AUTOMATIC GENERATION OF NC-CODE FOR HOLE CUTTING WITH IN-PROCESS METROLOGY

Thomas R. Kramer
Guest Researcher, National Institute of Standards and Technology
& Research Associate, Catholic University
Room A-127, Building 220, NIST, Gaithersburg, MD 20899

ABSTRACT

A new method to mill flat-bottomed circular holes with more accurate diameters has been added to the data preparation software for the Vertical Workstation of the Automated Manufacturing Research Facility at the National Institute of Standards and Technology. This software already had the capability to generate NC-programs automatically for cutting two-and-a-half dimensional parts. Additional design functions, a new process planning function and a new NC-code generating function have been added to the software to implement the new method. The new cutting algorithm uses a touch probe to measure the diameter of the semi-finished hole during the cutting process. The radius used to finish cut the hole is then changed from its nominal value by an amount equal to the difference between the nominal and measured values of the radius of the semi-finished hole.

The new hole milling process corrects errors caused either by tool deflection or by using a tool whose actual radius differs from its nominal radius. With the new process, errors in the diameter of a hole cut with an end mill have been reduced from roughly five mils (plus tool diameter error) to about one mil (regardless of tool diameter error), as compared with a process which does not measure during cutting. The new process is integrated into the Vertical Workstation system by allowing the user to specify the diameter tolerance of the hole during the design process. The automatic process planner then selects the new process for high tolerance holes.

BACKGROUND

Nature of the Problem

For small batch production, the inability to produce a high-tolerance hole without special tooling is an important practical problem. A hole with very high diameter tolerance must be finished with a tool (end mill, ream, or boring tool) whose cutting diameter is the same as the diameter of the hole. However, for a certain tolerance range, roughly 0.5 to 5 mils (1 mil = 0.001 inch), exact sized tools may not be required if good enough machining techniques are available.

It is desirable to avoid having to use exact sized tools in order to save time and money. If a hole is an odd size, a tool of exact size may not be on hand, and will have to be obtained by purchase or manufacture. There are 1000 exact sizes less than an inch if 1 mil accuracy is required. Having a tool inventory of that order of magnitude is very costly. Even if a tool is available, it may not be in the tool carousel of the machining center when it is time to cut the hole. The setup time to put it in and take it out is costly. If a small set of standard sized end mills can be used to make most milled holes, a good deal of money might be saved, particularly where only one or a few parts of a given design are to be made.

Description of the Vertical Workstation

The Automated Manufacturing Research Facility (AMRF) at the National Institute for Standards and Technology (NIST) - formerly the National Bureau of Standards - serves as a testbed for developing techniques and standards for automated manufacturing [13]. Small batch production is emphasized in the AMRF. The AMRF includes three machining workstations. One of these is the Vertical Workstation (VWS), which contains a Monarch VMC-75 Vertical Machining Center with a GE2000 controller. The VMC-75 is a 3-axis machine. It is equipped with a Renishaw touch probe.

Software has been developed for the Vertical Workstation, called the VWS2 system, which supports the automatic machining of a family of two-and-a-half dimensional parts [3]. The VWS was used as the testbed for the research reported here, and the techniques developed were embodied in software which was integrated in the VWS2 system. This work was done in conjunction with the Quality in Automation project being carried out in the AMRF [14].

Three types of documents: designs, process plans, and equipment control programs, are of key importance in the VWS2 system. A design may be created as a feature-based design using the VWS2 design editor [4]. For a more limited range of parts, a boundary representation design in PDES/STEP format may be parsed automatically into a feature-based design [7]. A process plan for machining is prepared automatically from the feature-based design [8], and then an NC-program for the GE2000 controller is prepared automatically from the design and the process plan [6].

Producing Holes

In the VWS2 design protocol, a hole is defined as a depression with a circular outline that has a flat or conical bottom, or goes all the way through the part. The parametric representation of a hole includes x and y coordinates of the center of the hole, plus diameter, depth, and bottom-type. In enhancements not previously documented, center-tolerance and diameter-tolerance have been added as parameters.

Holes may be made by many methods, of course. Process plan work elements and automatic NC-code generators are in place in the VWS2 system for drilling, milling, and counterboring. In this paper we deal only with milling. Before the research reported here was done, a hole-milling algorithm was already in place in the VWS2 system. We will call it the "old algorithm", and we will call the one reported here the "new algorithm".

Both algorithms use an end mill. The old algorithm starts the hole by one of three methods: cutting a slot down the middle, spiralling in, or plunge cutting, depending on the amount of room available for the tool in the hole. Next, material is removed by peripheral milling, if necessary, to make a hole whose depth is that of the designed hole but whose radius is 10 mils less. Finally, the old algorithm performs a finish cut on the sides of the hole to remove the last 10 mils and achieve the designed diameter. In calculating the tool path, the system uses the value of the diameter of the end mill stored in a database of current tooling.

Automatic Generation of NC-Programs

Computer systems for the semi-automatic (user-interactive) generation of NC-programs are widely used. At least 40 such systems are commercially available [9], and more exist in university, government, and private research laboratories. A few systems do not require user interaction once a design and process plan have been prepared [1]. Fully automatic generation of NC-programs which use probing for in-process metrology is extremely rare.

OBJECTIVES

The objectives of the work reported here were:

1. to develop an algorithm for making holes with a tighter diameter tolerance than was being achieved with the old algorithm, without requiring the use of a tool with the same diameter as the hole.

