

# LIBRARY Michigan State University

This is to certify that the

thesis entitled

GENERATION, VERIFICATION, AND CORRECTION OF NUMERICAL CONTROL TOOL PATHS

presented by

Daniel Arthur Wysocki

has been accepted towards fulfillment of the requirements for

M.S.\_\_\_\_degree in <u>Mechanical</u> Engineering

Major professor

Date\_November 5, 1987

**O**-7639

MSU is an Affirmative Action/Equal Opportunity Institution





# GENERATION, VERIFICATION, AND CORRECTION OF NUMERICAL CONTROL TOOL PATHS

By

Daniel Arthur Wysocki

A THESIS

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

1987

### ABSTRACT

# GENERATION, VERIFICATION, AND CORRECTION OF NUMERICAL CONTROL TOOL PATHS

By

Daniel Arthur Wysocki

Numerical Control (NC) milling is a very expensive procedure when machining errors are not detected during the generation and verification process of the Cutter Location (CL) file. This thesis presents a three step process in the generation, verification, and, if necessary, correction of a tool path. The first step uses chordal deviation techniques to generate a tool path with a minimum number of CL points. The second involves tool path correction when undercutting occurs in concave (inside corners) and convex (outside corners) regions of the tool path. If a more extensive verification is desired, a third process is used with verification algorithms developed by Oliver [5] for three-axis NC machining centers. These algorithms take a three-dimensional look at the tool path versus the desired surface geometry. If an undercut occurs, a correction is made in the tool z-axis direction. An example is given illustrating each of the three processes mentioned above.

To wife Mary, my parents Art and Cassie, and my sister Sharon (The Divine Miss W.),

for the love

and support

they always give me.

#### ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Erik Goodman, my major professor, Emery Szmrecsanyi and Mukesh Gandhi, the other members of my committee, for providing me with the support and confidence when I needed it the most. Their time and guidance is greatly appreciated and has enhanced the content of this thesis considerably.

I would to thank Jack Withrow, Glen Bibbins and the Engineering Systems Development Department at Chrysler Motors for the funding, time and technical support they provided during this research.

I would also like to thank Jim Oliver for all the profitable discussions we had in the area of Numerical Control (NC) Milling. His dissertation work gave me a great deal of insight in the area of NC verification.

To my friends, especially Maureen Agar, for providing me the made-to-order thesis formats, and Sharat Shetty, for always entertaining my crazy ideals, I thank you.

But most of all, a special thanks has to go to my wife, Mary, my parents, Art and Cassie, my sister, Sharon, and to God, for making this all worth while.

# TABLE OF CONTENTS

# Page

| LIST OF FIGURES                                       | •  | • | v  |
|---|----|---|----|
| Chapter 1 – INTRODUCTION                              | •  | • | 1  |
| Chapter 2 - BACKGROUND ON NC METHODS AND VERIFICATION | ON | • | 2  |
| Chapter 3 – TOOL PATH GENERATION                      | •  | • | 11 |
| Chapter 4 – TOOL PATH VERIFICATION AND CORRECTION .   | •  | • | 21 |
| Chapter 5 - RESULTS AND DISCUSSIONS                   | •  | • | 31 |
| Chapter 6 - CONCLUSIONS                               | •  | • | 37 |
| LIST OF REFERENCES                                    | •  | • | 39 |

# LIST OF FIGURES

•

| Figure | 2.1 | Surface parameterization problem                                   | 3  |
|--------|-----|--|----|
| Figure | 2.2 | Tool path gouging against an adjacent surface                      | 4  |
| Figure | 2.3 | Corrected tool path through the use of offset surfaces             | 4  |
| Figure | 2.4 | Tool path gouging surface being cut                                | 5  |
| Figure | 2.5 | Tool path corrected for gouging against the surface being cut.     | 6  |
| Figure | 2.6 | Chordal tolerances for linear interpolation                        | 7  |
| Figure | 2.7 | Flow diagram of tool path generation, verification, and correction | 10 |
| Figure | 3.1 | Local curvature of higher-order polynomial surfaces                | 12 |
| Figure | 3.2 | Chordal deviation with surface points                              | 13 |
| Figure | 3.3 | CL points with corresponding normals on a surface                  | 14 |
| Figure | 3.4 | Chordal deviation with surface normals                             | 15 |
| Figure | 3.5 | Tool pass with an excess number of CL points                       | 17 |
| Figure | 3.6 | Tool pass with a minimal number of CL points                       | 17 |
| Figure | 3.7 | Binary search for good surface points                              | 18 |
| Figure | 3.8 | Cusps or scallops between tool passes                              | 18 |
| Figure | 3.9 | Geometry for cusp height calculation                               | 19 |
| Figure | 4.1 | Undercutting concave and convex regions of a surface               | 21 |
| Figure | 4.2 | Undercutting of an inside corner of a surface                      | 22 |
| Figure | 4.3 | Tool center geometry for undercutting check                        | 22 |
| Figure | 4.4 | Geometric configuration for butterfly interpolation                | 24 |
| Figure | 4.5 | Geometry for undercutting an outside corner of a surface           | 26 |

