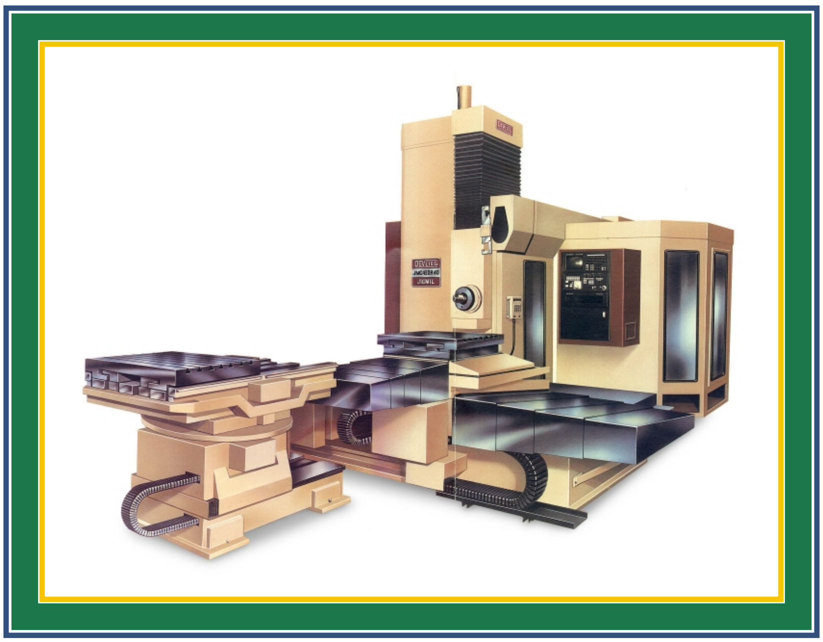
$39.00 USD

(Sample Pages Only)

7 Easy Steps to CNC Programming...

Book II

Beyond the Beginning



David S. Hayden
http://www.haydenpub.com

Table of Contents

Introduction

This book has been on my mind for a long time. However, it was not until I sold a couple thousand copies of 7 Easy Steps to CNC Programming. . .A Beginner's Guide that I realized there is such a thirst for information on CNC machines, programming and concepts.

Over the past couple of years I have collected a number of letters and requests for more detailed information. And, over the past 20 years as my duties evolved from that of a machinist to a programmer, supervisor and manager, I have become aware of the need to clear up a number of misconceptions about CNC machining.

The goals of this book are to do the following:

- Provide a glossary of CNC terms
- Present some of the basic concepts of CNC
- Discuss the current state of CNC programming
- Discuss the benefits of CNC machining
- Discuss some limitations of CNC machining
- Present some advanced CNC concepts
- Compare CNC machining with conventional machining
- Compare CNC control capability with Computer Aided Manufacturing (CAM) systems
- Offer some important management tips specifically related to CNC departments

I chose these topics for two reasons. First, because many readers of *7 Easy Steps to CNC Programming* asked for them. And, second as a manager of CNC and CAM departments, I have had to grapple with many of these issues on a day-to-day basis.

This book is laid out differently than most books on the subject. I realize that, as readers of this book, you are probably working in shops or departments that are understaffed and over committed. I suspect you don't have the time or inclination to wade through an exhaustive textbook.

So to better accommodate your needs, this book is designed to be a quick reference guide where you can find the information you need in the least amount of time.

The first part of this book is a comprehensive collection of answers to frequently asked questions. For your convenience there is a table of the FAQ's with the corresponding page numbers. (*see page 7*)

Following are additional chapters that provide some detailed discussion on:

- CNC Machining Accuracy
- Control Capability and Type of Work
- Machine Axis Definitions
- Selecting the Right CAM System
- Current State of CNC Programming

Frequently Asked Questions (FAQs)

CNC Background

What is NC / CNC?

NC is an acronym for Numerical Control. CNC is an acronym for Computer Numerical Control

What is the difference between NC and CNC?

The difference between NC and CNC is one of age and capability. The earliest NC machines performed limited functions and movements controlled by punched tape or punch cards.

As the technology evolved, the machines were equipped with increasingly powerful microprocessors (computers). With the addition of these computers, NC machines became CNC machines. CNC machines have far more capability than their predecessor.

Some of the enhancements that came along with CNC include: canned cycles, sub programming, cutter compensation, work coordinates, coordinate system rotation, automatic corner rounding, chamfering, and B-spline interpolation.

Where / how did CNC get started?

Controlling machines is practically as old as civilization. A good example is the early timekeeping devices such as water clocks.

Numerical control, as an extension of our desire to control machines, began in the late 1940's. At that time the US Air Force needed automated machining of complex, freeform airfoil surfaces. The first development in this area was the ability to drill holes at specific coordinates programmed on punch cards. This method, developed by John Parson, was the first of its kind.

By 1951 the Servo-Mechanisms lab at MIT had begun working at the task of improving the process. A year later, in 1952, MIT introduced the first NC machine; a milling machine adapted with electric servo-mechanisms and a control.