2. to integrate the algorithm into an automatic machining system, so that its use would be triggered by the tolerance requirements given in a design, with process planning and NC-programming handled automatically.

SOURCES OF ERROR

There are many ways for error to creep into machining a hole with an end mill. We will omit detailed discussion of novice-level errors such as using a 4-flute end mill in aluminum (or other tool-workpiece mismatches), plunge cutting with a non-center-cutting end mill, or using a dull tool.

Several less elementary errors are discussed below. Methods of correction are discussed for each error type. Geometric analyses of control algorithm error and small tool path radius error are given in the appendix to this paper.

Tool Diameter

An end mill may have a spinning volume which is a nicely shaped cylinder, but have a cutting diameter (the diameter of the cylinder) which is different from the tool's nominal diameter or last measured diameter. Tool wear might account for this. A tool may have been resharpened (a common practice) and be slightly undersized as a result. It is much more common for an end mill to be undersized than oversized, since wear and sharpening remove material.

Some machining centers, including the Monarch VMC-75, have cutter radius compensation. To use it, the tool must be measured, a parameter set in the machine tool controller, and an instruction issued in the NC-program. In the VWS2 system, another correction method is to use the exact cutter diameter in the current tooling database. This also requires measuring the tool.

Tool Deflection

Even if the spindle of the machine tool is following a correctly defined tool path exactly, the tip of the tool may not follow the correct path because cutting forces bend the tool to the side. In milling a hole, the last cut is normally a circular cut around the surface of the hole. For this type of cut, tool deflection will make a hole that is too small, and the sides of the hole may taper, so that the diameter is smaller at the bottom than at the top.

Tool deflection may be corrected by taking very light cuts to minimize the bending force on the tool, by reducing the feed rate (also to reduce bending force), or by enlarging the tool path slightly to compensate for bending.

Chatter

The tool may vibrate rapidly while cutting, making a loud noise called chatter. When a tool chatters it bangs against the workpiece. This results in a rough surface and large errors in surface location.

It is hard to predict when chatter will occur, but it may usually be eliminated by reducing the feed rate, taking lighter cuts, or changing the spindle speed.

The workpiece may also vibrate, typically if the ratio of the thickness of the part to the distance from where it is being machined to the nearest fixturing point is small. This problem is harder to deal with, and may require refixturing or changing the tool path.

Chip Interference

If chips of material cut by an end mill are not removed promptly from the vicinity of the end mill, the end mill may grab them and drag them against the workpiece. This results in a rough surface.

The cure for this is to clear chips away as soon as they are formed. If this is not feasible, periodic chip clearing (especially just before finish cuts) will help.

Position Measurement Error

NC machining centers perform machining by repeating a simple sequence of operations at a fixed rate of repetition [10]. Typically, a cycle lasts a few milliseconds. The operations are: measure the spindle position, calculate where it should be at the end of the next cycle, and issue the control signals required to move the spindle in a straight line to get it there at the right time.

If the position measurement system of the machine tool is not accurate, holes made by the machine tool will not be correctly made. Repeatable position errors may be compensated by mapping them and putting corrections into NC-programs.

Control Algorithm Error

In making a circular arc, the machine tool control system makes a series of straight line segments to approximate the arc. If the control algorithm makes segments whose endpoints are on the arc, the average diameter of the hole will be slightly too small. If the control algorithm makes segments that are tangent to the arc and whose endpoints lie outside the arc, the average diameter will be slightly too large. Other algorithms are likely to make segments that lie between those two extremes. As shown in the appendix, the maximum difference is approximately $d^2 / (2\ r)$, where d is the length of a segment, and r is the radius of the hole.

Although it is not clear whether control algorithm error will ever be significant, the worst it can get is when the radius of the arc is very small, since the difference is inversely proportional to the radius. A circular tool path 10 mils in radius made at a feed rate of 15 inches per minute will take about 0.25 second to make. If one segment is made each millisecond, so that there are 250 segments, the difference described above is about 0.003 mil. If it takes ten milliseconds to make a segment, so that there are 25 segments, the difference is about 0.3 mils.

Control algorithm error might be reduced by reducing the feed rate of the tool, so that d is small. Since feed rate is normally adjusted to make chips of a certain size, reducing it may cause problems in machining some (but not many) materials. Handling the adjustment automatically would require special test and correction routines in the Process Planning module.

Control Execution Error

There is always some error in the execution of an NC-code instruction to move the tool. The largest error usually results from overshoot or undershoot in the direction of tool movement. The error becomes noticeable when there is a large change in the direction of successive tool movements.

The simplest method of reducing control execution error is to reduce the feed rate, so that smaller movements are required in each clock cycle.

Overshoot and undershoot may also be reduced by using a special machine code provided for that purpose [12] or by avoiding large changes in the direction of tool movement. Large changes in direction are avoided by having each programmed linear or circular move start out in the same direction in which the last one finished (i.e. successive motions have a common tangent).

Dwell Error

If an end mill is allowed to dwell in one place against the wall of a hole it is cutting while it is spinning, for even a fraction of a second, it will make a slight depression in the wall [10]. This seems to be caused in part by the tool unbending after being subjected to cutting forces, but a depression will be made after even the lightest of cuts.