|            |   | Page |
|------------|---|------|
| Figure 4.6 | Inserted CL point to prevent undercutting                     | 27   |
| Figure 4.7 | Undercutting caused by the side of the tool                   | 28   |
| Figure 5.1 | Ruled surface with two inch cusp to be milled                 | 31   |
| Figure 5.2 | Sharp corners to help magnify the undercut problem            | 32   |
| Figure 5.3 | Three tool paths generated by chordal deviation methods       | 33   |
| Figure 5.4 | Elimination of inside and outside corner undercuts            | 33   |
| Figure 5.5 | Tool path spacing by cusp height                              | 34   |
| Figure 5.6 | Superimposed tool paths before and after Oliver's algorithm . | 35   |
| Figure 5.7 | "True" view of superimposed tool paths                        | 35   |

#### Chapter 1

#### INTRODUCTION

Although Numerical Control (NC) processes have been around since the 18th century in the Falcon (1728) and Jacquard looms (1800) [1], it has only been recently, the past four decades, that these processes were really integrated into a manufacturing environment. Precision components with complex shapes whose overall costs to manufacture were very high were virtually the only application of NC up to this time.

With the development of the computer during the past two decades, these costs were brought down and have allowed these NC processes to be expanded into many new areas such as the machining of free form or sculptured surfaces. Besides providing a more efficient way to generate cutter location (CL) files, computers have given the manufacturing sector a tool to help verify the goodness of a CL file and correct it if necessary. This process to date has not generally been automated and this research is a step in the direction of trying to make this happen.

#### Chapter 2

## BACKGROUND ON NC METHODS AND VERIFICATION

Machining of sculptured surfaces with three-, four-, and five-axis NC machining centers is being used in a wide variety of industries such as ship building, aerospace and automotive. The accuracy of the machined surfaces is critical in these industries. Examples of this are the areas of air flow over an airplane wing and the face of a gear tooth in an automobile transmission.

In all these industries, ballnose-type cutters are typically used when machining sculptured surfaces. Parametric surfaces are the most commonly used specification of sculptured surfaces for NC. For such surfaces, constant u or v parametric flow lines, where three-dimensional surface points are represented as a function of u and v, are the most common methods used to generate tool paths. However, there are two major problems with this method [2] -- surface parameterization being the first and tool placement the second, as explained below.

It is very difficult to machine a surface when the actual stepover distance at the top is larger or smaller than the distance at the bottom, even though the parameterization is equal at both ends. Sometimes this is referred to as the triangular patch problem. It causes areas on the surface to be either missed or cut more than once during the machining process, depending upon the tool path stepover distance. See Figure 2.1.

2



Figure 2.1 Surface parameterization problem.

The second problem deals with the placement of the tool. With a correct theoretical geometric placement of the tip of the cutter, gouging may occur on the surface being machined or on an adjacent surface. With the tool placement being correct at the tip of the tool, gouging may still be caused by other parts of the tool as it moves from point to point. See Figure 2.2. Gouging against an adjacent surface can be corrected by offsetting the surfaces of concern by the distance of the cutter radius and using the offset intersection as a limiting point or check surface for the tool path being generated. See Figure 2.3.



Figure 2.2 Tool path gouging against an adjacent surface.



Figure 2.3 Corrected tool path through the use of offset surfaces.

Gouging occurring on the surface being cut is harder to locate and correct because the intersecting point of the "butterfly" in the cut might not exist. This butterfly occurs when a tool is too large to machine an area, such as a corner or a cusp of a surface, without gouging. It can be shown graphically by tracing the center point path of the cutter along a tool path. The butterfly starts where the gouging first occurs and ends where the tool no longer violates the surface. This point is depicted by A in Figure 2.4. Since Figure 2.4 is illustrated in a two-dimensional form, a true intersection is shown. Surfaces, however, are three-dimensional and the normals describing the tool center path can be skewed. Thus, a true butterfly intersection point may not exist, but gouging can still occur.



Figure 2.4 Tool path gouging surface being cut.

In either case, these violation areas can be minimized by either selecting a smaller tool or looking for the extreme points where the surface can be cut without gouging. This can be done by finding the "fictitious butterfly," where the gouging begins and ends, and eliminating all points in the cut within this region. Minimum

distance techniques can be used to find these points (see Chapter 4). Results would be equivalent to the ones shown in Figure 2.5. The surface has not been fully machined at this point because areas within the butterfly have yet to be cut. Smaller radius cutters, thus, can be used to machine these areas.



Figure 2.5 Tool path corrected for gouging against the surface being cut.