NC machines continued to evolve from that point. Some of the early developments included the ability to use absolute programming, metric programming, tool length offsets, and a host of miscellaneous commands to control spindle speeds, coolant, spindle ranges, and so on.

1970 marked the introduction of Computer Numerical Control and the CNC controller. With onboard processing capability, the new CNC controls could finally do the complex computations necessary for cuter radius compensation (CRC), sub programming, canned cycles, etc.

In some ways today's CNC machines do not do a whole lot more than their 1970 counterparts. The newer controls have significantly faster processors, a wider variety of canned cycles, more memory and so on, but basically they still move the tool from point A to point B.

Without a doubt, newer CNC machines are far more reliable than before. The new controls no longer need huge cabinets stuffed full of heat generating, power robbing fragile wiring. Now, all of the processing takes place in one small cabinet. The central processing unit sends commands and receives feedback via small Input/output (I/O) boards connected to the various relays, encoders and other feedback devices.

To improve reliability, control manufacturers now use optical fiber rather than electrical wiring to handle the communications between the computer and various components. This reduces heat and electromagnetic interference that is inherent wherever there is current running through a wire.

What background should a CNC machinist have?

This has been the subject of many debates. There are a couple of schools of thought.

At one time, and maybe still today, machine tool salesmen sold CNC machines on the notion that they were so simple to operate a "trained orangutang" could operate them. Consequently, many shops and CNC departments are run is such a way that the machine operator is just a button pusher.

In other shops, management looks at CNC machines as some of the most expensive equipment in the shop and they want only the best machinists running them.

Both schools of thought have their advantages and disadvantages. And, each type of operation has a different set of requirements.

A CNC operator shop that depends exclusively on "operators" rather than machinists, requires more supervision, better programmers, and often specialized setup people. These shops also need to look for different qualifications in their employees. If your goal is to rely on lower cost operators then, don't hire people that aspire to be programmers, managers, salesmen, etc., unless you plan on moving them up in the company fairly quickly. Instead, find people who are conscientious, want to do a good job, like structure, are more interested in their personal life than their career. These operators must carefully and reliably follow written and/or verbal instructions.

To be successful in this operation, management must also have a strong supporting staff that can provide fast, quality set ups and error-free programs.

Since the operator may not be a strong source of ideas regarding process improvement, you'll need a very competent and attentive supervisor, programmer or manufacturing engineer to monitor the machines. No shop can survive by doing things the way they did them five years ago. New tools, grades of carbide and work holding methods serve to improve efficiency and reduce machining times.

Remember, your competitors are trying all the latest ideas so they can get more business. If you don't keep up, your shop will lose business to them.

Some advantages of the CNC operator shop include:

- Lower labor costs.
- It's easier to find new employees.
- Operators tend to do what they are told and do less second guessing.

Z.1 - (rapid to .1 above the face)
G83G98Z-6.Q1.F5 -(Canned cycle code to drill 6.0"deep and retract drill after every inch of cutting)
M09 - (Turn coolant off)
G28X0Y0M30 - (Return to machine zero reference and reset to start of the program)

There is a wide variety of canned cycles to simplify hole drilling, back facing, boring and so on. CNC lathes have numerous canned cycles that simplify the programming of threads, stock removal, and so on.

NURBS interpolation: What is it and do I need it?

Here, "interpolation" means to estimate, by computer calculation, intermediate value(s) between two known values in a sequence that describes a curve section.

NURBS is a mathematical formula for describing curves. NURBS is an acronym for **N**on-**U**niform **R**ational **B** **S**plines.

If the work you do is limited to lathe, simple 2½ axis or even 5+ axis machining you will probably never need NURBS interpolation.

The NURBS Advantage

A NURBS-capable machine allows highspeed machining operations on machines that cannot otherwise handle the large volume of very short positioning moves required to cut a NURBS curve.

Imagine the number of points required to program a smooth curve. The greater the number of points, the smoother the curve. (*see figs 2 and 3*) Another important factor in

Fig. 2 - Points on a Circle

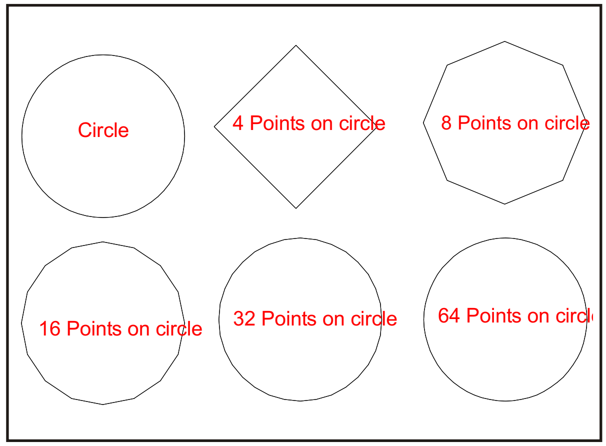
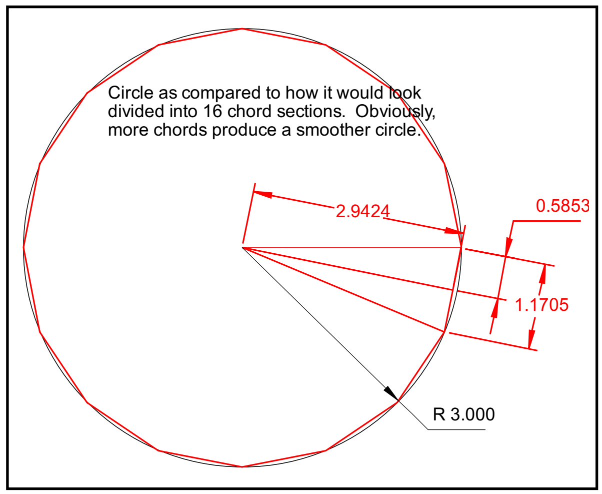


