the way to your company. In similar fashion, the CNC machine will execute a CNC program explicitly. If there is a mistake in the program, the CNC machine will not behave correctly.

Program:

O0001 (Program number)

N005 G54 G90 S400 M03 (Select coordinate system, absolute mode, and turn spindle on CW at 400 RPM)

N010 G00 X1. Y1. (Rapid to XY location of first hole)

N015 G43 H01 Z.1 M08 (Instate tool length compensation, rapid in Z to clearance position above surface to drill, turn on coolant)

N020 G01 Z-1.25 F3.5 (Feed into first hole at 3.5 inches per minute)

N025 G00 Z.1 (Rapid back out of hole) N030 X2. (Rapid to second hole)

N035 G01 Z-1.25 (Feed into second hole)

N040 G00 Z.1 M09 (Rapid out of second hole, turn off coolant)

N045 G91 G28 Z0 (Return to reference position in Z)

N050 M30 (End of program command)

While the words and commands in this program probably do not make much sense to you (yet), remember that we are stressing the sequential order by which the CNC program will be executed. The control will first read, interpret and execute the very first command in the program. Only then will it go on to the next command. Read, interpret, execute. Then on to the next command. The control will continue to execute the program in sequential order for the balance of the program. Again, notice the similarity to giving any set of step by step instructions.

Other notes about program makeup.

As stated programs are made up of commands and commands are made up of word. Each word has a letter address and a numerical value. The letter address tells the control the word type. CNC control manufacturers do vary with regard to how they determine word names (letter addresses) and their meanings. The beginning CNC programmer must reference the control manufacturer's programming manual to determine the word names and meanings. Here is a brief list of some of the word types and their common letter address specifications.

- O - Program number (Used for program identification)
- N - Sequence number (Used for line identification)
- G - Preparatory function
- X - X axis designation
- Y - Y axis designation
- Z - Z axis designation
- R - Radius designation
- F - Feed-rate designation
- S - Spindle speed designation
- H - Tool length offset designation
- D - Tool radius offset designation
- T - Tool Designation
- M - Miscellaneous function (See below)

As you can see, many of the letter addresses are chosen in a rather logical manner (T for tool, S for spindle, F for feed-rate, etc.). A few require memorizing.

There are two letter addresses (G and M) which allow special functions to be designated. The preparatory function (G) specifies is commonly used to set modes. We already introduced absolute mode, specified by G90 and incremental mode, specified by G91. These are but two of the preparatory functions used. You must reference your control manufacturer's manual to find the list of preparatory functions for your particular machine. Like preparatory functions, miscellaneous functions (M words) allow a variety of special functions. Miscellaneous functions are typically used as programmable switches (like spindle on/off, coolant on/off, and so on). They are also used to allow programming of many other programmable functions of the CNC machine tool. To a beginner, all of this may seem like CNC programming requires a great deal of memorization. But rest assured that there are only about 30-40 different words used with CNC programming. If you can think of learning CNC manual programming as like learning a foreign language that has only 40 words, it shouldn't seem too difficult.

### **Decimal point programming**

Certain letter addresses (CNC words) allow the specification of real numbers (numbers that require portions of a whole number). Examples include X axis designator (X), Y axis designator (Y), and radius designator (R). Almost all current model CNC controls allow a decimal point to be used within the specification of each letter address requiring real numbers. For example, X3.0625 can be used to specify a position along the X axis. On the other hand, some letter addresses are used to specify integer numbers. Examples include the spindle speed designator

(S), the tool station designator (T), sequence numbers (N), preparatory functions (G), and miscellaneous functions (M). For these word types, most controls do NOT allow a decimal point to be used. The beginning programmer must reference the CNC control manufacturer's programming manual to find out which words allow the use of a decimal point.

Other programmable functions.

All but the very simplest CNC machines have programmable functions other than just axis motion. With today's full blown CNC equipment, almost everything about the machine is programmable. CNC machining centers, for example, allow the spindle speed and direction, coolant, tool changing, and many other functions of the machine to be programmed. In similar fashion, CNC turning centers allow spindle speed and direction, coolant, turret index, and tailstock to be programmed. And all forms of CNC equipment will have their own set of programmable functions. Additionally, certain accessories like probing systems, tool length measuring systems, pallet changers, and adaptive control systems may also be available that require programming considerations. The list of programmable functions will vary dramatically from one machine to the next, and the user must learn these programmable functions for each CNC machine to be used. In the next lecture (Know your machine), we will take a closer look at what is typically programmable on different forms of CNC machine tools.

## **SECTION TWO**

### **Know your machine.**

A CNC user MUST understand the makeup of the CNC machine tool being utilized. While this may sound like a basic statement, a CNC user must be able to view the machine from two distinctly different perspectives. Here in this section, we will be viewing the machine from a programmer's perspective. Much later, in section seven, we will look at the machine from an operator's viewpoint.