An exact definition of the surface cannot be machined without using an infinite number of CL points. Since this is impossible, linear interpolation is commonly used to approximate curves along a surface. Chordal tolerances as small as .0001 inch (.003 mm) can typically be programmed. See Figure 2.6. In addition to linear interpolation, some NC machines can handle higher orders of interpolation such as circular and parabolic [3]. When the step increments are small, surface curve approximations can be made through use of osculating circles or local curvatures [4]. Approximations of this type are dependent upon a well-behaved surface being present. Surfaces of higher orders can produce misleading and incorrect results because of the small oscillations that can occur

with higher-order equations. These are the most commonly used techniques to generate tool paths; they don't, however, guarantee that the surface will be properly milled.



Figure 2.6 Chordal tolerances for linear interpolation.

For years, NC verification was performed by two methods, experimental trial or proofing runs and visual checking of two-dimensional drawings of generated CL points [5]. Proofing runs proved to be very time consuming and costly, especially when coordinate-axis measuring machines are used to verify the model. Visual checking is effective in identifying large milling errors, but not thorough enough for a final verification. Proofing runs often followed visual inspections because of this.

As computers keep getting faster and faster, the problem of automating the NC verification process, with its enormous amounts of mathematical calculations, is becoming more manageable.

SURFAPT [6] is an example of one of the many NC programs on the market to utilize these faster central processing unit (cpu) times. Checking is performed visually via watching the simulation on the graphics monitor. This technique is strictly view dependent and will only allow detection of machining errors visible in the view displayed on the graphics monitor. A user must be very knowledgeable in the part being programmed and NC in order to strategically pick the views from which a visual check should be made to detect possible NC machining errors, and must be able to recognize an error when it occurs.

An area in which a very large amount of energy is being spent to solve the NC verification problem is solid modeling. Voelcker and Hunt [7] [8] were among the first to suggest this. The mathematical calculations involved with solid modeling, however, are enormous and have made this process not a very desirable one at this time.

Roth [9] and Atherton [10] have proven in their work that solid modeling can be used in a more efficient manner for NC verification. Their techniques, however, were of a visual nature and the problems with visual verification were discussed earlier.

Wang [11] seems to have produced some better results. Wang is claiming to have the full capability of NC verification for five-axis milling applications via solid modeling techniques.

Oliver [5] [12] [13] has developed a procedure using either surface or B-rep solid modeling representations which has produced results in an O(N) time, where N is the number of tool path points, which compares to  $O(N^4)$  time complexity for direct solid modeling. This work differs sharply from the "traditional" solid modeling simulation approach using sequential boolean operations.

Oliver's research will verify whether a surface was properly milled and will color code the areas that were missed or gouged. The only view dependency is that machining errors will only be found in the areas visible in the view selected. Various views can be used to verify the entire NC machining process.

There is some confusion between simulation and verification. Simulation does not notify a programmer if an error occurs. The programmer has to identify the error as it happens on the replay of the machining process and recognize it as an error. If an error is displayed and the programmer misses it, the process is assumed to be correct. Verification, on the other hand, notifies the programmer where the machining error has occurred by color coding or some other means.

Most of the techniques mentioned above are ways that will only detect, but not correct, milling errors when machining a sculptured surface.

This research presents a method in which the process of generating a cut, verifying, and correcting it if necessary, can be automated. The process is divided into three procedures: tool path generation through chordal deviation techniques, tool path correction of undercutting caused by the tool along the tool path, and three-dimensional tool path verification and correction in the tool z-axis direction (See Figure 2.7).



Figure 2.7 Flow diagram of tool path generation, verification, and correction.

## Chapter 3

#### TOOL PATH GENERATION

One of the major objectives of this research was to develop a NC verification process which was independent of the surface type being evaluated. CL points, surface points, and normals are the only information needed for the procedure. Surfaces with higher-order polynomial equations tend to have small oscillations in them. These oscillations are generally numerical artifacts rather than desired features of the surface design. Therefore, if the CL points pass a series of chordal deviation checks, which are defined in this chapter, there is no need to be concerned with local surface curvature. NC machines cut on a point-to-point basis and not on local curvature. If a point and normal are correct, they can be used. See Figure 3.1. Therefore, local surface curvature evaluations are not necessary in any of the NC verification algorithms presented here.



Figure 3.1 Local curvature of higher-order polynomial surfaces.

In the work presented here, as in most surface machining codes, the CL points are generated along a constant u or v surface parameter direction. If a certain number of cuts is specified, for example, in the constant v direction, the distance between the cuts is incremented in u,v space by a distance of d, where

$$d = \frac{\text{maximum u of surface} - \text{minimum u of surface}}{\text{number of cuts} - 1}$$
(3.1)

The first cut is usually started at the minimum u surface parameter and gradually incremented until the maximum u surface parameter is reached. Each new cut is defined by the constant u parameter  $u_{new}$ . If ten cuts are requested, the desired cut mentioned in equation 3.2 is equal to 1 for the first cut, two for the second, etc.

$$u_{new} = minimum u of surface + d(desired cut - 1)$$
 (3.2)

Two types of chordal deviation are used to calculate whether a CL point is good or not. The first deals with three calculated points on a surface. A chordal height is found between the two outer points. The second involves distance checking between the lines formed by two surface points and two corresponding tool center points with respect to the radius as explained below.