Fig. 3 - Chords on a circle



machining a smooth curve is continuous tool movement. If the tool dwells for even a hundredth of second, it will leave a visible mark on the part.

Since some of the moves between points on the spline can be very small .001 or less for tight curves, and since some cutting speeds can be more than 100 inches per minute, it is conceivable that a control would have to generate 50,000 data blocks per minute or 833+ blocks per second. Even today, many controls are not capable of that volume of data throughput. So, if the programmer does not take this into consideration, he can create a situation known as data starvation. When the machine is starved for data, it stops and waits for the next block of information, leaving a mark on the surface of the part.

Some manufacturers advertise the control's ability to process data this fast. However, processing speed and total data throughput are different concepts. Any processing speed in excess of I/O or buss speed is meaningless with regards to preventing data starvation.

Data Starvation

Data starvation occurs when a tool has to stop and wait for the next command from the control. When the tool stops, it leaves a visible mark on the part.

Another advantage of NURBS interpolation has to do with a machine's movement. Typically when using point-to-point programming, when the machine moves from one point to another, the tool accelerates when the tool leaves the first point and decelerates as it approaches the second point. So, the shorter the distance, the less likely the tool will achieve the programmed feed rate. This can affect cutting times and tool performance because the tool never achieves the programmed, optimum feed rate for the tool or cutting conditions.

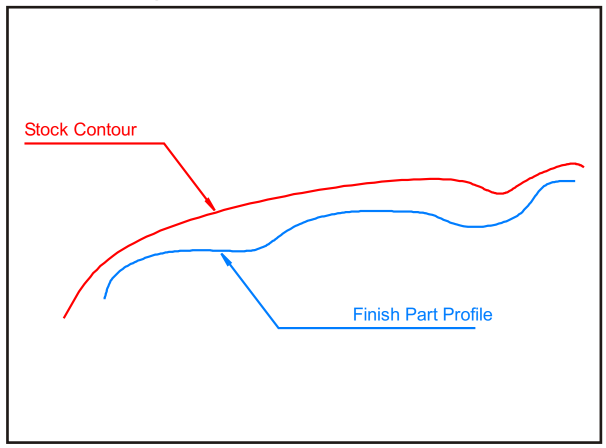
The NURBS Disadvantage

There are many NURBS algorithms. Thus, a machine's algorithm may not match the algorithm used by the CAD system that designed the part. Although algorithm differences are subtle, if you use NURBS interpolation to do very *close tolerance* work, the difference in algorithms between the machine and the CAD system may be enough to consume some or all of your tolerance.

Remember without NURBS interpolation, the machine spends much of its time accelerating and decelerating. NURBS interpolation minimizes this behavior. However, it has its own disadvantages.

Machining is the process of reshaping a piece of stock by removing material. The difference between the shape of the stock and the desired shape is what is know as the stock condition.

Fig. 4 - Stock Conditions



To optimize the life of the cutting tool and to preserve the integrity of the part, CNC programmers must maintain appropriate cutting methods based on existing stock conditions. Generally, the cutting method is based on worst-case conditions.

Particularly on the first pass, the stock condition will vary considerably depending on how closely the stock contour matches the finish part profile. As you can see in the previous diagram, there is a substantial difference between worst-case and best-case stock conditions.

Naturally, when the program is set for the worst-case, it is not optimized for the

Fig. 6 - Spline Curve

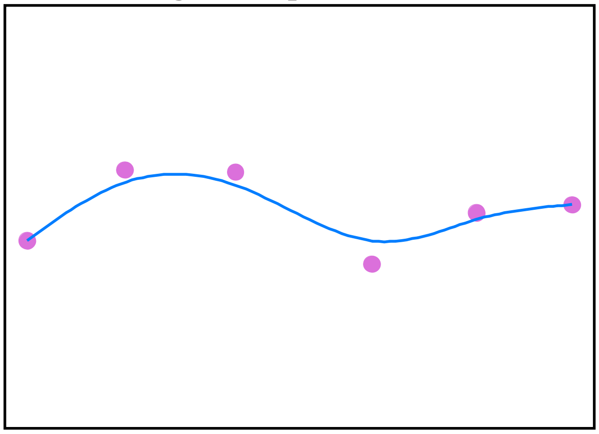
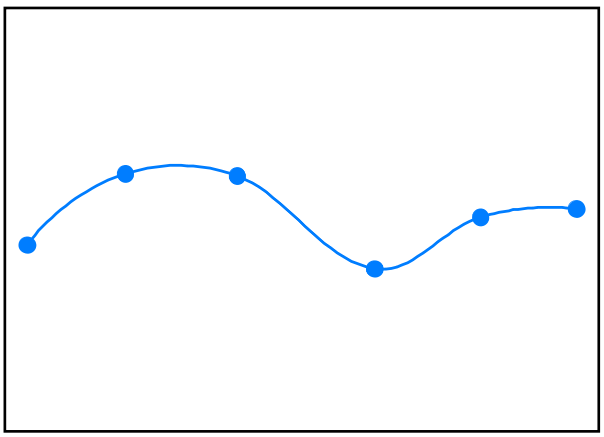