Dwelling often occurs during the execution of an NC-program, even if it is not part of the program. If the spindle is to be retracted after a cut, for example, there is usually a brief dwell between the end of the cut and the retraction.

Dwell error is eliminated by not dwelling during a finish cut. This requires knowing what sequences of NC-program steps may result in unintentional dwell and avoiding them.

Small-Circle Tool Path Error

If the radius of a circular tool path is very small, another interesting type of error crops up. The shape of the cross section of the swept volume of the tool becomes significantly different from a circle. This is because the tool revolves only few times as it travels around the tool path. The appendix gives a geometric analysis of this error.

As with control algorithm error, decreasing the feed rate should solve the problem, but may not be the most desirable solution.

NEW PROCEDURE

General Approach

The general approach taken in the new algorithm is:

1. Rough-cut the hole using the old techniques.

2. Make a circular semi-finish cut using a control radius 10 mils smaller than the radius that should nominally be required to cut the final hole.

3. Measure the diameter of the semi-finished hole and calculate the error in the radius of the semi-finished hole.

4. Make a circular final finish cut whose control radius has been adjusted by the error factor found in step 3. If the measured radius of the semi-finished hole was smaller than its nominal value, make the control radius of the finish cut larger by this amount. If the error was in the other direction, make the adjustment in the other direction.

The assumption behind this approach is that if the semi-finished hole and the finished hole (which are nominally identical holes except that the radius of the semi-finished hole is 10 mils smaller) are cut in the same manner, the errors made in cutting them will be essentially the same.

Because of the several undesirable side effects of tool paths with small radii (which were observed in early experiments), the tool used in the new algorithm is chosen to be significantly smaller than the hole being cut.

To avoid dwell marks on the side of the hole, the tool is not allowed to dwell against the side of the hole.

To avoid radial overshoot on starting the hole, the tool is brought into its cutting path on an arc tangential to the path.

Tool Path

The tool path for the initial rough cut is not critical. The rough-cut hole is nominally made 20 mils smaller in radius than the final hole.

The semi-finish cut removes a layer 10 mils thick around the inside of the hole, and then makes another trip around (nominally cutting nothing) to clean it up well under minimal cutting forces. A picture of the path is shown in Figure 1.

Next, machining comes to a halt, and a comment in the program appears on the console of the controller, reading:

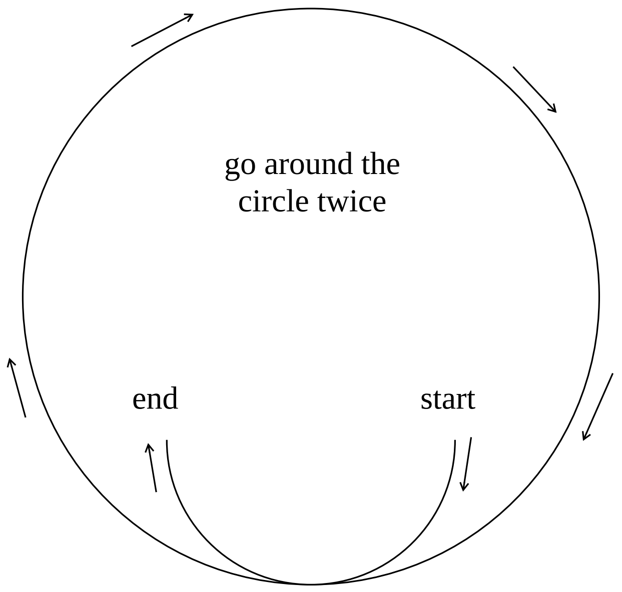
```
Changing tool to probe for measuring hole.
Please clean chips and coolant out of the hole.
Then press cycle start.
```

The console operator follows these instructions.

When the machine is restarted, the tool is changed to the probe, and the hole-measuring subroutine provided with the Monarch is run to find the diameter of the semi-finished hole. The subroutine automatically sets a parameter in the GE2000 controller to the value of the diameter. The new algorithm sets another parameter to the nominal value of the radius of the final cut plus the difference between the nominal and measured values of the semi-finished hole. This last parameter is used for the radius of the tool path of the finish cut.

Finally, the tool is changed again, so that the end mill is in the spindle, and the finish cut is made as shown in Figure 1.

