Int J Adv Manuf Technol (1994) 9:281–290

The International Journal of
Advanced Manufacturing Technology

Non-Constant Parameter NC Tool Path Generation on Sculptured Surfaces

Yunching Huang and James H. Oliver

Department of Mechanical Engineering, Iowa Center for Emerging Manufacturing Technology, Iowa State University, Ames, IA 50011, U S A

An algorithm for three-axis NC tool path generation on sculptured surfaces is presented. Non-constant parameter tool contact curves are defined on the part by intersecting parallel planes with the part model surface. Four essential elements of this algorithm are introduced: initial chordal approximation, true machining error calculation, direct gouge elimination, and non-constant parameter tool pass interval adjustment. A software implementation of this algorithm produces graphical output depicting the tool path superimposed over the part surface, and it outputs cutter location (CL) data for further post-processing. Several applications examples are presented to demonstrate the capabilities of the algorithm. The results of this technique are compared to those generated from a commercially available computer-aided manufacturing program, and indicate that equivalent accuracy is obtained with many fewer CL points.

Keywords: Computer-aided manufacturing (CAM); NC milling; Orthogonal projection; Sculptured surface modelling; Tool path generation

1. Introduction

Automatic NC tool path generation systems typically produce cutter location (CL) data directly from a mathematical model of a product design. The cutter location data can then be post-processed into machine executable controlling codes for actual production. This capability integrates CAM activities with CAD systems to shorten product lead time and reduce the cost of production [1,2].

Various NC tool path generation algorithms have been described in the research literature, and many commercial CAD/CAM systems provide such capabilities. Tool path generation algorithms for parts designed with sculptured surfaces can be broadly characterised as either constant parameter or non-constant parameter techniques. Much of the current research focuses on the constant-parameter approach for locating tool contact curves on the part surfaces [3–5]. This approach is generally efficient because the tool contact curves are easy to retrieve from the surface definitions [6]. The drawback of this approach, however, is that the relationship between the parametric coordinate and the corresponding physical (Cartesian) coordinate is not uniform [7]. Therefore, the accuracy and efficiency of the constant parameter approach for tool path generation may vary depending on the geometry of the part surfaces. A typical example of this drawback is a "fan-shaped" surface shown in Fig. 1. On such a surface, constant parameter curves are close to one another at one end but further apart at the other. Of course, the curves could be generated to the desired accuracy at the wide end of the fan, but this would result in an unnecessarily large number of tool motions at the narrow end.

The non-constant parameter approach for NC tool path generation does not suffer from this problem. Typically, *cutting curves* are defined by the intersections of a group of

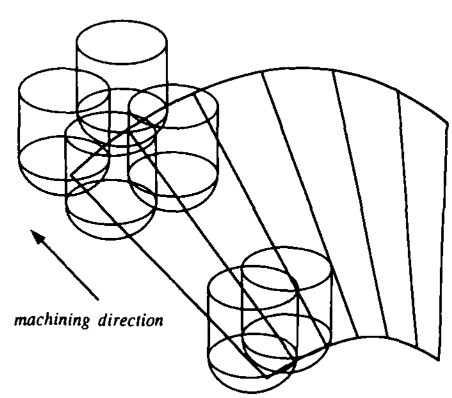

Fig. 1. Tool moves along constant parameter curves on a "fan-shaped" surface.

Correspondence and offprint requests to: Dr James H. Oliver, Department of Mechanical Engineering, Iowa Center for Emerging Manufacturing Technology, Iowa State University, Ames, IA 50011, USA.

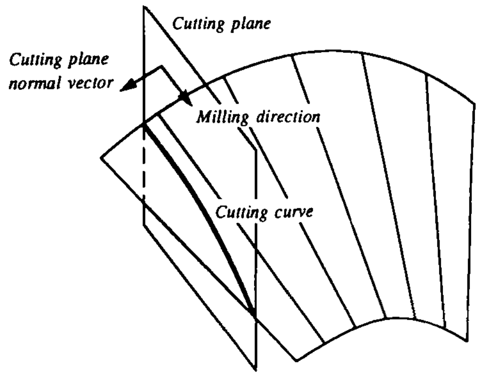

Fig. 2. Cutting plane and cutting curves definition.

parallel planes (*cutting planes*) and the part surfaces, as shown in Fig. 2. Users can define any tooling direction for each part without being limited by the parametric coordinate. Although such capabilities are provided in many commercial CAD/CAM systems, the actual algorithms are rarely published in the open literature. Furthermore, many systems with non-constant parameter generation capabilities are not sufficiently accurate for precise milling applications, or they suffer from heavy computational burden.