Fig. 7 - Bezier Curve

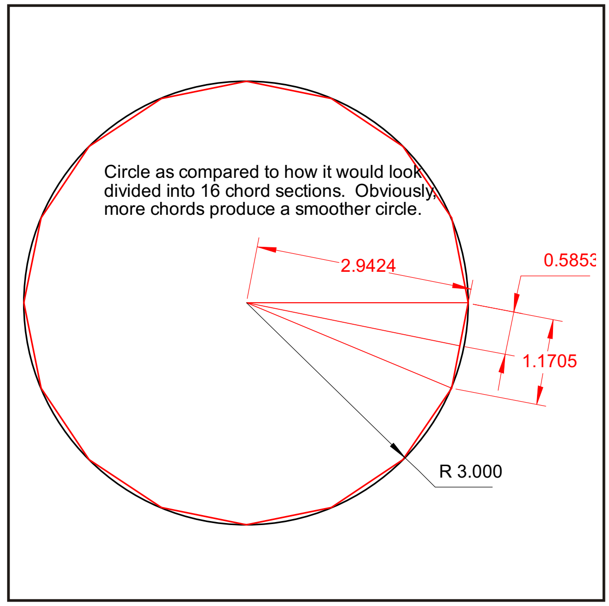


Chord and Accuracy

You know now that, when machining without NURBS interpolation, the programmer or CAM system must describe the motion as a series of closely spaced points along the curve. Naturally, as the machine makes a straight move from point-to-point, the cut part will deviate sightly from the perfect profile (*see the example below*). Again, the amount of deviation is dependent upon the number of programmed points.

Notice in **Figure 8**, the radius of the circle is 3.000 inches, while the distance from the center of the circle to the center of the chord is 2.9424 inches. Therefore, in this example, the deviation from the circle = 3.000-2.9424 = .0576.

Fig. 8 - Chords on a circle



This example only uses 16 chords to graphically represent the concept. When programming NURBS, the programmed deviations will generally be .001 or less. Production machines typically are not capable of holding a profile tolerance of .001 or less. This topic will be discussed in much greater detail in the chapter Accuracy and CNC Machining.

What are CNC Machine Parameters?

Parameter could be defined as any value designed to improve or customize a program. At the heart of every CNC machine is a computer program known as the *executive program*. This program tells the computer how to move the Axes, turn on the spindle, read CNC programs and so on.

The CNC control must know a lot about the machine if it is going to make good parts. For example, it must know the

The point at which the stops will be the Z Home for the machine.

Main Program - CNC machines can have "layers" of programs where one program calls another program that may in turn call another program. The main program is the program at the top of the stack. (*see diagram below*)

Material Removal Rate - The rate at which material is removed from the workpiece. Usually expressed in cubic inches of material removed per minute.

NC - Numerical Control.

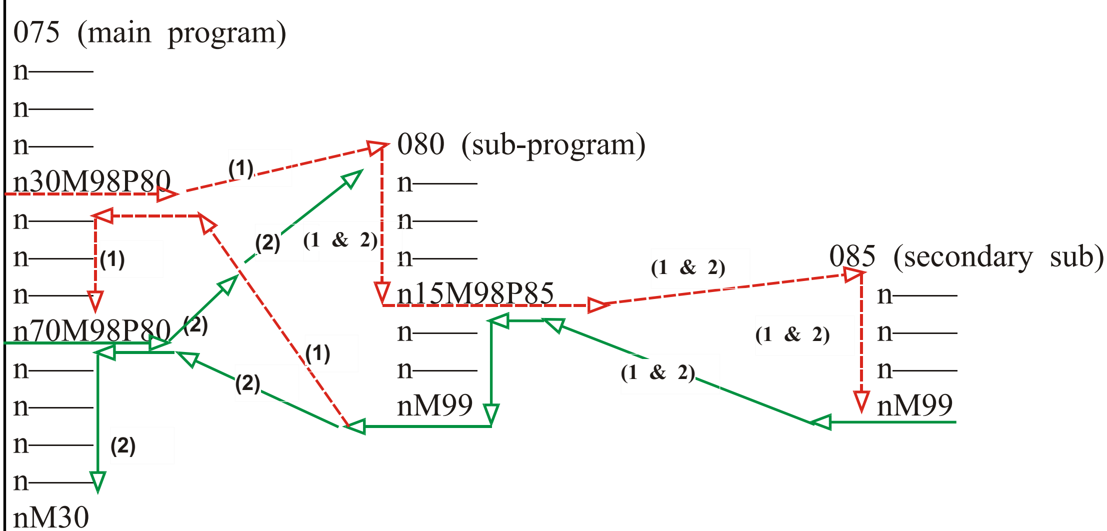
Nose Radius - The radius of the tool tip.