Figure 1. Tool Path for Finish Cut



Probing Measurement Method

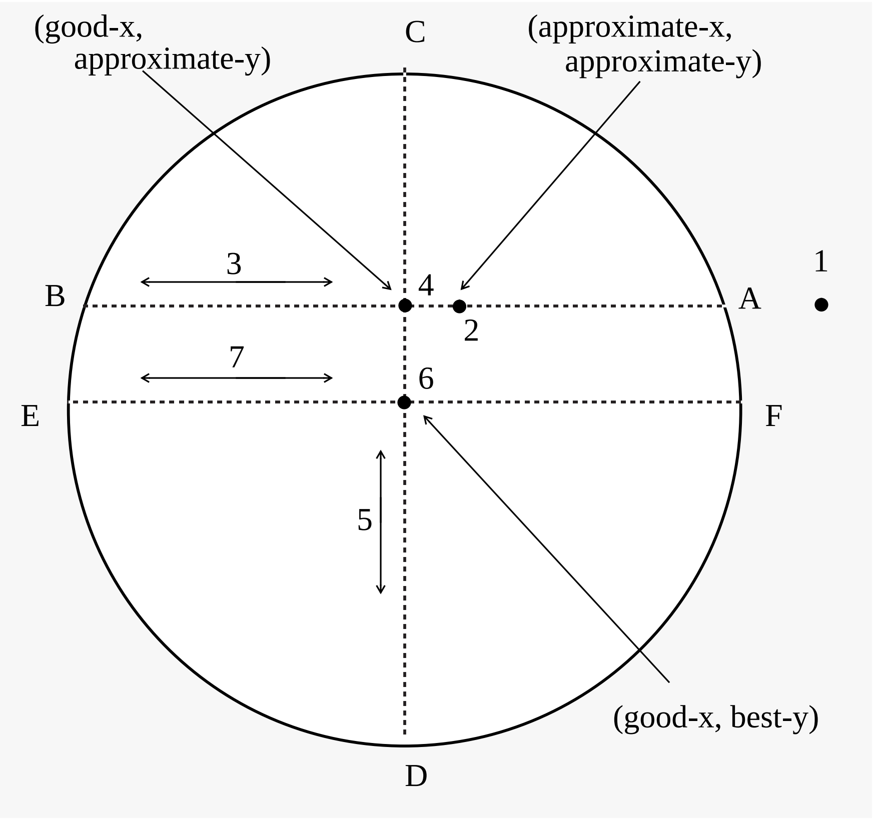
The probing subroutine provided with the Monarch [11] is used in the new algorithm. The main NC-program provides the subroutine with approximate-x and approximate-y for the center of the hole and the approximate diameter of the hole. The tool path used by the subroutine is shown in Figure 2, and is as follows. Numbered items below correspond to numbers on the figure.

1. Probe the surface of the part outside the hole to find the z-location of the hole.
2. Insert the probe in the hole at the approximate center.
3. Move the probe parallel to the x-axis back and forth to opposite sides of the hole, touching at A and B. Let good-x be the average of the x-values at A and B. Good-x will be very close to the x-value of the center of the hole, if the hole is round.
4. Move the probe to (good-x, approximate-y).
5. Move the probe back and forth parallel to the y-axis to opposite sides of the hole, touching at C and D. Let best-y be the average of the y-values at C and D. Store best-y as the y-value of the center of the hole. Store the length of CD as a value for the diameter of the hole.
6. Move the probe to (good-x, best-y).
7. Move the probe parallel to the x-axis back and forth to opposite sides of the hole, touching at E and F. Let best-x (not shown on the figure) be the average of the x-values at E and F. Store best-x as the x-value of the center of the hole. Store the length of EF as another value for the diameter of the hole.

The average of the two values of the diameter is stored in a parameter of the controller as the diameter of the hole.

The depth of insertion into the hole must be set in the NC-program. The new algorithm uses a quarter inch or 0.02 inches less than the depth of the hole, whichever is less.

Figure 2. Tool Path for Probing



Error Sources Compensated

The new algorithm compensates principally for tool deflection error and tool size error. It is designed to avoid other errors to the extent that can be done in NC-code. To the extent other errors cause a hole to be the wrong size without throwing it out of round, the algorithm will compensate for them (but it is hard to show there are any such errors).

EXPERIMENTAL RESULTS

To test the new procedure, three pairs of holes were made using tools which were already in the tool carousel of the machining center. Both holes in a pair were the same nominal size and were made with the same tool. Three different end mills were used. The largest hole was made with an 0.625 inch diameter end mill that had been used a long time, and, judging from the results of cutting with it, may have been resharpened at some time, so that its actual diameter is significantly smaller than the diameter listed in the current tooling database.

The data are shown in Table 1. These data were taken on July 28, 1988 by the author with a hand-held dial caliper, accurate to about 1 mil. Separate measurements were taken by Mr. David Caparelli of the NIST Precision Metrology Group using a Mitotoyo coordinate measuring machine. Those measurements agreed within about one mil.

In addition to the data shown in Table 1, measurements of the center location of holes made with the new algorithm and the circularity of all holes were made. Both types of error did not exceed about 1 mil for any measurement.

The data may be summarized by observing that diameter errors have been reduced from roughly five mils (plus tool diameter error) to about one mil (regardless of tool diameter error). If the old algorithm included an extra pass around the final cut, as the new one does, the hole diameter error for the old algorithm might be slightly smaller.

TABLE 1. EXPERIMENTAL RESULTS

cutting tool nominal diameter (inches)	hole design diameter (inches)	old or new	hole measured diameter (inches)	hole diameter error (mils)
0.375	0.752	old	0.747	5
		new	0.753	1
0.5	0.8741	old	0.868	6
		new	0.873	1
0.625	1.300	old	1.284	16
		new	1.301	1

INTEGRATION

The new procedure has been fully integrated in the VWS2 system.

Design

When a hole is being created or edited using the VWS2 design editor, the editor prompts the user to specify whether the diameter_tolerance for the hole is high or medium, and the user must choose one of the two. For backward compatibility, a hole from an old design with no value for diameter_tolerance is treated as if the diameter_tolerance were medium.

The "features" database has been updated so that the design verification system (which is data driven) will check that the diameter_tolerance parameter has an appropriate value, if there is a value.

Process Planning

The Process Planning module decides to drill a hole if the hole has a conical bottom or if it is a through hole for which a drill of the right size can be found in the tool catalog. Otherwise it decides to mill the hole.