To address these issues, this paper presents an algorithm of non-constant parameter NC tool path generation that exploits the planarity of the cutting curves for maximum computational efficiency. This algorithm is introduced in four component techniques: initial chordal approximation, true machining error calculation, direct gouge elimination and tool pass interval adjustment. First, the initial chordal approximation method efficiently locates a good initial surface point to calculate the exact tool motion. Secondly, the true machining error calculation employs orthogonal projection to find the maximum actual machining error. This resulting error is then used to adjust the surface point in an iterative search. The direct gouge elimination technique works simultaneously with the above procedures. Finally, the tool pass interval adjustment technique calculates the maximum tool pass interval for a specified cusp-height tolerance. The details of each technique are discussed in later sections.

2. Machining Error

Normal machining error in multi-axis NC milling operation is due to the approximation of surface curves by linear tool motions [1]. Fig. 3 depicts three points that lie on a curve of a sculptured surface. If the curve is to be approximated by a line segment (chord) from point $\mathbf{P}_1$ to $\mathbf{P}_2$, the chordal deviation is defined as the maximum distance from the curve to the chord. In this paper, such deviation defined by the curve and the chord formed by a pair of cutter contact (CC) points, is referred to as the *nominal chordal deviation*.

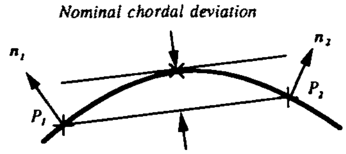

Fig. 3. Linear curve approximation.

A simple method to approximate the nominal deviation is by assuming that the furthest curve point from the chord is defined parametrically by half of the total parametric variation [5]. This technique is generally sufficient for surfaces with uniform parametric variation, but if the underlying surface definition is characterised by a non-uniform parametric variation, the error of this approach may be significant. Loney and Ozsoy [3] address this problem and provide a numerical method to solve for the parameter value that yields the maximum nominal chordal deviation for a constant parameter surface curve. This method is more accurate than the midpoint approximation, though it requires more computational effort. Both techniques start from a guess of the endpoint of the chord. The desired endpoint that yields an acceptable deviation is generally obtained via numerical iteration. For example, Loney and Ozsoy [3] use a curve subdivision technique, while Wysocki et al. [5] apply a cast-and-correct method based on a binary search.

Many tool path generation algorithms for sculptured surfaces assume that the nominal chordal deviation is the actual machining error [2,4]. This is true, however, only when the surface normal vectors at $\mathbf{P}_1$ and $\mathbf{P}_2$ (Fig. 3) are parallel, and both are perpendicular to the chord. Wysocki et al. [5] point out that the true machining error must be determined by considering the physical interference between the tool and the part surface, as shown in Fig. 4. Both the nominal chordal deviation and the distance between the tool centre trajectory and the corresponding chord between cutter contact points must be characterised to determine the true machining error. The calculation of both of these components is substantially complicated when non-constant parameter tool paths are considered.

A true machining error calculation method is thus developed for finding the machining error for non-constant parameter tool contact curves. This technique employs the orthogonal projection method [8] to calculate the exact distance between a tool motion and the part surfaces. Since this machining error calculation method is based on the physical interference between tool and part surface, it is more accurate than those methods based on nominal chordal deviation. In addition, by finding the longest linear motion that yields the specified machining tolerance, this technique effectively minimises the

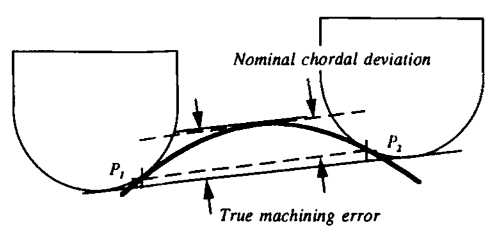

Fig. 4. Physical interference between tool and part surface.

total number of tool motions, and thus produces very efficient NC programs.