Given the three points P1, P2 and P3 (see Figure 3.2), the chordal height, H1, is equal to the minimum distance from P3, which is at the u,v midpoint between P1 and P2, to the line formed by P1 and P2. This distance lies in the plane determined by P1, P2, and P3, which does not necessarily contain the locus of the tool axis between P1 and P2. P3 is chosen to be the parametric midpoint, where it is assumed that the chordal deviation is near its maximum. In other words, the assumption is made that the surface is fairly well-behaved with at most a gradual change in curvature between points P1 and P2.



Figure 3.2 Chordal deviation with surface points.

Assume chordal deviations have been within tolerance up to some point P1 on the surface, and a possible cut to P2 is being examined. If the chordal height H1 is greater than the specified chordal tolerance, P2 is considered a bad point and a new P2 and P3 must be recalculated at a smaller u,v distance away from P1. If H1 is within the specified chordal tolerance, P2 is considered a good point. The distance between P1 and P2 should keep increasing until a bad P2 point is found, as later described. This, in effect, will create a tendency to spread CL points out along flat areas and bunch them closer together along the curved areas of a surface (see Figure 3.3).



Figure 3.3 CL points with corresponding normals on a surface.

In addition to the chordal deviation check mentioned above, a second chordal deviation check using the radius of the ballnose cutter can be used. Two additional points, PN1 and PN2, are calculated (see Figure 3.4). PN1 and PN2 are points along the normal vector a distance equal to the radius of the cutter from points P1 and P2 respectively. First the minimum distance, Dmin, between the line segments

L2 and L1 formed by the two surface points, P1 and P2, and the points PN1 and PN2 respectively must be calculated. The minimum distance point on L2 must also be constrained to intersect L2 between the points P1 and P2. In general, these lines, L1 and L2, are not coplanar.



Figure 3.4 Chordal deviation with surface normals.

Then the chordal height, H2, is defined as

H2 = radius of cutter - Dmin between the points P1 and P2. (3.3)

To find the total chordal deviation between P1 and P2, it must be decided if the local curvature between these points is concave or convex. This can easily be determined by using the magnitudes of lines L1 and L2. If |L1| is greater than or equal to |L2|, the surface is convex and if |L2| is greater than |L1|, the surface is concave. For a convex surface, the total chordal deviation is equal to the sum of H1 and H2 and for a concave surface, it is equal to the greater of the two values of H1 and H2. If the total chordal deviation is greater than the specified chordal tolerance, P2 is considered a bad point and must be recalculated.

Increment doubling and binary search is used to find both good and bad P2 points along the cut. Several techniques have been used to save compute time when looking for good P2 points.

The first technique is that P3 only has to be calculated until the first successful P2 is found. When looking for a good P2 at a distance further away from P1, a doubling of the last successful u,v increment is made until a bad P2 or the end of the surface is reached. As the distance keeps doubling, each new P3 is equal to the previous P2. This is one way to save on compute time with the recalculation of P3.

The binary search technique really becomes efficient when a good P2 is followed by a bad one. At this point, a finite region has been established where a final good P2 can be found. It is between the midpoint, P3, and the current bad point, P2. The next calculation will narrow this region down considerably.

Convergence on the outermost P2 from P1 must continue for one reason – to minimize the number of CL points in the tool path. If the method is halted after the first bad P2 is found, a larger than necessary number of CL points is inserted into the CL file. See Figure 3.5. If the method is continued, the number of CL points is minimized. See Figure 3.6.



Figure 3.5 Tool path with an excess number of CL points.



Figure 3.6 Tool path with a minimal number of CL points.

Let P2new be the next calculated point in the binary search. If P2new is defined as a good point, the next P2new will be calculated in region B. If it is a bad point, it will be in region A. See Figure 3.7. The trick here is that the regions have been narrowed down to a small enough area that a "permanent" midpoint can be assigned. If the search is in region A, P3old is used as the permanent midpoint, and if in region B, P3 is used. These fixed midpoints are not the true midpoints in u,v space, but they are close enough that the savings in compute time are worth the small errors in chord height which are introduced. When the final P2 point is found, a true u,v midpoint can be calculated to verify the chordal tolerance test. If a more accurate chordal tolerance must be held, P3 can be recalculated as the actual true midpoint during the binary search procedure in regions A and B.



Figure 3.7 Binary search for good surface points.

There is a small amount of material left on the surface being machined between consecutive tool paths, which is referred to as cusps or scallops. The height of these cusps or scallops should be minimized or kept under a pre-defined tolerance. See figure 3.8.



Figure 3.8 Cusps or scallops between tool paths.