#### **Basic machining practice - the key to success with any CNC machine**

Many forms of CNC machines are designed to enhance or replace what is currently being done with more conventional machines. The first goal of any CNC beginner should be to understand the basic machining practice that goes into using the CNC machine tool. The more the beginning CNC user knows about basic machining practice, the easier it will be to adapt to CNC. Think of it this way. If you already know basic machining practice as it relates to the CNC machine you will be working with, you already know what it is you want the machine to do. It will be a relatively simple matter of learning how to tell the CNC machine what it is you want it to do (learning to program). This is why machinists make the best CNC programmers, operators, and setup personnel. Machinists already know what it is the machine will be doing. It will be a relatively simple matter of adapting what they already know to the CNC machine. For example, a beginner to CNC turning centers should understand the basic machining practice related to turning operations like rough and finish turning, rough and finish boring, grooving, threading, and necking. Since this form of CNC machine can perform multiple operations in a single

program (as many CNC machines can), the beginner should also know the basics of how to process work-pieces machined by turning so a sequence of machining operations can be developed for work-pieces to be machined. This point cannot be overstressed. Trying to learn about a particular CNC machine without understanding the basic machining practice related to the machine would be like trying to learn how to fly an airplane without understanding the basics of aerodynamics and flight. Just as a beginning pilot will be in for a great number of problems without understanding aerodynamics, so is the beginning CNC user have difficulty learning how to utilize CNC equipment without an understanding of basic machining practice.

## **Learning about a new CNC machine - the key points.**

From a programmer's standpoint, as you begin to learn about any new CNC machine, you should concentrate on four basic areas. First, you should understand the machine's most basic components. Second, you should become comfortable with your machine's directions of motion (axes). Third, you should become familiar with any accessories equipped with the machine. And fourth, you should find out what programmable functions are included with the machine and learn how they are programmed.

## **Machine components.**

While you do not have to be a machine designer to work with CNC equipment, it is important to know how your CNC machine is constructed. Understanding your machine's construction will help you to gauge the limits of what is possible with your machine. Just as the race car driver should understand the basics of suspension systems, breaking systems, and the workings of internal combustion engines (among other things) in order to get the most out of a given car, so must the CNC programmer understand the basic workings of the CNC machine in order to get the most from the CNC machine tool. For a universal style slant bed turning center, for example, the programmer should know the most basic machine components, including bed, way system, headstock & spindle, turret construction, tailstock, and work holding device. Information regarding the machine's construction including assembly drawings is usually published right in the machine tool builder's manual. As you read the machine tool builder's manual, here are some of the machine capacity and construction questions to which you should find answers.

- What is the machine's maximum RPM?
- How many spindle ranges does the machine have (and what are the cut-off points for each range?)
- What is the spindle and axis drive motor horsepower?
- What is the maximum travel distance in each axis?
- How many tools can the machine hold?
- What way construction does the machine incorporate (usually square ways, dovetail, and/or linear bearing ways)?
- What is the machine's rapid rate (fastest traverse rate)?
- What is the machine's fastest cutting feed-rate?

These are but a few of the questions you should be asking yourself as you begin working with any new CNC machine. Truly, the more you know about your machine's capacity and construction, the easier it will be to get comfortable with the machine.

### **Directions of motion (axes).**

The CNC programmer MUST know the programmable motion directions (axes) available for the CNC machine tool. The axes names will vary from one machine tool type to the next. They are always referred to with a letter address. Common axis names are X, Y, Z, U, V, and W for linear axes and A, C, and C for rotary axes. However, the beginning programmer should confirm these axis designations and directions (plus and minus) in the machine tool builder's manual since not all machine tool builders conform to the axis names we show. As discussed in section one, whenever a programmer wishes to command movement in one or more axes, the letter address corresponding to the moving axes as well as the destination in each axis are specified. X3.5, for example tells the machine to move the X axis to a position of 3.5 inches from the program zero point in X (assuming the absolute mode of programming is used).

### **The reference point for each axis**

Most CNC machines utilize a very accurate position along each axis as a starting point or reference point for the axis. Some control manufacturers call this position the zero return position. Others call it the grid zero position. Yet others call it the home position. Regardless of what it is called, the reference position is required by many controls to give the control an accurate point of reference. CNC controls that utilize a reference point for each axis require that the machine be manually sent to its reference point in each axis as part of the power up procedure. Once this is completed, the control will be in sync with the machine's position.

### **Accessories to the machine/**

The third area a beginning CNC user should address is related to other possible additions to the basic machine tool itself. Many CNC machine tools are equipped with accessories designed to enhance what the basic machine tool can do. Some of these accessories may be made and supported by the machine tool builder. These accessories should be well documented in the machine tool builder's manual. Other accessories may be made by an after-market manufacturer, in which case a separate manual may be involved. Examples of CNC accessories include probing systems, tool length measuring devices, post process gauging systems, automatic pallet changers, adaptive control systems, bar feeders for turning centers, live tooling and C axis for turning centers, and automation systems. Truly, the list of potential accessory devices goes on and on.

### **Programmable functions.**

The programmer must also know what functions of the CNC machine are programmable (as well as the commands related to programmable functions). With low cost CNC equipment, often times many machine functions must be manually activated. With some CNC milling machines, for example, about the only programmable function is axis motion. Just about everything else

may have to be activated by the operator. With this type of machine, the spindle speed and direction, coolant and tool changes may have to be activated manually by the operator. With full blown CNC equipment, on the other hand, almost everything is programmable and the operator may only be required to load and remove work-pieces. Once the cycle is activated, the operator may be freed to do other company functions. Reference the machine tool builder's manual to find out what functions of your machine are programmable. To give you some examples of how many programmable functions are handled, here is a list a few of the most common programmable functions along with their related programming words.