Since orthogonal projection is comparatively time-consuming in calculating machining error, the initial chordal approximation method is implemented to accelerate the process. The initial chordal approximation method provides a good initial point for the true machining error calculation to minimise the number of iterations and ensure convergence. These two techniques are combined to provide an efficient and accurate method for NC milling error calculation.

2.1 Initial Chordal Approximation

The initial chordal approximation method takes a different approach to other methods for calculating deviation. Instead of guessing an endpoint of the chord and then checking the deviation, this method first searches for the location of the climax that yields the specified deviation, then it calculates the endpoint of the chord. In addition, this method takes advantage of the planar geometry used for the non-constant parameter NC tool path generation.

To generate points on a cutting curve, the golden section method [9] is applied to intersect the cutting plane with the part surface. First, the surface boundary curves are searched to find the starting and ending points of the cutting curve, denoted by $\mathbf{P}_s$ and $\mathbf{P}_e$, respectively. Then one of the surface parameters, u or v is designated as the independent variable (denoted by w), based on the parametric difference between $\mathbf{P}_s$ and $\mathbf{P}_e$. If $|v_s - v_e| > |u_s - u_e|$ then $w = v$, otherwise $w = u$. In practice, to trace the cutting curve, w is incremented by some amount, and the other variable (u or v) is found by the golden section method.

As shown in Fig. 5, the starting point for the initial chordal approximation is denoted by $\mathbf{P}_1$, the nominal chordal deviation at $\mathbf{P}_r$ by d, and the endpoint of the chord by $\mathbf{P}_2$. The cutting curve, though it is not calculated explicitly, is denoted by $\mathbf{C}$, and the cutting curve tangent vector by $\mathbf{C}^t$. The goal of the initial chordal approximation method is to find the endpoint of the chord that yields the specified nominal chordal deviation.

Since the tangent at any point of the cutting curve is contained in the cutting plane and the tangent plane of the part surface, this indicates that $\mathbf{C}^t$ is perpendicular to both the cutting plane normal $\mathbf{n}_p$ and the surface tangent plane normal $\mathbf{n}_s$. Thus, $\mathbf{C}^t$ is calculated by,

$$\mathbf{C}^t = \frac{\mathbf{n}_s \times \mathbf{n}_p}{|\mathbf{n}_s \times \mathbf{n}_p|} \tag{1}$$

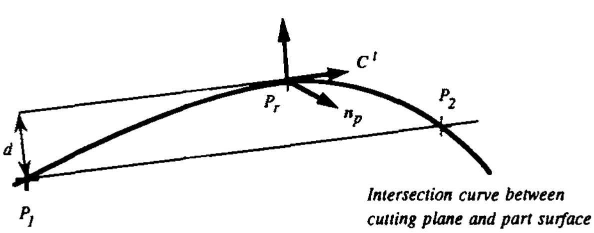

Fig. 5. Initial chordal approximation method for locating a chord with specified tolerance.

the nominal chordal deviation d at $\mathbf{P}_r$ is obtained by,

$$d = \overrightarrow{P_1P_r} \cdot \frac{\mathbf{n}_p \times \mathbf{C}^t}{|\mathbf{n}_p \times \mathbf{C}^t|} \tag{2}$$

where $\mathbf{n}_p \times \mathbf{C}^t/|\mathbf{n}_p \times \mathbf{C}^t|$ forms the unit vector which is perpendicular to $\mathbf{C}^t$ and also on the cutting plane.

The procedure begins from the current CC point, $\mathbf{P}_1$. The nominal chordal deviation of a neighbouring point, $\mathbf{P}_r$, on the cutting curve can be calculated by the formulations described above. If the nominal deviation d at $\mathbf{P}_r$ is substantially different from the specified tolerance, T, i.e. $|d - T| > \epsilon$, where ϵ is a small magnitude, a binary search scheme locates another point on the intersection curve until the nominal chordal deviation matches the specified tolerance within $\pm\epsilon$.