NURBS -Non-uniform Rational B Spline.

NURBS Interpolation - A feature on some controls that enable it to cut a NURBS. Without this feature, the program must be created using a series of chords to approximate the curve.

Open Architecture - A control that does not use proprietary hardware. Open controls can use a variety of "off-the-shelf" components available from various sources rather than the original manufacturer.

<u>The following is a diagram of the Main/sub-programming process</u>



To interpret this diagram start at the O75 (main program). As you read down, notice at line n30, the program calls (with an M98), sub program 80. Moving through program 80 at line n15, sub-program 85 is called. Program 85 runs down to M99 which, returns control to program 80. Program 80 continues processing until it reaches its M99 command, which returns control to the main program.

From here, the main program continues to line n70 where control is returned to programs 80 and 85. After programs 80 and 85 finish, control is returned to the main program, which continues until it reaches the end at the M30.

Follow the arrows on the diagram on **page 29** to see how program control moves from main program to sub-programs and back.

Proper use of sub-programming can save hours of programming time, significantly reduce the length of a program, and simplify editing.

When you want to call a sub-program, use the command M98P(xxxx). M98 commands the control to leave the current program and complete the instructions in the called program (specified by the P-code).

Sub-programs always end with an M99 which commands return to the program that called the sub-program.

Surface Feet Per Minute (SFM) - The speed at which the surface of a spinning part or cutter is traveling as expressed in feet per minute.

To calculate a desired RPM for a Specific SFM use this formula:

$$\text{RPM} = \frac{\text{Desired SFM} * 3.82}{\text{Diameter}}$$

To calculate the SFM for a specific RPM use this formula:

$$\text{SFM} = \frac{\text{RPM} * \text{Diameter}}{3.82}$$

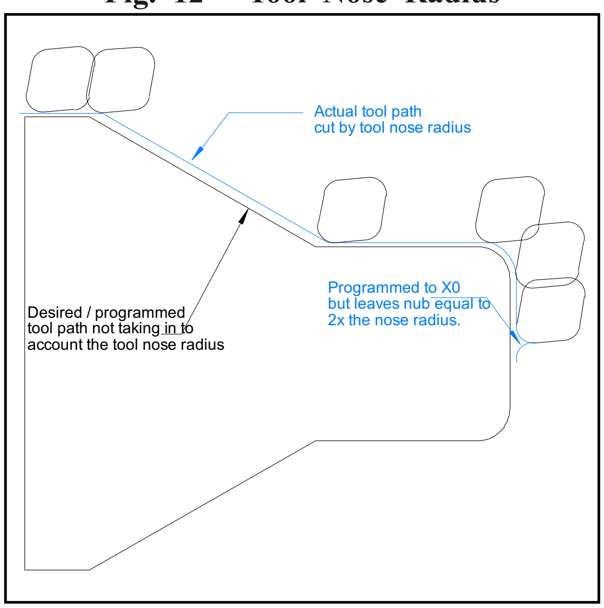
Tool Length Compensation - Tool length compensation is the feature of CNC controls that enables the operator to enter the length of a tool.

To program older controls, programmers were required to know the length of the tools before creating the NC program. The tool length offset available on CNC machines makes it possible for the control to compensate for any length of tool so the programmer is only required to specify which offset to use.

Tool Nose Radius - Lathe tools rarely have a perfectly sharp corner at the tool nose. As a result, when you program lathes, it is important to compensate for the tool nose radius.

As you look at the tool nose radius example below, imagine these two situations. Assume you are using a tool with a .031 nose radius. As you cut down the face of the part, if you stop at X0, the tool, the actual cut will be .031 above the center of the part, leaving a .062 nub at the center of the part. Then if you want to drill a hole in the center of the part, the drill walks off center because of the nub.

Fig. 12 - Tool Nose Radius



Accuracy and CNC Machining

Computer Numerical Control (CNC) implies a certain degree of accuracy, but how accurate is accurate? There is a big difference between positional accuracy and contouring accuracy.

It is expected that a CNC lathe can accurately cut a part face within .0001 in. or cut a diameter within .0002 in. Similarly, CNC milling machines are expected to position to a face, hole or shoulder locations within .0001 in.

If these machines can position that accurately, shouldn't it follow that machined contours would be just as accurate? The answer is no because cutting contours is far more complex than cutting surfaces that are perpendicular to a machine axis.

First let's discuss positional accuracy. When turning a diameter, facing a part, or locating for a drilled hole, we are asking the machine to:

- Move to a start point
- Lock all machine Axes except one
- Move the tool along the unlocked axis until the cut is completed

At no point during the actual cutting process do we have more than one axis moving. Just as important, the cutting edge of the tool that starts the cut is the same cutting edge that finishes the cut.