### **Spindle control**

An "S" word is used to specify the spindle speed (in RPM for machining centers). An M03 is used to turn the spindle on in a clockwise (forward) manner. M04 turns the spindle on in a counter clockwise manner. M05 turns the spindle off. Note that turning centers also have a feature called constant surface speed which allows spindle speed to also be specified in surface feet per minute (or meters per minute)

### **Automatic tool changer (machining center)**

A "T" word is used to tell the machine which tool station is to be placed in the spindle. On most machines, an M06 tells the machine to actually make the tool change. Tool change (on turning centers) A four digit "T" word is used to command tool changes on most turning centers. The first two digits of the T word specify the turret station number and the second two digits specify the offset number to be used with the tool. T0101, for example specifies tool station number one with offset number one.

### **Coolant control**

M08 is used to turn on flood coolant. If available M07 is used to turn on mist coolant. M09 turns off the coolant.

### **Automatic pallet changer**

An M60 command is commonly used to make pallet changes.

### **Other programmable features to look into**

An M60 command is commonly used to make pallet changes.

As stated, programmable functions will vary dramatically from one machine to the next. The actual programming commands needed will also vary from builder to builder. Be sure to check the M codes list (miscellaneous functions) given in the machine tool builder's manual to find out more about what other functions may be programmable on your particular machine. M codes are commonly used by the machine tool builder to give the user programmable ON/OFF switches for machine functions. In any case, you must know what you have available for activating within your CNC programs. For turning centers, for example, you may find that the tailstock and

tailstock quill is programmable. The chuck jaw open and close may be programmable. If the machine has more than one spindle range, commonly the spindle range selection is programmable. And if the machine has a bar feeder, it will be programmable. You may even find that your machine's chip conveyor can be turned on and off through programmed commands. All of this, of course, is important information to the CNC programmer.

SECTION THREE.

## **You Must Understand The Motion Types Available On Your CNC Machine.**

During the first section, we discussed how end points for axis motion are commanded utilizing the rectangular coordinate system. During that presentation, however, we were only concerned with describing how the CNC machine determines the END POINT position for each motion. To effectively command motion on most CNC machines requires more than just specifying end points for positioning movements. CNC control manufacturers try to make it as easy as possible to make movement commands within the program. For those styles of motion that are commonly needed, they give the CNC user interpolation types.

### **Understanding interpolation.**

Say for example, you wish to move only one linear axis in a command. Say you wish to move the X axis to a position one inch to the right of program zero. In this case, the command X1.would be given (assuming the absolute mode is instated). The machine would move along a perfectly straight line during this movement (since only one axis is moving). Now let's say you wish to include a Y axis movement to a position one inch above program zero in Y (with the X movement). We'll say you are trying to machine a tapered or chamfered surface of your work-piece in this command. For the control to move along a perfectly straight line to get to the programmed end point, it must perfectly synchronize the X and Y axis movements. Also, if machining is to occur during the motion, a motion rate (feed-rate) must also be specified. This requires linear interpolation.

During linear interpolation commands, the control will precisely and automatically calculate a series of very tiny single axis departures, keeping the tool as close to the programmed linear path as possible. With today's CNC machine tools, it will appear that the machine is forming a perfectly straight line motion. However, Figure 3.1 shows what the CNC control is actually doing during linear interpolation. Figure 3.1 - Actual motion generated with linear interpolation. Notice the series of very tiny single axis movements. The step size is equal to the machine's resolution, usually 0.0001 in or 0.001 mm. In similar fashion, many applications for CNC machine tools require that the machine be able to form circular motions. Applications for circular motions include forming radii on turned work-pieces between faces and turns and milling radii on contours of machining center work-pieces. This kind of motion requires circular interpolation. As with linear interpolation, the control will do its best to generate as close to a circular path as possible.

## **Other interpolation types.**

Depending on the machine's application, you may find that you have other interpolation types available. Again, CNC control manufacturers try to make it as easy as possible to program their controls. If an application requires a special kind of movement, the control manufacturer can give the applicable interpolation type. For example, many machining center users perform thread milling operations on their machines. During thread milling, the machine must move in a circular manner along two axes (usually X and Y) at the same time a third axis (usually Z) moves in a linear manner. This allows the helix of the thread to be properly machined. This motion resembles a spiraling motion (though the radius of the spiral remains constant). Knowing that their customers need this type of motion for thread milling, CNC machining center control manufacturers offer the feature helical interpolation. With this feature, the user can easily command the motions necessary for thread milling.

## **The three most basic motion types.**

While your particular CNC machine may have more motion types (depending on your application), let's concentrate on becoming familiar with the three most common types of motion. These three motion types are available on almost all forms of CNC equipment. After briefly introducing each type of motion, an example program that stresses the use of all three will be shown. These motion types share two things in common. First, they are all modal. This means they remain in effect until changed. If for example, several motions of the same kind are to be given consecutively, the corresponding G-code need only be specified in the first command. Second, the END POINT of the motion is specified in each motion command. The current position of the machine will be taken as the starting point.

## **Rapid motion (also called positioning)**

This motion type (as the name implies) is used to command motion at the machine's fastest possible rate. It is used to minimize non-productive time during the machining cycle. Common uses for rapid motion include positioning the tool to and from cutting positions, moving to clear clamps and other obstructions, and in general, any non-cutting motion during the program. You must check in the machine tool builder's manual to determine a machine's rapid rate. Usually this rate is extremely fast (some machines boast rapid rates of well over 1000 RPM!), meaning the operator must be cautious when verifying programs during rapid motion commands. Fortunately, there is a way for the operator to override the rapid rate during program verification. The command almost all CNC machines use to command rapid motion is G00. Within the G00 Command, the end point for the motion is given. Control manufacturers vary with regard to what actually happens if more than one axis is included in the rapid motion command. With most controls, the machine will move as fast as possible in all axes commanded. In this case, one axis will probably reach its destination point before the other/s. With this kind of rapid command, straight line movement will NOT occur during rapid and the programmer must be very careful if there are obstructions to avoid. With other controls, straight line motion will occur, even during rapid motion commands.