The binary search calculates another neighbouring point by either adding Δw to or subtracting Δw from the current value of the independent variable w depending on the sign of $(d - T)$. The step size Δw is adjusted similarly by multiplying or dividing by a factor of two, depending on the sign of $(d - T)$. The procedure continues until the nominal deviation converges to the desired point where $|d - T| \leq \epsilon$. The endpoint of the chord, $\mathbf{P}_2$, which generates the desired nominal deviation d, is approximated by doubling the difference in the independent variable between $\mathbf{P}_1$ and $\mathbf{P}_r$, and the other parameter value of $\mathbf{P}_2$ is found by using the golden section method. Since the true machining error calculation requires only a close estimate of the desired cutter contact point, this estimate of $\mathbf{P}_2$ is generally sufficient. The point $\mathbf{P}_2$ is further adjusted by the true machining error calculation method to become the next CC point.

2.2 True Machining Error

Typically a pair of tool centre (TC) points define consecutive tool positions, and the tool moves linearly from one TC point to the other. The true machining error, E, is defined by $E = |R - d|$, where R is the tool radius and the tool centre distance, d, is defined as the minimum distance from the tool trajectory line to the part surface. To find d, the orthogonal projection method is used [8]. This technique is originally proposed as a method to generate blending surfaces. Here it is employed to project the tool trajectory line onto the part surface, as illustrated in Fig. 6. Since the orthogonal projection

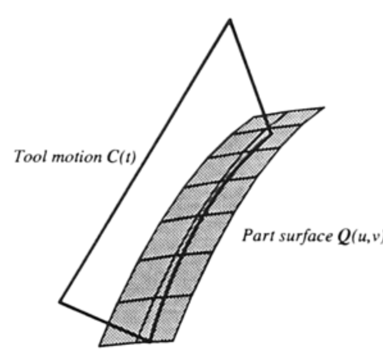

Fig. 6. Orthogonally projecting tool motion to part surface.

method casts a space point to a surface via the corresponding surface normal, the distance between the space point and the projection point on the surface is minimum.

Denoting a general space curve as $C(t)$ and the parametric surface derivatives at point $\mathbf{Q}$ as $\mathbf{Q}^u$ and $\mathbf{Q}^v$, the orthogonal projection method is formulated by,

$$\begin{bmatrix} \dfrac{du}{dt} \\ \dfrac{dv}{dt} \end{bmatrix} = K^{-1} \begin{bmatrix} \dfrac{d\mathbf{C}}{dt}\mathbf{Q}^u \\ \dfrac{d\mathbf{C}}{dt}\mathbf{Q}^v \end{bmatrix} \tag{3}$$

where

$$K = \begin{bmatrix} \mathbf{Q}^u\cdot\mathbf{Q}^u + \overrightarrow{PQ}\cdot\mathbf{Q}^{uu} & \mathbf{Q}^u\cdot\mathbf{Q}^v + \overrightarrow{PQ}\cdot\mathbf{Q}^{uv} \\ \mathbf{Q}^v\cdot\mathbf{Q}^u + \overrightarrow{PQ}\cdot\mathbf{Q}^{vu} & \mathbf{Q}^v\cdot\mathbf{Q}^v + \overrightarrow{PQ}\cdot\mathbf{Q}^{vv} \end{bmatrix} \tag{4}$$

This formulation characterises the relationship between a space curve and the corresponding projection curve on the surface. When the parameter of the space curve is incremented by dt, the corresponding projected point can be found by calculating the parametric variation (du, dv) from equation (3) evaluated at the previous projected point on the part surface. Since the goal is to find the minimum distance between the tool trajectory line and the part surface, the trajectory line is sampled at uniform intervals and thus the corresponding surface projection points are found as described above. The number of sampling points can be adjusted according to the complexity of the part surface and the machining accuracy required. It should be noted that smaller values of dt will yield more sampled projected points on the surface, so the distance between the tool trajectory line and the part surface will be more accurate. On the other hand, a smaller dt value will also require longer processing time.

The true machining error calculation method first calculates the distance, d, between the sample points on the tool trajectory line and the part surface, as shown in Fig. 7. Since the space curve in this case is the tool trajectory line on which the tool centre is located, subtracting the distance from the tool radius yields the tool machining errors at these sampled points. The maximum machining error among all sampled points is the true machining error of this tool motion. This true machining error is used to adjust the tool motion according to a binary search similar to the one used in the initial chordal approximation method. Since the calculation of true machining error is the most computationally intensive part of this tool path generation system, the close CC point provided by the initial chordal approximation improves the overall efficiency of the algorithm.