Contouring can be very different because contouring requires the coordination of multiple Axes. Further, contouring requires using different points of contact along the cutting edge of the tool.

Fig. 14 - Straight Turning

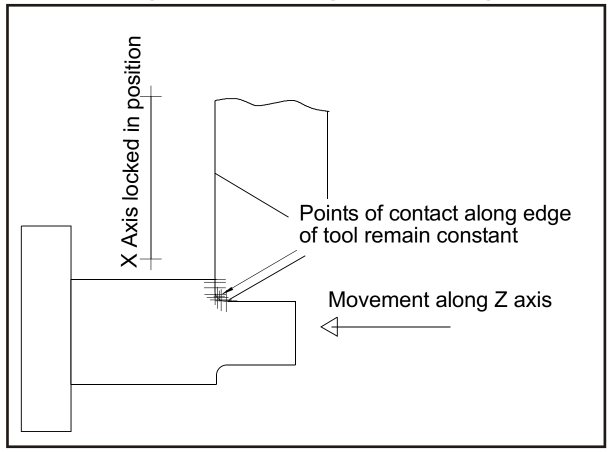


Fig. 15 - Straight Facing

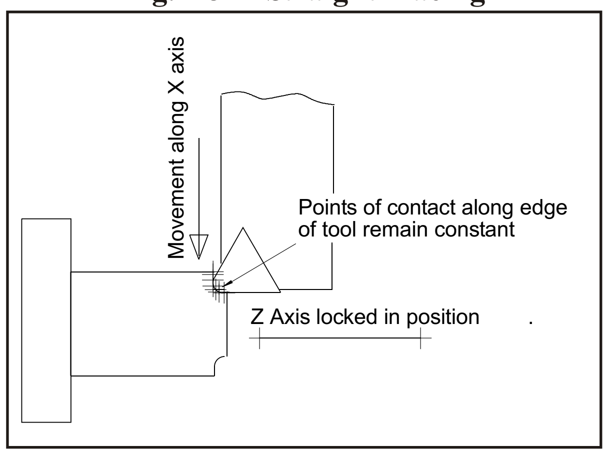
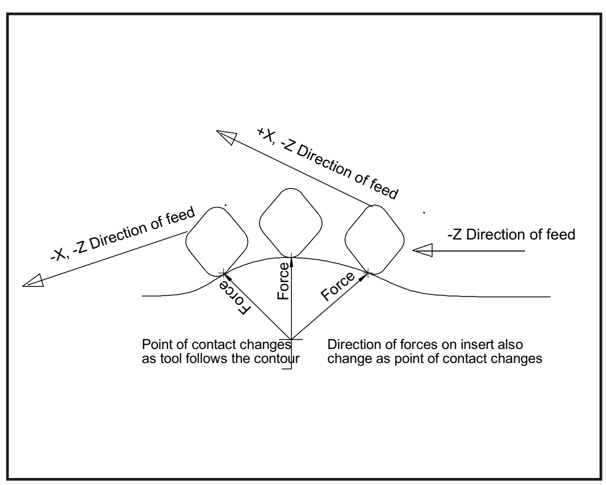


Fig. 16 - Lathe Contour Cutting

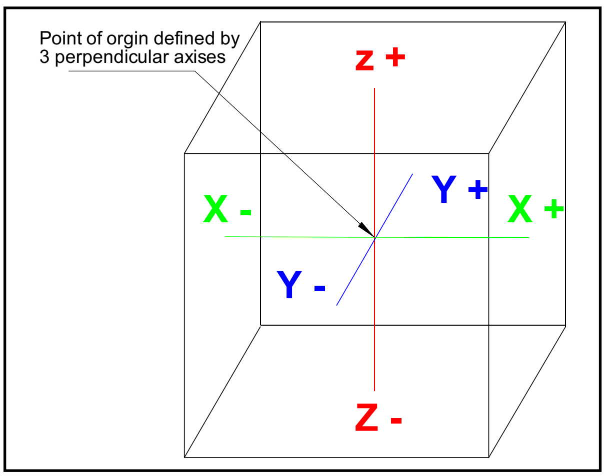


Cartesian Coordinates

A basic understanding of the Rectangular and /or Cartesian Coordinate System is essential in CNC Programming. First we will define Rectangular Coordinates. Then we will discuss how we use this system to program parts.

Over 300 years years ago a French mathematician named René Descartes developed a system that enables people to define the location of any point in space. This system uses three axis that are intersecting and mutually perpendicular. This system uses one axis for each of the following; width, length and height, typically labeled X, Y, Z. *(see Fig. 23 below)*

Fig. 23 - 3 Dimensional Space



While you may not be familiar with the names Rectangular or Cartesian Coordinates, you use this system every time you navigate through your home town. Most cities have an origin point defined by the intersection of two streets. As you travel away from this intersection (origin point), the block or building numbers increase in predictable increments. Addresses like 900 West Third St., Suite 1058 are descriptions based on rectangular coordinates.