## **Straight line motion (also called linear interpolation)**

This motion type allows the programmer to command perfectly straight line movements as discussed earlier during our discussion of linear interpolation. This motion type also allows the programmer to specify the motion rate (feed-rate) to be used during the movement. Straight line motion can be used any time a straight cutting movement is required, including when drilling, turning a straight diameter, face or taper, and when milling straight surfaces. The method by which feed-rate is programmed varies from one machine type to the next. Generally speaking, machining centers only allow the feed-rate to be specific in per minute format (inches or millimeters per minute). Turning centers also allow feed-rate to be specified in per revolution format (inches or millimeters per revolution). A G01 word is commonly used to specify straight line motion. Within the G01, the programmer will include the desired end point in each axis.

## **Circular motion (also called circular interpolation)**

This motion type causes the machine to make movements in the form of a circular path. As discussed earlier during the lecture of circular interpolation, this motion type is used to generate radii during machining. All feed-rate related points made during our discussion of straight line motion still apply.

Two, G-codes are used with circular motion. G02 is commonly used to specify clockwise motion while G03 is used to specify counter clockwise motion. To evaluate which to use, you simply view the movement from the same perspective the machine will view the motion. For example, if making a circular motion in XY on a machining center, simply view the motion from the spindle's vantage point. If making a circular motion in XZ on a turning center, simply view the motion from above the spindle. In most cases, this is as simple as viewing the print from above. Additionally, circular motion requires that, by one means or another, the programmer specifies the radius of the arc to be generated. With newer CNC controls this is handled by a simple "R" word. The R word within the circular command simply tells the control the radius of the arc being commanded. With older controls, directional vectors (specified by I, J, and K) tell the control the location of the arc's center point. Since controls vary with regard to how directional vectors are programmed, and since the R word is becoming more and more popular for radius designation, the examples will show the use of the R word. If you wish to learn more about directional vectors, you must reference your control manufacturer's manual.

## **Example program showing three types of motion.**

In this particular example, we are milling around the outside of a work-piece contour. Notice that we are using a one inch diameter end-mill for machining the contour and we are programming the very center of the end mill. Later, in the next section, how to actually program the work-piece contour (not the cutter center-line path) will be discussed. While you may not understand all commands given in this program, concentrate on understanding what is happening in the motion commands (G00, G01, and G02/G03). With study, you should be able to see what is happening. Messages in parentheses are provided to document what is happening in each command.

## Program

O0002 (Program number)

N005 G54 G90 S350 M03 (Select coordinate system, absolute mode, and start spindle CW at 350 RPM)

N010 G00 X-.625 Y-.25 (Rapid to point 1)

N015 G43 H01 Z-.25 (Instate tool length compensation, rapid tool down to work surface)

N020 G01 X5.25 F3.5 (Machine in straight motion to point 2)

N025 G03 X6.25 Y.75 R1.0 (CCW circular motion to point 3)

N030 G01 Y3.25 (Machine in straight motion to point 4)

N035 G03 X5.25 Y4.25 R1.0 (CCW circular motion to point 5)

N040 G01 X.75 (Machine in straight motion to point 6)

N045 G03 X-.25 Y3.25 R1.0 (CCW circular motion to point 7)

N050 G01 Y.75 ((Machine in straight motion to point 8)

N055 G03 X.75 Y-.25 R1.0 (CCW circular motion to point 9)

N060 G00 Z.1 (Rapid away from workpiece in Z)

N065 G91 G28 Z0 (Go to the machine's reference point in Z)

N070 M30 (End of program)

Keep in mind that CNC controls do vary with regard to limitations with motion types. For example, some controls have strict rules governing how much of a full circle you are allowed to make within one circular command. Some require directional vectors for circular motion commands instead of allowing the R word. Some even incorporate automatic corner rounding and chamfering, minimizing the number of motion commands that must be given. Though you must be prepared for variations, and you must reference your control manufacturer's programming manual to find out more about your machine's motion commands, at least this lecture has disclose to you the basics of motion commands and you should be able to adapt to your particular machine and control with relative ease.

## SECTION FOUR.

## **You must Understand the forms of compensation**

All types of CNC machine tools require some form/s of compensation. Though applied for different reasons on different machine types, all forms of compensation allow the CNC user to allow for unpredictable conditions related to tooling as the program is developed. Before discussing how compensation applies to CNC usage, let's look at compensation in general terms.