When positioning an adjacent tool path, these cusp heights are calculated in the following manner. Corresponding tool centers on two adjacent tool paths must be within one cutter diameter distance of each other. If this is not true, a smaller stepover distance must be used. Otherwise, areas of the surface will not be machined. Calculate P1, P2, P3, PN1 and PN2 in the same manner mentioned earlier. PM is the midpoint between PN1 and PN2. Create the plane, PL1, with the three points PN1, PN2 and P3. Next project P1 and P2 into plane, PL1. See Figure 3.9.



Figure 3.9 Geometry for cusp height calculation.

Check and see if the projected points  $P1^{-}$  and  $P2^{-}$  and the point P3 are collinear. If they are, a flat surface cusp height approximation can be made:

cusp height = (3.4)  
radius of cutter - 
$$\sqrt{(\text{radius of cutter})^2 - (\text{distance from PM to PN2})^2}$$

If they are not collinear, find the radius, R2, of the circle formed by the intersection of the two cutter spheres by the Pythagorean theorem.

R2 = 
$$\sqrt{(\text{radius of cutter})^2 - (\text{distance from PM to PN2})^2}$$
 (3.5)

The cusp height can now be calculated as,

$$cusp height = (distance from P3 to PM) - R2.$$
(3.6)

Arc length may vary along the constant surface parametric flow lines because of the surface parameterization problem described in Chapter 2. The surface parametric distance may be equal at the top and bottom of the surface, but the arc length may vary. See figure 2.1. Therefore, a few samples should be taken along the surface when machining along constant parametric flow lines to maintain the desired cusp height. For instance, if a surface is being machined along constant u parametric lines, samples can be taken at v = 0, .25, .5, .75, and 1.0 if the parameterization in the v direction of the surface is from 0.0 to 1.0. The smallest u,v stepover sample within the cusp height tolerance is used to assure the cusp height is not likely exceed its defined tolerance. Many areas between tool paths will contain cusps that are well within the tolerance. This is okay, because the objective is to machine the surface so that the cusps or scallops are equal to or less than the cusp height tolerance.

Related work dealing with the topics discussed in this chapter can be found in the following references: [6] [14] [15].

#### Chapter 4

### TOOL PATH VERIFICATION AND CORRECTION

Chordal deviation checks, such as those mentioned in Chapter 3, do not guarantee that a tool will not undercut or gouge the surface being machined. There are two situations where undercuts could occur. The first happens when the cutter is too large to properly mill concave regions, such as sharp corners, of a surface. These regions are depicted as A in Figure 4.1. The second deals with the convex regions such as outside corners. See region B in Figure 4.1.



Figure 4.1 Undercutting concave and convex regions of a surface.

A sequence of four CL points is needed to make the check and correction for the undercutting of an inside corner of a surface as denoted in region A. The "butterfly" or the point at which the undercutting begins and ends will occur between the three-dimensional line segments P1P2 and P3P4 respectively. See Figure 4.2. P2 and P3 do not have to be adjacent CL points in the tool path, but will converge upon each other as the corner becomes sharper or comes to a point.



Figure 4.2 Undercutting of an inside corner of a surface.

Figure 4.3 shows the definitions used in discussing the finding of the butterfly. Four points PN1, PN2, PN3, and PN4, are created at the tool center positions relative to P1, P2, P3, and P4. These tool center points are a radius distance away from the surface points along the corresponding normals.



Figure 4.3 Tool center geometry for undercutting check.

To find the areas of undercutting, set P2 equal to the second point and P3 equal to the second from last point of the tool path. Iterate P3 backwards along the cut until the distance between P2 and PN3 or P3 and PN2 is less than the radius of the cutter. At this point a butterfly will occur between the two three-dimensional line segments P1P2 and P3P4. Interpolate to the intersection point of the butterfly (this interpolation process is described below) and eliminate all CL points in between the start and end points of the butterfly. If the distance check does not fail, set P2 equal to the third point of the tool path and start the iteration process again with P3 equal to the second from last point of the tool path. This sequence is performed until all possible iterative combinations have been exhausted. If a butterfly is not encountered, the maximum number of iterations is equal to (n-3)!, where n is equal to the number of CL points in the tool path. However, if a butterfly is encountered, and CL points within the tool path between P1 and P2 are eliminated, the total number of iterations will be less than (n-3)!.

When the distance check mentioned above fails, the butterfly intersection point can be found by using the following interpolation method. Refer to Figure 4.4.

Form the following vectors (the lines formed by PN1, PN2 and PN3, PN4 respectively are most likely not coplanar):

$$A = PN1PN2 \tag{4.1}$$

$$B = PN3PN4. \tag{4.2}$$

Next, find the vector, C, normal to vectors A and B.

$$\mathbf{C} = \mathbf{A} \mathbf{x} \mathbf{B} \tag{4.3}$$

Find the vector, D, normal to vectors A and C.

$$\mathbf{D} = \mathbf{A} \times \mathbf{C} \tag{4.4}$$

Form the plane, PL1, normal to vector D at PN1 and containing the vector A.

$$(\mathbf{P} - \mathbf{PN1}) \bullet \mathbf{D} = \mathbf{0} \tag{4.5}$$



Figure 4.4 Geometric configuration for butterfly interpolation.