Some cities are not laid out so conveniently but many are. If for example the previous address was in Denver, Colorado, your origin point would be the intersection of Broadway and Ellsworth. Broadway runs north and south and Ellsworth runs east and west. The address 900 West Third St., Suite 1058 would put you on the 10th floor of a building 9 blocks west of Broadway and 3 blocks north of Ellsworth.

The 3 Dimensional Space drawing is shown to help you visualize this system. The box around the axis lines can help you to visualize a three dimensional space. Using the street address, including the 10th floor location may also help you to visualize the dimension of height.

After reviewing some basic geometry terminology we will return to The 3 Dimensional Space drawing for more clarification. Meanwhile, a basic understanding of points, lines, and planes will aid you in using the rectangular coordinate system.

POINT: A **point** is defined as a location only. It defines a place not an object, so it has no length, width, height, dimension or extent.

LINE: A **line** is the shortest distance between two points.

Fig. 24 - Line

In geometry, a line is conceived as being straight and having infinite length but no other dimension such as width or breadth.

Machine Axis Definitions

The previous chapter on rectangular coordinates discussed how any point in space can be described by 3 mutually perpendicular Axes.

NOTE: To understand this chapter, you must understand the 3-dimensional rectangular coordinate system. This chapter requires much visualization, so I occasionally re-use the same figure to illustrate different concepts.

So, why do we even need 4, 5, or more axis machines?

The rectangular system I discussed in the last chapter defines a static (stationary) space in X, Y, Z coordinates. In CNC, we need *motion.* So, we must have a coordinate system (other machine axes) in which to program tool *motion*. We do this with *motion commands*.

Sure, we can describe *any point in space* with rectangular coordinates. But, it is not necessarily possible to make a good part by limiting *motions* to the three primary axes.

You can program both *linear* and *rotational* moves. As an overview, here are the programmable axes of NC machines:

A, B, and C commands control *rotation* around the X, Y, and Z axes, respectively.

U, V, W, X, Y and Z commands control *linear* motion with respect to the X, Y, Z axes.

Now, for the details.

Controlling Linear Motion

Study Figures 26-27 starting on this page.

Fig. 26 - NC Axes, Lathe

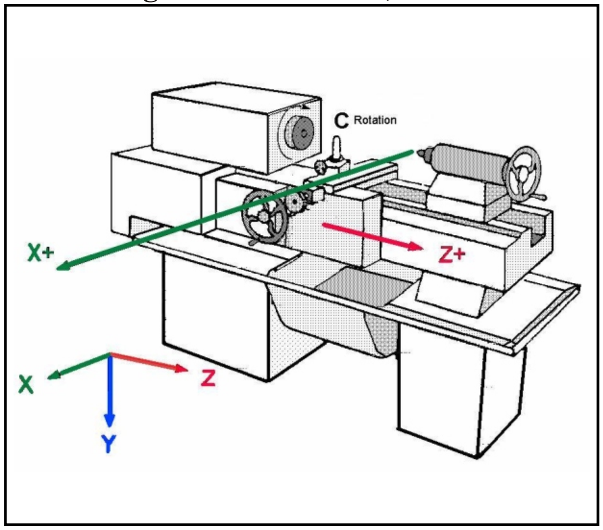
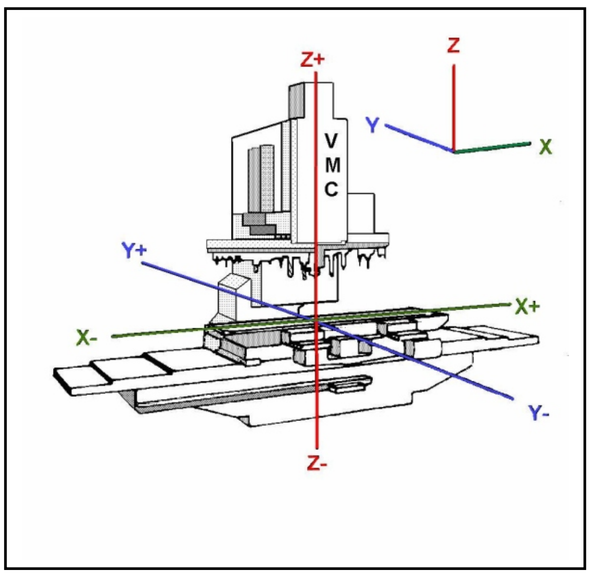


Fig. 27 - NC Axes, VMC



Notice that the Z axis command is most common among machines because it is always parallel to the centerline of the rotating spindle.

STEP 1. Determine how you are going to machine the part. You will need to know:

- appropriate speeds and feeds
- fixturing
- the order and sequence of operations
- the number of programs and setups required
- the machine limitations
- any safety considerations.

WHY: You must know how you are going to make the part before you can program the machine to make it.

STEP 2. Make up a tool list of the tools you intend to use.

WHY: The tooling you use will make a difference in how the part should be programmed.

STEP 3. Make up a setup sheet and define the tool/cutter paths as they relate to the program origin.

WHY: This step organizes your thinking and provides the operator with description of how you intend to set up and run the part. Making a drawing that details the cutter paths around the part will keep you on track and help with calculations.

STEP 4. Write the program by using the setup sheet.