The Process Planning module takes diameter_tolerance into account for holes which it has determined should be milled. If the tolerance is high, it selects the process named "mill_hole_probe" for making the hole. Otherwise, it selects "mill_pocket". For machining purposes, a hole is just a degenerate form of a pocket, as far as the system is concerned.

If the mill-pocket operation is used to mill the hole, the Process Planning module will select the largest end mill in the tool catalog whose radius is at least 10 mils smaller than the hole radius. If mill_hole_probe is used, the Process Planning module will select the largest end mill in the tool catalog whose radius is at least 10 mils smaller than *half* the hole radius. This is to avoid the errors caused by a small tool path radius described earlier.

As currently implemented, the value of diameter_tolerance in the design serves only as a two-way switch in the Process Planning module for determining which hole milling algorithm will be used for a milled hole. That module also uses the value of the center_tolerance of the hole (also high or medium) to determine if a hole to be drilled should be center-drilled beforehand. In the long run, numeric values should be used for these tolerance parameters.

NC-programming and Verification

In the Data Execution module, which does NC-programming [6], the new algorithm is used if the process plan step is "mill_hole_probe", and the old algorithm is used if the process plan step is "mill_pocket".

The Data Execution module is also responsible for verifying that a step in a process plan can be carried out safely. It does this before writing NC-code. A separate verification function has been written for the mill_hole_probe operation. In addition to the checks on hole milling that already existed in the system and are described in [5], the "mill_hole_probe_test" function checks that the hole radius is at least 1.75 times the tool radius (to avoid small tool path radius errors), that the hole is at least 0.3 inches in diameter (so that the probe will fit into it), and that the hole is at least 0.15 inches deep (also so that the probe will fit).

In addition to generating NC-code, the NC-coding function performs additional verification, in case the user has turned off or overridden the verification system. In particular, the NC-coding function checks that the radius of the hole is at least 0.04 inch larger than the tool radius, and that the hole is at least 0.15 inches deep. It will inform the user of the problem and will not write code if either of these two checks fails.

DISCUSSION

Limitations

The new procedure for milling holes has a number of limitations. Seven of these are discussed here, with brief comments on how they might be overcome.

First, although diameter accuracy improved significantly, diameter error continued to be of the order of one mil. For many applications, such as fitting a shaft tightly in a hole, the diameter error needs to be kept smaller. Improvements in the new algorithm may be feasible (for example by calibrating the probe during the procedure), but improvement beyond the designed location accuracy of the machine (0.3 mil) does not seem likely. The new procedure, therefore, does not replace additional operations, such as reaming, which are normally used to achieve diameter tolerances of a tenth of a mil.

Second, the new procedure relies on the machining center to be able to make a round hole. Unless all measurements of the diameter of a hole are within x of each other, it does not make sense to say that the diameter of the hole is within x of the desired value [2]. The Monarch VMC-75 makes very round holes, as long as the tool diameter is not close to the hole diameter, so this has not been a problem, but some other machine may not do so well.

Third, the new procedure is not taking any special steps to control other dimensions of the hole, such as depth or center location. Center location is measured by the probing subroutine which is being used to find the diameter, and it has been very good. As long as the hole is round, the measured values of the center location could be used to compensate for errors in that location, but this is not being done now.

Fourth, the new procedure is significantly slower than the old one because of the need to change tools during machining, the time taken for probing, and the time taken to clean the hole before probing.

Fifth, the new procedure is not fully automatic. A human cleans the hole before it is probed. A machining center with good chip control and a directable air stream to dry the hole should be able to overcome this limitation.

Sixth, the new procedure does not work if the semi-finished hole is larger than the final designed size. This may happen, for example, if the tool is more than 10 mils oversized. Unless the wrong tool is used, this is very unlikely. A semi-finished hole larger than the final designed size never occurred in testing the new algorithm.

Seventh, the new procedure uses the hole-measuring subroutine provided with the machining center. This routine starts by probing the surface of the part just outside the hole to find the vertical location of the hole. If that surface has been machined away, the subroutine may not work properly. This limitation is easily removed by rewriting the subroutine. The vertical location of the hole is known from the design and fixturing specifications, so there is no need to probe for it.

Future Development

It should be feasible to extend the method to cutting other shapes. The first new shape would be rectangular depressions with rounded corners (commonly called pockets). Probing and correction algorithms would be significantly more complex, but the approach of using the results of probing to set parameters used for calculating the final tool path should be workable.

It may be feasible to achieve tighter tolerances by refining the algorithm. Random error in positioning is supposed to be one or two tenths of a mil on the Monarch VMC-75 used in the VWS (a typical figure for high-quality machining centers), so that should be an upper limit on possible improvement. No ideas for refinements have been tested.