Compensation is used in many facets of everyday life. The airplane pilot must compensate for wind velocity and direction as a heading is set. The race car driver must compensate for weather and track conditions as a turn is negotiated. A bowler must compensate for the spin of the bowling ball as the ball rolls down the alley. A marksman firing a rifle must compensate for the distance to the target. The marksman analogy is amazingly similar to what happens with many forms of compensation on CNC equipment, so let's discuss it further. Say a marksman is standing 50 yards from a target. By one means or another, the marksman would adjust the sight on the rifle to allow for the 50 yard distance. The marksman would make the necessary adjustment, but until the first shot is fired, the marksman will not know for sure whether the initial sight adjustment was perfectly correct. Once the first shot is fired and the marksman can see the resulting hole location, the sight may have to be fine tuned to adjust for minor imperfections with the initial adjustment. In similar fashion, the CNC user will be faced with several situations when it will be impossible to predict the result of certain tooling related problems. So one form or another of compensation will have to be used to handle the problem. But just as the marksman may have to fine tune after the initial shot, so should the CNC user have to fine tune the initial compensation entry. More on how and why in a little while.

### **What are offsets?**

All forms of compensation work with offsets. You can think of CNC offsets as like memories on an electronic calculator. If your calculator has memories, you know you can store a constant value into each memory for use during a calculation. This keeps you from having to enter the number over and over again with redundant calculations. Like the memories of an electronic calculator, offsets in the CNC control are storage locations into which numerical values can be placed. Just as the value within the memory of a calculator has no meaning until referenced by its user within a calculation, neither does the value within an offset of the CNC control have any meaning until it is referenced by a CNC program. From the marksman analogy, you can think of the values stored in CNC offsets as like the amount of adjustment required on the sight of the rifle necessary to compensate for the distance to the target. Keep in mind that the rifle only needed adjustment for one purpose, to adjust for the distance to the target. With most CNC machine tools, there is a need for at least one offset per tool.

### **Reasons for tool offsets**

Offsets can be used for several purposes depending on the style of machine tool and type of compensation being used. Here are some of the more common applications for offsets.

#### **To specify tool each tool's length**

For machining center applications, it would be very difficult for the programmer to predict the precise length of each tool used in the program. For this reason, the feature tool length compensation allows the programmer to ignore each tool's length as the program is written. At the time of setup, the setup person measures the length of each tool and inputs the tool length value into the corresponding offset.

### **To specify the radius of the cutting tool**

When milling on the periphery of the cutter (contour milling), it can be cumbersome and difficult for the programmer to program the cutter's path based on the size of the milling cutter being used. Also, if the cutter size must change (possibly due to re-sharpening), it would be infeasible to change the program based on the new cutter size. For this reason, the feature cutter radius compensation allows the programmer to ignore the cutter size as the program is written. The setup person inputs the size of each milling cutter into its corresponding tool offset. In similar fashion, turning centers have a feature called tool nose radius compensation. With this feature, an offset is used to specify the radius of the very tip of the turning or boring tool.

### **To assign program zero**

Machining centers that have fixture offsets (also called coordinate system shifting) allow the user to specify the position of the program zero point within offsets, keeping the assignment of program zero separate from the program. In similar fashion many turning centers allow the assignment of program zero with offsets (this feature is commonly called geometry offsets).

### **To allow sizing on turning centers**

Tool offsets are used on all turning centers to allow the operator to hold size with tools used within their programs. This allows the operator to adjust for imperfections with tool placement during setup. It also allows the operator to adjust the tool's movements to allow for tool wear during each tool's life.

## **Organizing offsets**

With some CNC controls, the organization of offsets is very obvious. Some machining center controls, for example, automatically make the offset number correspond to the tool station number. With this kind of machine, when tool station number one is commanded, the control will automatically invoke offset number one. Within offset number one, the operator can store a tool length value as well as a tool radius value. Unfortunately, not all controls make it this simple. In many controls, each offset contains only one value and the offset number has no real relationship to the tool station number. In this case, the programmer must cautiously organize which offset/s are used with each tool.

For example, the tool length compensation offset numbers can be made the same as tool station numbers. Cutter radius compensation offset numbers can be made equal to the tool station number PLUS a constant value larger than the number of tools the machine can hold. If the machine can hold 25 tools, tool station number one could be made to use offset number one to

store its length compensation value and offset number thirty-one could be used to store its cutter radius compensation value. With this method of offset organization, the programmer and operator are constantly in sync. The offset table on most turning centers incorporate at least two values per offset. Generally speaking, the programmer will instate the offset number corresponding to the tool station number for each tool offset used. That is, tool number one will use (only) offset number one, tool two will use offset two, and so on. Additionally, most turning center offset tables allow the user to enter data related to the tool's radius (for tool nose radius compensation). Typically the radius (R column of the offset table) and the tool type (the T column of the offset table) can be specified within the turning center's offset table.

## **Types of compensation**

Now let's discuss the compensation types for the two most popular forms of CNC machine tools, machining centers and turning centers. Keep in mind that while the actual use of these functions vary dramatically from one machine to the next, the basic reasoning behind each compensation type remains remarkably similar. With an understanding of why the compensation type is required, and with an elementary understanding of how it is applied to one specific control, you should be able to adapt to any variations that you come across.

### **Tool length compensation.**

This machining center compensation type allows the programmer to forget about each tool's length as the program is written. Instead of having to know the exact length of each tool and tediously calculating Z axis positions based on the tool's length, the programmer simply instates tool length compensation on each tool's first Z axis approach movement to the work-piece. At the machine during setup, the operator will input the tool length compensation value for each tool in the corresponding offset. This, of course, means the tool length compensation value must first be measured. If tool length compensation is used wisely, the tool length compensation value can be measured off line (in a tool length measurement gage) to minimize setup time. With this method, the tool length compensation value is simply the length of the tool. Many CNC controls allow the length of the tool to be used as the offset value. One popular command to instate tool length compensation is G43. Within the G43 command, the programmer includes an H word that specifies the number of the offset containing the tool's length value. Here is an example program that utilizes tool length compensation with two tools. The program simply drills two holes (one with each tool). Notice that tool length compensation is being instated in lines N015 and N055.