Since the true machining error is defined as an absolute value, the method is valid for both concave and convex regions of the surface. For a concave region, the maximum machining error occurs where the tool centre distance is a maximum. In such a case the tool undercuts (i.e. misses) the design surface. The converse is true for a convex region. Furthermore, if the tool trajectory line passes through the surface (e.g. in a sharp convex region) the tool centre distance, d, will be calculated with a negative value and E will be greater than R.

The integrated procedure for initial chordal approximation and true machining error calculation is referred to as CC point location and is illustrated in Fig. 8. This procedure generates consecutive CC points that meet the specified machining tolerance, and minimises the number of CC points. However, this procedure calculates only the error that is created by each linear tool motion. To address gouging problems, a direct gouge-elimination process is presented. This process works simultaneously with the CC point location procedure to ensure that the final tool path is gouge-free.

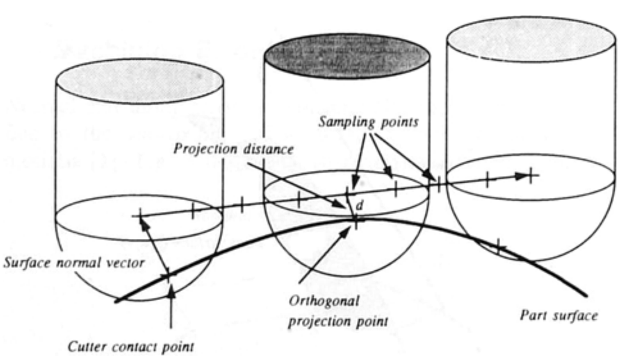

Fig. 7. Calculate the machining error of a tool motion by sampling.

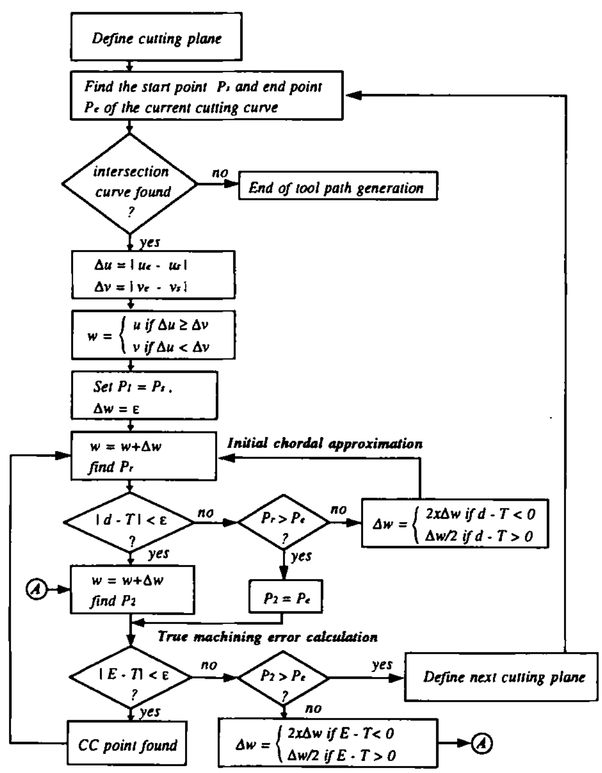

Fig. 8. Initial chordal approximation and true machining error calculation for CC point generation.

3. Direct Gouge-Elimination Process

Throughout the CC point location procedure, surface points are generated by the golden section search, these points are checked against a list of previous TC points to detect possible gouging. This procedure is depicted in Fig. 9, where the intermediate surface points found by the golden section are indicated by the symbol ○, the current target CC point by □, and previous TC points by ×. If the distance between a TC point and any suface point (intermediate or current target) is less than the tool radius, gouging exists. (In actual implementation the square of the distance is calculated and compared to save computation time.) When gouging is detected, the TC point record is modified to avoid the gouge, thus all gouging tool motions can be modified or purged.