Conclusions

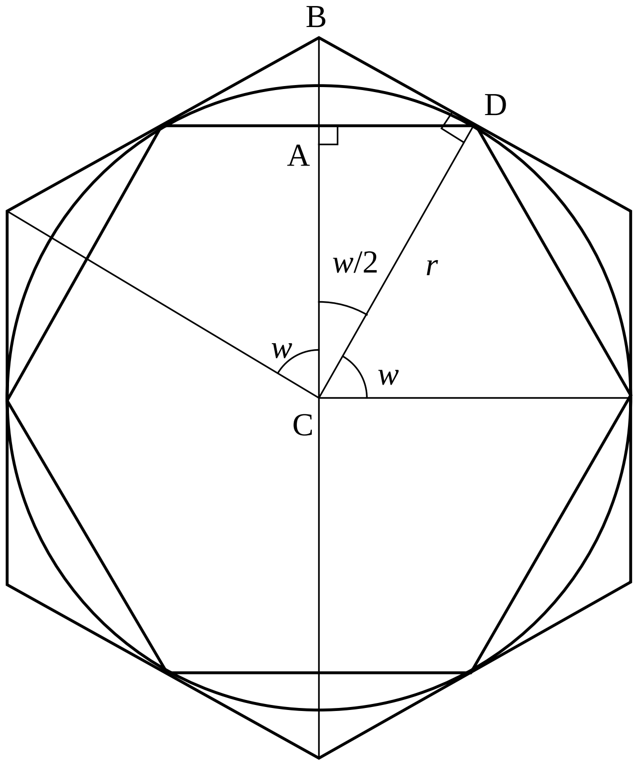
It is feasible to generate NC-code instructions automatically for cutting holes using a procedure in which the hole is measured by a touch probe during machining, and the results of probing are used to calculate the final tool path. The procedure will make holes to a significantly higher tolerance than a procedure which does no error compensation.

APPENDIX - GEOMETRIC ANALYSES

Control Algorithm Error

In making a circular arc, the machine tool control system makes a series of straight line segments to approximate the arc. If the control algorithm makes segments whose endpoints are on the arc, the average diameter of the hole will be slightly too small. If the control algorithm makes segments that are tangent to the arc and whose endpoints lie outside the arc, the average diameter will be slightly too large. Other algorithms are likely to make segments that lie between those two extremes. The geometry of the two algorithms is shown (with the size of segments highly exaggerated) in Figure 3. The arc to be cut is the circle shown in the figure. The inner and outer polygons are the paths made by the two algorithms. Other lines are construction lines inserted to make the geometry clear.

Figure 3. Control Algorithm Error



As shown in Figure 3 the maximum difference between the diameters determined by the two algorithms given above is twice the length of BA, which may be found as follows:

2BA = 2(BC - AC)
but, from inspecting triangle BCD, BC = $r \sec(w/2)$
and, from inspecting triangle ACD, AC = $r \cos(w/2)$, so

$$2BA = 2\,r\,[\sec(w/2) - \cos(w/2)]$$

where r is the radius of the circle, and w is the angle subtended by a segment.

If w is small, three approximations may be applied:

$w = d/r$ where d is the segment length
$\cos w = 1 - (w^2/2)$
$\sec w = 1 + (w^2/2)$

With these approximations, the equation for the difference reduces to:

$$d^2 / (2\,r)$$

If the polygons are rotated with respect to one another, the maximum difference is less. The objective in these calculations is to get an estimate of the magnitude of the error.

Small-Circle Tool Path Error

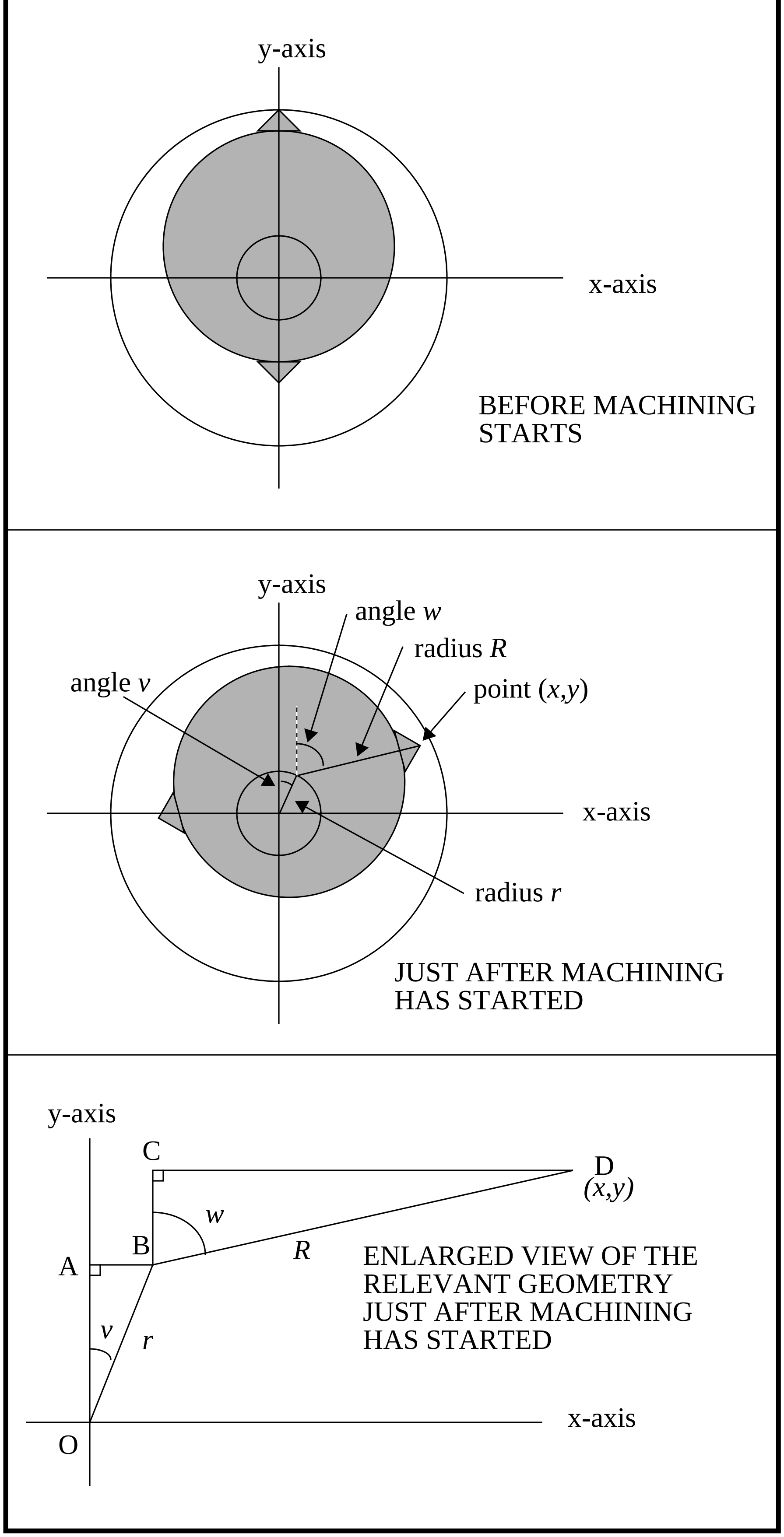
If the radius of a circular tool path is very small, another interesting type of error crops up. The shape of the cross section of the swept volume of the tool becomes significantly different from a circle. This is because the tool revolves only few times as it travels around the tool path.