Program

O0001 (Program number)

N005 T01 M06 (Place tool number one in the spindle)

N010 G54 G90 S400 M03 T02 (Select coordinate system, absolute mode, start spindle CW at 400 RPM, get tool number two ready)

N013 G00 X1.0 Y1.0 (Rapid to first XY position)

N015 G43 H01 Z.1 M08 (Instate tool length compensation on first Z move, turn on coolant)

N020 G01 Z-1.5 F4. (Drill hole)

N025 G00 Z.1 M09 (Rapid out of hole, turn off coolant)

N030 G91 G28 Z0 M19 (Return to tool change position, orient spindle)

N035 M01 (Optional stop) N040 T02 M06 (Place tool number two in spindle)

N045 G54 G90 S400 M03 T01 (Select coordinate system, absolute mode, start spindle at 400 RPM, get tool number one ready)

N050 G00 X2. Y1. (Rapid to first XY position)

N055 G43 H02 Z.1 M08 (Instate tool length compensation on tool's first Z move, turn on coolant)

N060 G01 Z-1.2 F5.5 (Drill hole)

N065 G00 Z.1 M08 (Rapid out of hole, turn off coolant)

N070 G91 G28 Z0 M19 (Return to tool change position, orient spindle)

N075 M30 (End of program)

As stated, this feature varies dramatically in use from one control model to the next. You must reference your control manufacturer's programming manual to learn more about how tool length compensation applies to your particular machining center. Sizing with tool length compensation in the marksman analogy, we said that the marksman would not know for sure whether the initial sight adjustment is perfectly correct until the first shot is fired. In similar fashion, the CNC operator will not know for certain whether the tool length compensation value is perfectly correct until the first work-piece is machined. Say for example, the tool length measurement was made incorrectly. During the measurement, the operator finds the tool to be 6.5372 in long. But the actual tool length is 6.5355 in. In this case, the tool would machine slightly shallower in Z than it is supposed to. After machining, the minor depth change can be made by adjusting the offset, **NOT THE PROGRAM**. In some cases, even if the tool length value is measured perfectly, other problems may cause the tool not to machine to the proper depth. If, for example, the work-piece or setup is quite weak, tool pressure may cause the work-piece to tend to push away from the tool doing the machining. For critical surfaces or when tool pressure is unpredictable, the operator can even trial cut the work-piece under the influence of an offset slightly **LARGER** than the measured value to ensure that some excess stock will be left. After machining, the operator can measure the surface to determine precisely how much offset change is necessary to machine the work-piece to size.

## **Cutter radius compensation.**

Just as tool length compensation allows the machining center programmer to forget about the tool's length, so does cutter radius compensation allow the programmer to forget about the cutter's radius as contours are programmed. While it may be obvious, let us point out that cutter radius compensation is ONLY used for milling cutters and only when milling on the periphery of the cutter. You would NEVER consider using cutter radius compensation for a drill, tap, reamer, or other hole machining tool.

### **Reasons for cutter radius compensation**

Let's begin by discussing four reasons why cutter radius compensation is not only required, but also very helpful to the CNC user.

**Program coordinates are easier to calculate.**

Without cutter radius compensation, machining center programmers must program the center-line path of all milling cutters. An example program using this technique was shown during the discussion of motion types (section three). When programming center-line path, the programmer must know the precise diameter of the milling cutter and calculate program movements based on the tool's center-line path. With cutter radius compensation, the programmer can program the coordinates of the work surface, NOT the tool's center-line path. This eliminates the need for many calculations. Keep in mind that we are now talking about manual programming. If you have a CAM (computer aided manufacturing) system, your CAM system can probably generate center-line path just as easily as work surface path.

**Range of cutter sizes.**

Say you do program center-line path for a given work-piece contour and do not use cutter radius compensation. Say you have programmed based on a one inch diameter tool. But when the job is to be run, you find that your company does not have any one inch end mills. Say the closest you have is a 0.875 in cutter. In this case, the entire cutter path would have to be changed in the program to match the new cutter size. With cutter radius compensation, handling this problem is as simple as changing an offset value.

**Easy sizing.**

As with tool length compensation, the operator can use the cutter radius compensation offset to help with sizing. If the contour is not coming out to size (possibly due to tool pressure), an offset can be changed to allow for the imperfection.

**Roughing and finishing.**

This is also a manual programming related reason for using cutter radius compensation. If contours must be rough and finish milled, cutter radius compensation allows the programmer to

used the same programmed coordinates needed to finish mill the work-piece to rough mill the work-piece. This keeps the programmer from having to calculate to sets of milling coordinates (one for roughing and one for finishing). To leave stock for finishing during the rough milling, the operator will simply make the cutter radius compensation offset value slightly larger than the cutter's actual size. This will keep the cutter away from the surface being milled and leave the desired finishing stock.