To find the approximate butterfly minimum distance point PN3' (intersection point in the planar case) between points PN3 and PN4, intersect the line PN3PN4 with plane PL1. The intersection point is PN3'. This process will also produce the parametric value, T3, which represents [PN3'-PN3] / [PN4-PN3]. Form a line in u,v parametric space between P3 and P4. Substitute T3 into the equation of this line to get a new interpolated u,v point, U3' and V3'. Evaluate this new surface point, P3' and form the new tool center point, PN3". PN3' and PN3" will most likely not be coincident because of the parametric spacing problem described in Chapter 2. PN3", however, will be much closer to the butterfly intersection point than PN3 is.

This process can be followed to find PN2" by substituting equation 4.6 for 4.4 and 4.7 for 4.5.

$$\mathbf{D} = \mathbf{B} \times \mathbf{C} \tag{4.6}$$

$$(P - PN3) \bullet D = 0$$
 (4.7)

PL1 will now be normal to the vector D at PN3 and contain the vector B. PN2' is be found by intersecting the line PN1PN2 and PL1. PN2" is found by the same process mentioned above to find PN3".

Find the distance between PN2" and PN3". If they are not coincident, substitute PN2" for PN2 and PN3" for PN3 and repeat the entire interpolation process. After a few interpolations, PN2" and PN3" will converge upon each other or upon the minimum distance points between the two lines PN1PN2 and PN3PN4. PN2" and PN3", at this point, are the start and end points of the butterfly in question.

Undercutting of outside corners is detected and corrected in a manner similar to that discussed above for inside corners. Four surface points, P1, P2, P3, and P4, are needed for the undercut check.

For an undercut to occur on an outside corner of a surface, the following conditions must be met (refer to Figure 4.5):

$$|P2P3| < radius of cutter,$$
 (4.8)

| PN2PN3 | > | P2P3 |, and (4.9)

H2 > pre-defined chordal height tolerance. (4.10)



Figure 4.5 Geometry for undercutting an outside corner of a surface.

Condition 4.9 indicates the surface normals at P2 and P3 are pointed in an outward direction away from the surface points. H2 in 4.10 gives the amount of undercutting caused by tool path motion from CL points P2 to P3. H2 is calculated in the same manner as was mentioned in equation 3.3.

To machine the surface correctly without undercutting the surface, a CL point, P5, and its associated normal, must be inserted into the tool path. See Figure 4.6. P5 will allow the cutter to clear the outside corner of the surface while it proceeds along the other side, from P2 to P3, without undercutting or gouging the corner at P2 and P3. P5 is calculated from the theoretical intersection point of the three-dimensional lines PN1PN2 and PN3PN4.



Figure 4.6 Inserted CL point to prevent undercutting.

PN5 is the theoretical intersection point of the two lines mentioned above. Being that these lines are three-dimensional, they may be skewed. If they are, a true intersection point will not exist. PN5 is then equivalent to the midpoint of PN2' and PN3'. PN2' and PN3' are calculated in the same manner as they were for the inside corner problem mentioned above. The one difference is that PN5 is calculated only from PN2' and PN3'. Since there is not a surface parametric spacing problem, as there was with the inside corner problem, only one iteration is needed to find the exact solution. P2', P3', PN2", PN3", and T3 are also not needed for the calculation; therefore, the iteration process can be stopped at this point during the initial pass.

To find P5, form a vector from PN5 to the midpoint found between the points P2 and P3. This vector will give the direction where P5 can be found in relation to

PN5. P5 is located a radius distance from PN5 in this direction. The CL point associated with P5 and the direction vector formed by P5 to PN5 is then inserted into the tool path between CL points associated with P2 and P3. Therefore, the outside surface corner formed by the points P1, P2, P3, and P4 will not be gouged because of the insertion of point P5.

The method just described detects and corrects undercutting caused by a ballnose cutter on a surface with reference to the surface points at the tip of the tool. The algorithms, however, are only two-dimensional with respect to the cutter. They do not have the capability to detect if the side of the tool is violating the surface even when the tool path at the center is correct. See Figure 4.7.



Figure 4.7 Undercutting caused by the side of the tool.

To solve this problem of gouging caused by the side of the tool, verification algorithms developed by Oliver [5] [12] [13] are implemented. Oliver's algorithm is a new application of surface and solid modeling technology. Oliver's goal was to minimize unnecessary computations when checking tool paths against the desired part geometry. In other words, Oliver has developed an interactive algorithm to verify NC tool paths against part geometry while outputting the magnitudes of the undercuts as they occur during three-axis milling operations. This research will use the results generated by these NC verification algorithms to correct undercutting caused by tool path errors. Therefore, Oliver's research is the final step in automating the process of generating, verifying, and correcting NC tool paths of three-axis milling operations.