If spindle speed (in rpm) is s, and feed-rate (in inches per minute) is f, and the radius of the tool path (in inches) is r, then the time taken to traverse the tool path is $(2\,\pi\,r) / f$. The number of revolutions of the tool in time t is $(s\,t)$, so the number of revolutions made during the cut is $(2\,\pi\,r\,s) / f$. Notice that for a fixed feed-rate and speed, the number of revolutions approaches zero as r approaches zero.

As an example, consider a two-flute one-inch diameter end mill running at 600 rpm, and 15 inches per minute being used to cut a hole 1.02 inches in diameter. The tool path is a circle 0.02 inches in diameter, which is about 0.0625 inches long. Thus it takes about 0.25 second to cut it. In this amount of time, the tool makes 2.5 revolutions.

The error in circularity of the cross section of the swept volume in this example is small enough that it would not show up in a picture, but it can be calculated. Figure 4 shows a line drawing of the cross section of the tool (shown shaded), with the tool not drawn to scale. A coordinate system is located with its origin at the center of the hole to be cut. The hole is the outer circle. The tool path is the inner circle. The flutes are shown lying on the y-axis in the top picture. In the middle picture, the situation is shown as it would be after the tool starts to rotate. An enlarged view of the relevant geometry after machining has started is shown at the bottom.

Figure 4. Tool Cross Section



Let R be the radius of the tool, r be the radius of the tool path, v be the angle of rotation of the center of the tool with respect to the origin, and w be the angle of rotation of the tool around its axis. It may be seen from the bottom figure that the x and y coordinates of the tip of the upper flute of the tool are given by:

x = AB + CD
y = OA + BC

But AB = $r \sin v$ and CD = $R \sin w$, so

$$x = r \sin v + R \sin w$$

Also, OA = $r \cos v$ and BC = $R \cos w$, so

$$y = r \cos v + R \cos w$$

But, we determined above that $w = 2.5\, v$, so

$$y = r \cos v + R \cos [2.5\, v]$$
$$x = r \sin v + R \sin [2.5\, v]$$

For the cross-section of the swept volume of the tool to match the hole, the minimum value of y for the tip of one of the flutes should reach -0.51 on the cross-section. We will show that it does not reach this value.

At the minimum value of y, the derivative of y with respect to v should be zero.

$$dy/dv = - r \sin v -2.5\, R \sin[2.5\, v]$$

$$0 = - r \sin v -2.5\, R \sin[2.5\, v]$$

$$-r / [2.5\, R] = \sin[2.5\, v] / \sin v$$

For our example, $r = 0.01$ and $R = 0.5$, so

$$-0.008 = \sin[2.5\, v] / \sin v.$$

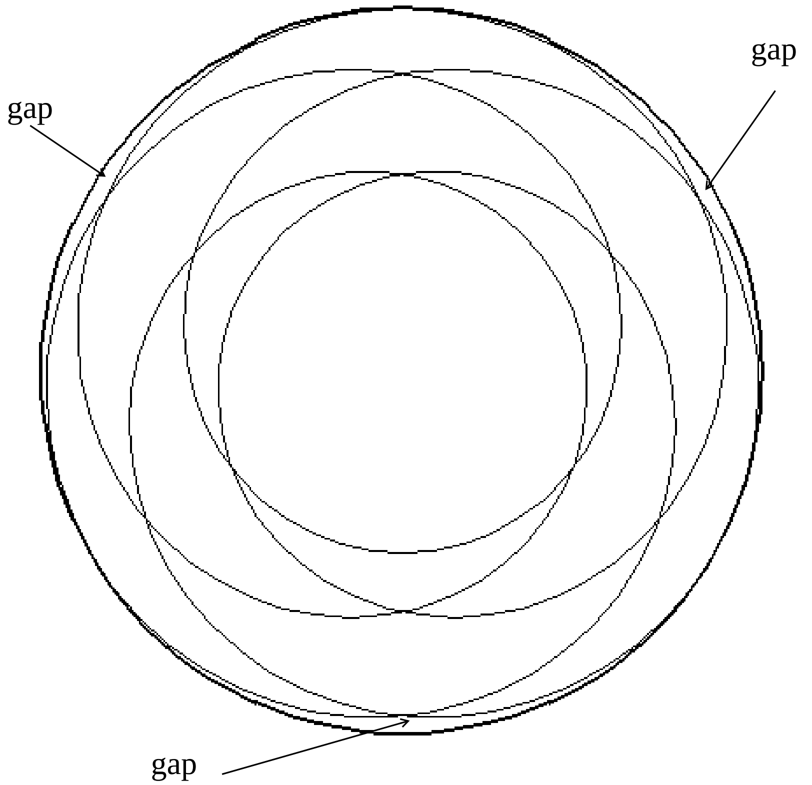
This has several solutions, since it includes local minima and maxima. The solutions $v = 3.768$ radians and $v = 8.798$ radians yield the minimum value $y = -0.508$, which is two mils above the desired minimum.