## **How to program cutter radius compensation**

The usage of cutter radius compensation does vary from one control to the next. Additionally, each control will have a set of strict rules that specify how cutter radius compensation is instated, used, and cancelled. Here we just show the basics of how it is programmed and give an example for how it is used on one popular control model. You must refer to your CNC control manufacturer's manual for more on your particular control. Most controls use three G codes with cutter radius compensation. G41 is used to instate a cutter left condition (climb milling with a right hand cutter). G42 is used to instate a cutter right condition (conventional milling). G40 is used to cancel cutter radius compensation. Additionally, many controls use a D word to specify the offset number used with cutter radius compensation. To determine whether to use G41 or G42, simply look in the direction the cutter is moving during machining and ask yourself if the cutter is on the left or right side of the surface being machined. If on the left, use G41. If on the right, use G42. Once cutter radius compensation is properly instated, the cutter will be kept on the left side or right side (depending on whether G41 or G42 is used to instate) of all surfaces until the G40 command to cancel compensation.

## **Dimensional tool (wear) offsets.**

This compensation type applies only to turning centers. When setting up tools, it is infeasible to expect the setup person to perfectly set each tool into position. It is likely that some minor positioning problem will exist. And even if all tools could be perfectly positioned, as any single point turning or boring tool begins cutting, it will begin to wear. As a turning or boring tool wears, the tool wear will affect the size of the work-piece being machined. For these reasons, and to allow easy sizing of turned work-pieces, dimensional tool offsets are required (also called simply tool offsets). Tool offsets are instated as part of a four digit T word. The first two digits command the tool station number and the second two digits command the offset number to be instated. The command T0101, for example, rotates the turret to station number one and instates offset number one. It is wise to always make the number of the primary offset used with a tool the same as the tool station number. When a tool offset is instated, the control actually shifts the entire coordinate system by the amount of the offset. It will be as if the operator could actually move the tool in the turret by the amount of the offset. Each dimensional offset has two values, one for X and one for Z. The operator will have control of what the tool does in both axes as the work-piece is being machined. Here's an example that should help you understand how dimensional tool offsets work. Say you have written a program to use tool number one (with offset number one) to turn a three inch diameter. After machining the three inch diameter, it is measured and found to be 3.005 in. That is, the work-piece is 0.005 in oversize. In this case, the

X value of offset number one will be reduced by 0.005 in. When the program is run again, tool number one will machine the work-piece 0.005 smaller.

### **How to guarantee your first work-piece comes out on size.**

If working on an engine lathe, manually turning a precise diameter, you would first allow the tool to skim cut to find out exactly where the tool is located. After skim cutting, you can determine precisely how much to turn the crank or handle to make the tool turn the desired diameter. In the same way, you can use dimensional tool offsets to ensure that any tool will not violate the work-piece on its first cut. Outside diameter turning tools, for example, could be offset slightly plus in X to ensure that some excess stock is left. Inside diameter bored holes could be offset slightly minus in X for the same purpose. In either case, the first time the tool is run, the operator can rest assured that the work-piece will come out with some excess finishing stock (it will NOT be scrapped). After machining the first time, the surface can be measured. The operator will then adjust the corresponding offset accordingly and re-machine with the tool this time the surface will be machined perfectly to size. Using this technique on each tool in the program will almost guarantee that the first work-piece will not be scrapped.

### **Tool nose radius compensation.**

This turning center compensation type is very similar to cutter radius compensation. In fact the same three G codes are used. G41 instates tool nose radius compensation in a tool left condition. G42 instates with a tool right condition. G40 cancels tool nose radius compensation. For this reason, this minimize the discussion of tool nose radius compensation to avoid repeating information. Just as cutter radius compensation allows the programmer to program work surface coordinates (not allowing for tool radius), so does tool nose radius compensation. To determine G41 or G42, simply look in the direction the tool is moving during the cut and ask yourself which side of the work-piece the tool is on. If the tool is on the left, use G41 (this would be the case when boring toward the chuck). If the tool is on the right, use G42 (turning toward the chuck). Once you determine which to use, include the proper G code in the tool's first approach to the work-piece. Once tool nose radius compensation is instated, it remains in effect until cancelled. Keep in mind that the tool nose radius is quite small (usually 1/64, 1/32, 3/64, or 1/16 in), meaning the deviation from the work surface will also be quite small. It is possible that if you are only chamfering corners to break sharp edges, you may not need tool nose radius compensation. However, if the surfaces being machined are critical (Morse taper, for example), you must compensate for the radius of the tool. Also, you should only need tool nose radius compensation when finishing. You should not use it for roughing operations.

### **Other types of compensation.**

The compensation types discussed so far have been for machining centers and turning centers. But all forms of CNC equipment have some form of compensation to allow for unpredictable situations. Here are some other brief examples. CNC Wire EDM machines have two kinds of compensation. One, called wire offset works in a very similar way to cutter radius compensation to keep the wire center-line away from the work surface by the wire radius plus the over-burn amount. It is also used to help make trim (finishing) passes using the same series of motion

coordinates. The second form of compensation for wire EDM machines is taper cutting. For machining the clearance angle needed with dies and form tools, the programmer can easily specify the direction of the taper (left or right) and the angle desired. The operator fills in some offsets to tell the control the position of the upper guide relative to the work-piece and the control does the rest. Laser cutting machines also have a feature like cutter radius compensation to keep the laser the radius of the laser beam away from the surface being machined. CNC press breaks have a form of compensation to allow for bend allowances based on the work-piece material and thickness. Generally speaking, if the CNC user is faced with any unpredictable situations during programming, it is likely that the CNC control manufacturer has come up with a form of compensation to deal with the problem.