One of the advantages of the direct gouge-elimination method is that it detects and eliminates gouging tool motions simultaneously with the CC point location procedure. Furthermore, this method checks each tool motion against surface points that lie either along or beside the tool pass. This is more accurate than checking tool motions only with the CC points of a tool pass for gouging [5], yet it is more efficient than comparing tool positions with all CC points of the entire tool path [10]. Fig. 10 demonstrates the surface points that are checked for gouge while generating a single tool pass. It should be noted that more surface points are checked in the area that is closer to the tool pass, and a higher density of points is checked for gouge at areas with larger curvature.

Fig. 11 depicts a typical gouge that occurs where the radius of surface curvature is smaller than the tool radius. In this figure, $\mathbf{CC}_0$, $\mathbf{CC}_1$ and $\mathbf{CC}_2$ are cutter contact points that have been generated from the CC point location procedure. The points $\mathbf{TC}_0$, $\mathbf{TC}_1$ and $\mathbf{TC}_2$ are the TC points offset from the corresponding CC points by the tool radius along the surface normal vectors. When the surface point **P** is tested for gouging, the tool motion from $\mathbf{TC}_1$ to $\mathbf{TC}_2$ is found gouging the part surface, i.e. the distance between $\mathbf{TC}_1$ and **P** is greater than the tool radius, R, and the distance between $\mathbf{TC}_2$ and **P** is less than R. To avoid gouging, a new cutter location point $\mathbf{TC}_2'$ is generated and inserted as a substitute for $\mathbf{TC}_2$. The point $\mathbf{TC}_2'$ is calculated so that the distance between it and **P** is equal to the tool radius. The substitute point is defined by the equation,

$$|\mathbf{TC}(t) - \mathbf{P}| = R \tag{5}$$

and is solved for t. In this equation $\mathbf{TC}(t)$ is the parametric line segment representing the tool trajectory, i.e. $\mathbf{TC}(t) = \mathbf{TC}_1(1-t) + \mathbf{TC}_2 t$. After modification, the tool motion will stop at $\mathbf{TC}_2'$ without gouging the part surface at point **P**.

To detect gouging in deep concave regions, each generated surface point is checked not only against the last successful TC point, but also against a list of several of the most recent TC points, referred to as the active TC list. Consider, for example, the situation shown in Fig. 12. Proceeding from left to right, as the CC point location and direct gouge elimination procedures work together, tool centre points $\mathbf{TC}_0$ through $\mathbf{TC}_3$ are generated without gouging. However, in searching for $\mathbf{CC}_4$, surface points are found which indicate that $\mathbf{TC}_3$ would gouge the surface near $\mathbf{CC}_4$. $\mathbf{TC}_3$ is thus modified by the method described above, and the CC point location procedure continues from $\mathbf{CC}_4$ without adding a TC point corresponding to it. Similarly, in searching for $\mathbf{CC}_5$, surface points are found which indicate that both $\mathbf{TC}_2$ and the modified $\mathbf{TC}_3$ gouge the surface. Thus, $\mathbf{TC}_3$ is deleted and $\mathbf{TC}_2$ is modified. As new TC points are generated or deleted, the active TC list is updated in a last-in–first-out manner.

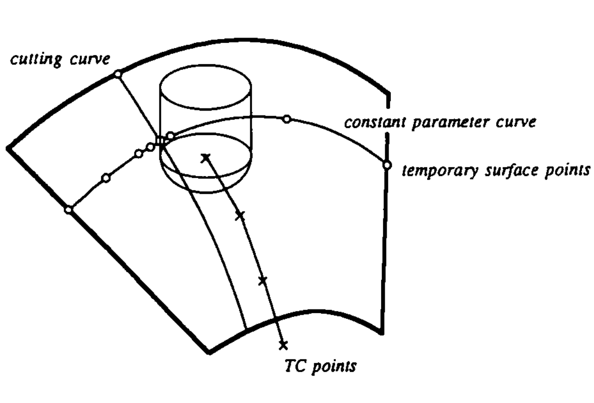

Fig. 9. Surface points generated by the golden section method are checked with previous TC points for gouging during CC point location procedure.