The other flute does not get any lower because the other flute follows the same path (since the first flute is at the location of the second flute when v has gone through one complete turn).

To give a qualitative feel of the shape of the cross-section, the path of the tip of the tool (the envelope of which is the cross section) is shown in Figure 5. In Figure 5, the tool radius is three times the tool path radius, rather than 50 times as large. This produces a small but easily visible error. The intended shape of the hole is shown with a heavy line.

The actual cross-section would differ from that shown in Figure 5, even if the figure were drawn to scale, because the phase angle between *v* and *w* is not necessarily 0, as used in the calculations. Also, actual machine tool control is not likely to keep *w*=2.5*v* exactly, since acceleration and deceleration around the tool path are required. A gap of similar size is still likely to occur.

Figure 5. Swept Volume Cross-Section



The outer, heavier line shows the outline of a circle. The inner line shows the path of the tip of the flute of the end mill. Note the gaps between the circle and the outermost arcs (the envelope) of the path.

REFERENCES

[1] T. C. Chang, D. C. Anderson, and O. R. Mitchell, *QTC - An Integrated Design/Manufacturing/Inspection System for Prismatic Parts*, Proceedings of the 1988 ASME Computers in Engineering Conference, San Francisco, California, August 1988, Vol. 1, pp. 417 - 426.

[2] Theodore H. Hopp, *What is a Tolerance? The Problem of Methods Divergence in Flexible Automation*, notes for a talk given at Conference on Uncertainty in Engineering Design, unpublished, May 1988.

[3] Thomas R. Kramer, and Jau-Shi Jun, *Software for an Automated Machining Workstation*, Proceedings of the 1986 International Machine Tool Technical Conference, Chicago, Illinois, September 1986, pp. 12-9 through 12-44.

[4] Thomas R. Kramer, and Jau-Shi Jun, *The Design Protocol, Part Design Editor, and Geometry Library of the Vertical Workstation*, NBSIR 88-3717, National Bureau of Standards, 1988, 101 pages.

[5] Thomas R. Kramer, and W. Timothy Strayer, *Error Prevention and Detection in Data Preparation for the Vertical Workstation Milling Machine in the Automated Manufacturing Research Facility at the National Bureau of Standards*, NBSIR 87-3677, National Bureau of Standards, 1987, 61 pages.

[6] Thomas R. Kramer, and Rebecca E. Weaver, *The Data Execution Module of the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards*, NBSIR 88-3704, National Bureau of Standards, 1988, 58 pages.

[7] Thomas R. Kramer, *A Parser That Converts a Boundary Representation into a Features Representation*, NISTIR 88-3864, National Institute of Standards and Technology, 1988, 18 pages.

[8] Thomas R. Kramer, *Process Plan Expression, Generation, and Enhancement for the Vertical Workstation Milling Machine in the Automated Manufacturing Research Facility at the National Bureau of Standards*, NBSIR 87-3678, National Bureau of Standards, 1987, 56 pages.

[9] Thomas R. Kramer, *Automatic Generation of NC-Programs for Metal-Cutting Machines Literature Search*, Not yet published, 1988, 20 pages.

[10] Douglas A. Milner and V. C. Vasiliou, *Computer-Aided Engineering for Manufacture*, McGraw Hill, 1987, pp. 20 - 26.

[11] Monarch Cortland Co., *Probing on the Monarch VMC with General Electric Controls*, Monarch Cortland Co., Cortland, New York, Publication E1985-5.

[12] Monarch Cortland Co., *Programming Manual for Monarch VMC-75 and VMC-150 General Electric 2000MC Controls*, Monarch Cortland Co., Cortland, New York, Publication Number PRG GE2000MC-4, Chapter 7.

[13] Philip Nanzetta, *Update: NBS Research Facility Addresses Problems in Setups for Small Batch Manufacturing*, Industrial Engineering, June 1984, pp. 68 - 73.

[14] Theodore Vorburger, *Quality in Automation*, keynote address at Symposium on Quality Issues in Automation held as part of 109th ASME Annual Winter Meeting, Chicago, Illinois, November 1988, not published.

Certain commercial equipment and software are identified in this paper in order to adequately specify the experimental facility. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment or software identified are necessarily the best available for the purpose.