## SECTION FIVE

### You must understand how to format programs

As stated in section one, the CNC control will execute a CNC program in sequential order exactly as it is written. All commands necessary to make the machine do the required operations must be included in the CNC program in the proper order. And of course, part of learning how to program is understanding the program structure a CNC machine requires. To this point, you have been exposed to several features and programming functions related to programming. With all the new ideas and commands introduced, you may be getting somewhat confused trying to keep them all straight. You may be worried about how you're going to memorize all of this. One of the main reasons to strictly format CNC programs has to do with making it easy to write your first few programs. When writing your first program, the related commands will by no means be memorized. However, if you have good example formats to go by, writing your first few programs will be much easier. We relate this to driving a car. It is unlikely that any driver can recite from memory all road signs used to direct traffic. However, when a driver sees a road sign, it is quite likely the driver will recognize its meaning. In the same way, it is unlikely that even an experience CNC programmer could recite every word used with CNC programming. But when even a relative newcomer to CNC sees a command, it is likely its meaning will be remembered. One of our intentions with program formatting is to keep you from having to memorize all commands needed for programming. Instead, you will be looking at an example and simply recollecting the function of each command. A second reason for strict program formatting is consistency. Once you have a format that works, use it. If you use the same format (or structure) for all programs you write, you will be able to repeat past successes. If all programmers in the university use the same format for a given CNC machine, each programmer will be easily able to work on another's program. The third (and most important) reason for strictly formatting programs is related to multi tool jobs. Almost all CNC machining center and turning center programs require that more than one tool in the program. For this kind of program, there will be MANY times when it will be necessary to rerun only one tool in the program a second, third, or fourth time.

Say for example, you have a machining center program that uses ten tools. After running a work-piece, you determine that the fifth tool in the program did not go quite deep enough. After fixing the problem (changing tool offset or Z position in program), you will need to run the fifth tool again. However, you would NOT want to run the entire program just to get to tool number five.

Doing so would be a waste of time and may actually cause unwanted problems with work-piece accuracy and finish. Instead, you will want to be able to run ONLY tool five a second time. To do so will require that ALL information necessary to get the machine running (just like at the beginning of the program) is included at the beginning of tool five. If the programmer makes certain assumptions related to modal information from a previous tool, it may not be possible to run tool number five by itself. Here is an example of a time when the programmer must include some redundant information at the beginning of a tool in order to give the ability to rerun the tool. In our previous ten tool example, we still wish to run tool number five a second time. Say that tools four and five both run at 500 RPM. Say the last tool in the program (tool number ten) runs at 1500 RPM. Spindle speed is modal. The programmer may decide to leave out the S500 word at the beginning of tool five, expecting it to carry over from tool four. After running the entire program, it is determined that tool number five did not go deep enough. The operator fixes the problem and intends to run only tool number five. In this case, tool number five will start at the same spindle speed as the last tool in the program (1500 RPM), not 500 RPM! This is but one time when redundant information must be programmed from tool to tool in order to give the capability to rerun tools. At the beginning of each tool, the programmer MUST include all information necessary to begin the tool, even if it means including some redundant information. In essence, the programmer must treat each tool as a mini-program that can run separate from the rest of the program. When you think about it, this actually simplifies the programming task, allowing the programmer to break a seemingly complicated multi-tool program into smaller and easier to handle pieces. Each tool makes up one piece of the program.

### **The four kinds of program format.**

For machines that have the ability to perform operations with several tools, there are four kinds of program format: program start-up format, tool ending format, tool start-up format, and program ending format. The programmer will begin every program with program start-up format. At the completion of program start-up format, the tool will be ready to begin cutting. At this point, the programmer will program the cutting operations with the first tool. When finished cutting, the programmer will follow the format to end the tool (tool ending format). Then tool start-up format to begin the second tool. The programmer will then toggle among cutting information, tool ending format and tool start-up format until the finished cutting with the last tool. At this point, the programmer will follow the format to end the program. For an example of the four kinds of program format, refer to the program given during our discussion of tool length compensation (i.e. section four). This program uses two tools and follows the strict format we are now discussing. Let's determine what commands are related to each kind of format. The first four commands (beginning with the program number) makeup the program start-up format. At the completion of line N015, the tool is ready to begin machining. Lines N020 and N025 makeup the cutting commands for the first tool. (In line N030, the feed-rate should be considered part of program start-up format.) Lines N030 and N035 form the tool ending format. Lines N040 through N055 are tool start-up format. (In line N060, the feed-rate should be considered part of tool start-up format.) Lines N060 and N065 are the cutting commands for the second tool. And lines N070 and N075 are program ending format. By breaking up the program in this manner, you should be able to see just how much of the program is nothing more than program format that can be copied from one program to another. Of course, certain word values like spindle speeds, feed-rates, axis positions, and tool station and offset numbers will change based on the

program you are currently writing. But the basic structure can be copied, keeping you from leaving out important information. Note that there are only four commands that do any cutting in this program. The bulk of the program is just format.

### **How do you come up with program format information for your machine?**

The best way is to take an example program that is currently running successfully and break it up in the manner shown above. When doing this, analyze just what each tool is doing to determine the various types of format. Ensure that each tool contains all information needed to run independently.

If you are working from scratch with a new CNC machines and have **no examples to go by, contact your machine tool builder to gain an understanding of how your** programs should be formatted. You may also find example programs given within your machine tool builder's programming manual.

Thanks