The necessary input for Oliver's NC verification algorithms are as follows; surface geometry to verify against, discrete grid size of the surfaces, radius of cutter, height of cutter, and successive pairs of tool CL points.

The discretization process is a grid network of surface points and normals from which to make the NC verification check. The accuracy of the results is limited by the grid size. The denser the surface grid, the greater the accuracy. The denser grid also causes an increase in computational time, so optimizing the grid size will optimize the response time. There are three choices currently used by this research; a default grid size, a user-specified grid size, or a grid size generated relative to the size of the cutter. A grid size generated with respect to the cutter will compute a pre-defined number of grid points for every cutter diameter distance on the surface.

A decision making process must be formulated to direct the tool cutter to mill the surface properly when a violation is detected by Oliver's NC verification algorithms. The algorithms provide the necessary information to determine which side of the cutter is actually violating a surface and by what magnitude. A decision is then made in which direction to move the cutter to prevent this violation. As a first step, this research is only directing the cutter to move in the z-axis direction with respect to the cutter. Moving the cutter in this direction will cause areas of the surface to be missed during the machining process. These missed areas are marked and used as a starting point for another tool-path to be generated with a smaller radius cutter.

Related work dealing with the topics discussed in this chapter can be found in the following references: [2] [16] [17].

#### Chapter 5

### **RESULTS AND DISCUSSIONS**

Chapters 3 and 4 presented a method on how the generation, verification, and correction, if necessary, of tool paths can be automated in NC three-axis milling procedures. This chapter will illustrate some results of these methods.

The surface to be machined is a ruled surface ten inches long by seven inches wide. A two inch high cusp exists in the middle and fades out along the width of the surface. See Figure 5.1.



Figure 5.1 Ruled surface with two inch cusp to be milled.

The corners of the cusp are sharp to help magnify the undercuts which will occur when machining these areas. The tool or cutter will be of a ballnose-type, five inches in diameter. Tool paths will be generated in the direction across the cusp. See Figure 5.2.



Figure 5.2 Sharp corners to help magnify the undercut problem.

When machining in the direction across of the cusp, undercutting occurs at each of the sharp corners. Chordal deviation methods described in Chapter 3 cannot prevent the undercutting. They can, however, space the CL points in an efficient manner to detect them. CL points along the tool path are minimized with respect to the surface curvature which is found through the chordal deviation checks. The spacing between the CL points is maximized by the chordal deviation method used with three surface points while the cusps are detected by the surface normal vector chordal deviation check. See Figure 5.3.



Figure 5.3 Three tool paths generated by chordal deviation methods.

By using the methods described in Chapter 4, the area of undercutting can be eliminated from the tool paths. Undercutting of inside corners is corrected by the elimination of the violating CL points and the insertion of two new CL points at the beginning and end point of the butterfly generated at the tool center path. Undercutting of outside corners is eliminated by the insertion of a CL point at the theoretical intersection of the two tool center point lines before and after the cusp. See Figure 5.4.



Figure 5.4 Elimination of inside and outside corner undercuts.

The number of cuts generated can be determined by two methods. The first is simply by specifying the number of cuts, as illustrated in Figure 5.3, and the second by specifying the cusp height and letting the cuts be spaced accordingly. In Figure 5.5, a cusp height of 0.002 inches is requested with a result of nine cuts or tool paths. The cuts may not be equally spaced because the distance between cuts is relative to cusp height and not stepover distance on the surface. Tool paths in Figure 5.5 are illustrated by the normals at each of the CL points.



Figure 5.5 Tool path spacing by cusp height.

Undercutting greater than the pre-defined tolerance may occur between tool paths. The methods illustrated above only guarantee that undercuts will not occur along a tool path within the pre-defined tolerance. These methods, however, are only a two-dimensional check and correction with respect to the cutter. The side of the cutter might be violating the surface. NC verification algorithms developed by Oliver [5] are used to detect and correct these violations.

With the cusp fading out from one side of the surface to the other and with the size of cutter being used, undercutting occurs between the tool paths. Oliver's algorithms detected these undercuts with a discrete grid size of fifty by fifty. Maximum degree of gouging was found to be 0.075 inches. CL points violating the surface were adjusted along the tool z-axis direction. Figure 5.6 and 5.7 show the tool paths before and after the correction was made. Tool center paths are used to illustrate the tool paths because of the complexity of the CL file.



Figure 5.6 Superimposed tool paths before and after Oliver's algorithm.



Figure 5.7 "True" view of superimposed tool paths.

Currently, there is only a one-way link between the two software packages (tool path generation and correction, and Oliver's NC verification). The tool path generation and correction software can pass as input the tool motions to Oliver, but it cannot interpret the results that are returned. Therefore, the results obtained from Oliver are interpreted outside the two software packages. These analyzed results are used to correct the tool motions by hand in the tool z-axis direction.