WHY: This keeps the process organized and gives you a reference to go back to in the event you get interrupted.

STEP 5. Proofread your manuscript for errors.

WHY: Because you will catch mistakes that could be costly.

STEP 6. Type/load the program into the machine control.

STEP 7. If you manually input the program using the MDI panel or have reason to doubt the reliability of your input device, use the CRT of the machine to proofread the program for errors.

WHY: See step 5. . .every time you input data, there is a potential for error.

7 Easy Steps to CNC Programming . . . A Beginner's Guide focused on steps 2, 3, 4, 5, and 7. Steps 1 and 6 are dependent on the work you're doing, the type of machinery and fixtures you have available in your shop.

When I wrote the first book, I had worked primarily in small machine shops that rarely had the financial resources to purchase a Computer Aided Manufacturing software package. Even the less expensive CAM systems were beyond the budgets of the company. Added to the CAM system costs were issues of training people and associated costs. Naturally, because these shops generally paid lower wages, if they did get someone trained, a larger shop with more money was likely to hire them away.

That scenario has not changed much. Small shops are still cash strapped and more likely to rely on manual programmers than CAM systems. However, mounting pressures in the industry may force smaller shops to use CAM software.

For one, companies subcontracting their work are putting pressure on smaller shops to use CAM systems, often compat-

Selecting the Right CAM System

Computer Aided Manufacturing (CAM) software can be a powerful tool for most shops. Still many shops may never need a CAM system. For example, a small shop that does mostly simple turning, milling, and drilling may have all the power they need in the machine's CNC control. In this case, buying CAM software may be a waste of money.

Shops that frequently make complex parts requiring complicated calculations to complete the NC program could benefit from CAM software. Another compelling reason to buy a CAM system is to bring in more business. Often customers require CAM capability.

Once you decide to buy CAM software, your work really begins. There are dozens of very good CAM systems. Most of them can do all the basic NC programming. And, there are vast differences in capabilities, features, costs, learning curves, training expenses, customer support, etc.

As a CAM Systems Administrator, one of my duties is to stay on top of trends in the CAD/CAM Industry. Over the years, I have learned a lot about selecting and maintaining CAD/CAM software, and integrating old technologies with new. Following are the guidelines I use and /or recommend to upper management regarding CAM system selection.

The size and future direction of your shop should have a strong influence over your selection of CAM software. If you are a small job shop with limited resources and you need CAM capability, your choices will be based on cost vs features.

If your shop must interface with engineering departments, customer CAD systems and so on, your decision will be far more complex.

Depending on your situation, you should review and answer some or all of the following questions:

- Is the CAM system going to be of strategic importance to the company?
- What is our goal for having a CAM system?
- What is the budget for purchasing the software?
- How much money can our company afford to budget annually for maintaining the software and hardware?
- What are our competitors using?
- What are our customer expectations with regards to our NC / CAM programming capabilities?
- What CAD programs are our customers using?
- What are our machine capabilities / requirements? Turning, milling, 5 or more axis positioning, 3 or more axis simultaneous movement?
- What is the local labor pool from which we can draw programmers?
- Is our NC / CAM programming done by dedicated programmers or by machinists, operators, supervisors, methods engineers, etc.?
- Do we need an MIS or IT staff to support the software and hardware?
- Will CAM software ownership enhance our shops market position?

Fig. 32 - Sample Criteria Matrix

	Software Criteria	Value	Score	Brand W Value x Score	Score	Brand X Value x Score	Score	Brand Y Value x Score	Score	Brand Z Value x Score
REQIRED	Must have lathe module	5	2	10	4	20	3	15	1	5
	Must be easy to use	5	5	25	2	10	1	5	3	15
	Must support Parasolid	5	5	25	5	25	5	25	0	0
DESIRED	TAC must be less than $15,000	5	1	5	4	20	4	20		
	Want Local Training	5	0	0	0	0	3	15		
	Want ability to run on PC w/ Windows XP	2	5	10	5	10	5	10		

In the Sample Criteria Matrix example above, notice how Brand Y *(green)* has the highest score. The more thoroughly you test the software, the more reliable your score will be.

Notice that Brand Z has no total score because it failed the "required" test for total acquisition cost. Once a product fails a required test, there is no point evaluating it further. You can save a lot of time by evaluating all required criteria first.

Notice the values columns In this example, the criteria are rated for importance on a scale of 1-5. All of the required items have an importance rating of 5.

The highly desirable criteria, or wants do not need an importance rating of 5. So in this example, you can see local training is more important than the software's feature recognition capability.

Brand W scored the highest for feature recognition but still had a total score lower than Brand Y because local training is more important.

The score for each brand is calculated by multiplying the Value column by your quantified subjective score as to how well the product will meet that particular criteria.

For example, Brand W above has a lathe module, but not a very good one. So it's score for lathe module is 5 (the value) x 2 (a low score for the lathe module) = 10 Brand X obviously must have a better lathe module.

You can use a similar matrix for any significant purchase whether it is a new machine tool for the plant or a computer for home. I know of one woman that very successfully used a similar matrix concept for choosing her spouse!