# Chapter 6

#### CONCLUSIONS

This thesis has demonstrated a method to automate the generation, verification, and correction of tool paths for three-axis NC milling procedures. The method is divided into three segments; generation through chordal deviation techniques, correction of undercuts of inside and outside corners, and three-dimensional verification through Oliver's algorithms. Algorithms for the generation and correction of cuts along a tool path were kept simple for fast computational time. They are also independent of the surface type. They have been developed in a manner that is independent of the number of surfaces that can be processed. In other words, a tool path can extend across an arbitrary number of surfaces of various types, because the algorithms are only concerned with surface points and normals and not the type of surface. Therefore, this makes the method very attractive for the manufacturing sector.

In an example, tool paths were generated on a seven inch wide by ten inch long ruled surface. A two inch cusp existed on one side of the surface and faded out along the width of the surface to the other side. Tool paths were generated in the direction across from that of the cusp using a five inch diameter ballnose cutter. Chordal deviation techniques allowed the tool paths to be generated with a minimal amount of CL points. Undercutting detection and correction algorithms found and eliminated gouging caused by the tool in the tool path direction while

37

verification algorithms developed by Oliver helped detect and correct areas of gouging not detected by these computational fast means.

Future development will include a more sophisticated approach in determining tool path spacing when a cusp height is specified. By taking a more global look at the surface, undercut or violation areas can be avoided by proper tool path spacing. Expansion of Oliver's algorithms from its present state of three-axis NC verification to five-axis will render it more useful. Finally, a tree-structured decision making process is needed to correct tool path violations when detected by Oliver's algorithm. In other words, instead of going over a violation area, find a way around it.

#### LIST OF REFERENCES

- Bezier, P., Numerical and Controls Mathematics and Applications, John Wiley and Sons, 1970.
- Bobrow, J.E., "NC Machine Tool Path Generation From CSG Part Representations," Computer Aided Design, Vol. 17, No. 2, March, 1985.
- Drozda, T.J. and Wick C., Tool and Manufacturing Engineers Handbook, Vol. 1, Machining, Society of Manufacturing Engineers, 1985.
- 4. Faux, I.D. and Pratt, M.J., Computational Geometry for Design and Manufacture, Ellis Horwood, 1979.
- Oliver, J.H., Graphical Verification of Numerically Controlled Milling Programs for Sculptured Surface Parts, Ph.D. Dissertation, MIchigan State University, 1986.
- Mackin, N., Machining of Complex Surfaces with SURFAPT, SAE paper 870897.
- Voelcker, H.B. and Hunt, W. A., "The Role of Solid Modelling in Machine– Process Modelling and NC Verification," *Proceedings of SAE 1981 International Congress and Exposition*, Detroit, Michigan, February, 1981.
- Hunt, W.A. and Voelcker, H.B., "An Exploratory Study of Automatic Verification of Programs for Numerically Controlled Machine Tools," Production Automation Project, Technical Memo No. 34, University of Rochester, 1982.
- 9. Roth, S.C., "Ray Casting for Modeling Solids," Computer Graphics and Image Processing, 18, February, 1982.

- Atherton, P.R., "A Scan-Line Hidden Surface Removal Procedure for Constructive Solid Geometry," Computer Graphics, Proceedings of SIGGRAPH, Vol. 17, No. 3, July, 1983.
- Wang, W.P., "Integration of Solid Geometric Modeling for Computerized Process Planning," Computer-Aided/Intelligent Planning - PED, Vol. 19, Book No. G00334, American Society of Mechanical Engineers, 1985.
- Oliver, J.H. and Goodman, E.D., "Color Graphic Verification of N/C Milling Programs for Sculptured Surfaces," 10th Annual ESD/SMI Automotive Computer Graphics Conference and Exposition, Engineering Society of Detroit, Detroit, Michigan, December, 1985.
- Oliver, J.H. and Goodman, E.D., "Computer Verification of Numerical Control Programs for Sculptured Surface Parts," 1st International Conference Computers in Design, Manufacture and Operation in the Automotive Industry, Geneva, Switzerland, March 10-12, 1987.
- Sata T., Kimura F., Okada N. and M. Hosaka, "A New Method of NC Interpolation for Machining the Sculptured Surface," Annals of the CIRP, Vol. 30/1/1981.
- 15. Loney, G.C., "Interactive Design and NC Machining of Free Form Surfaces," Master's Thesis, Lehigh University, PA, USA, December 1985.
- Chen, Y.J., Ravani B., "Offset Surface Generation and Contouring in Computer-Aided Design," Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 109, March 1987.
- "System/370 Automatically Programmed Tool Advanced Contouring Numerical Control Processor: Program Reference Manual", Program Number 5740-M53, International Business Machines Corporation, Second Edition, 1982.

- 18. Faux, I.D. and Pratt, M.J., Computational Geometry for Design and Manufacture, John Wiley and Sons, 1979.
- 19. Bowyer, A. and Woodwark, J., A Programmer's Geometry, Butterworths, 1983.