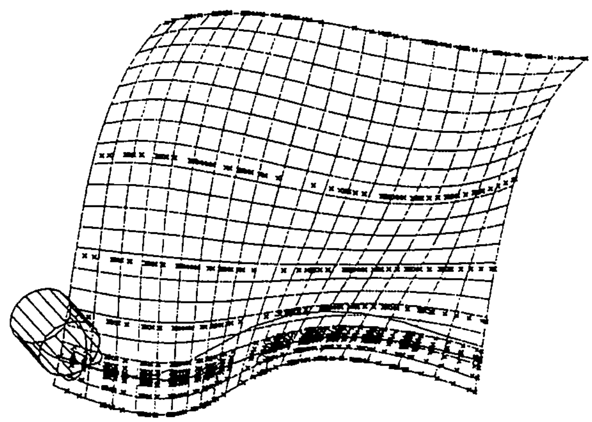

Fig. 10. Surface points checked for gouge while generating a single tool pass.

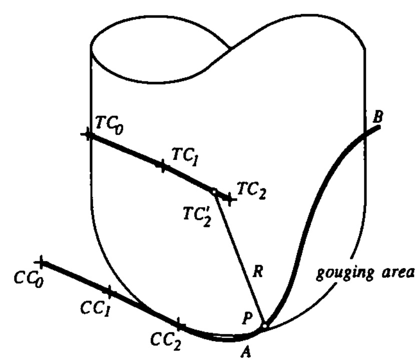

Fig. 11. Adjusting TC points to eliminate gouge.

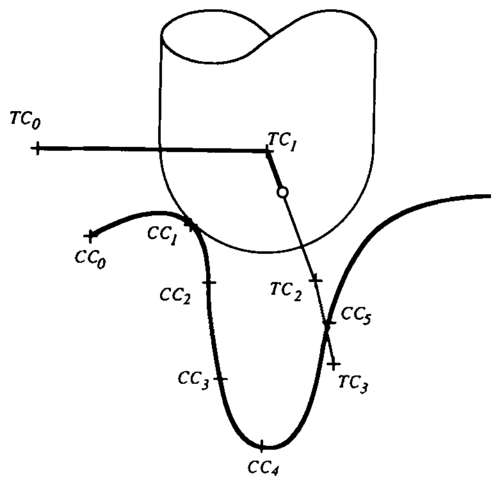

Fig. 12. An example of the direct gouge elimination procedure.

The procedure continues to modify or delete all gouging tool locations in the tool path.

The size of the active TC list, i.e. the number of previous TC points against which each generated surface point must be checked, affects both the accuracy and efficiency of the direct gouge-elimination process. It is adjusted according to the complexity of the part surface and the desired machining accuracy.

4. Tool Pass Interval Adjustment

The distance between two adjacent tool passes is referred to as the tool pass interval, as depicted in Fig. 13. The cusp is the remaining material between two adjacent tool passes, and it affects the smoothness of the machined part surface. If the part surface is flat, the tool pass interval is constant for all tool passes. However, for sculptured surfaces, the tool pass interval is generally different from one pass to another, depending on the tool radius and the local surface curvature [11]. Tool pass intervals must be calculated from a specified cusp height tolerance to ensure surface smoothness.

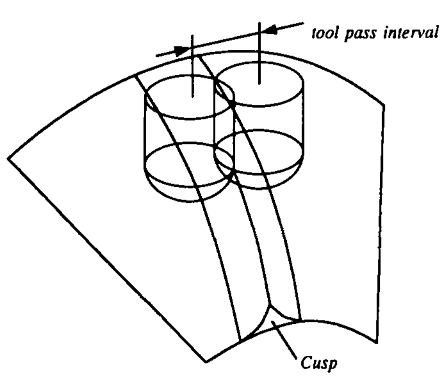

Fig. 13. Tool pass interval and cusp.

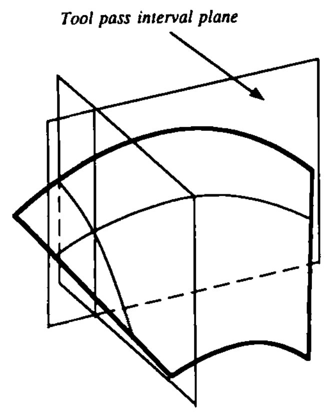

Fig. 14. Tool pass interval plane.

The approach developed here is an adaptation of tool pass interval adjustment methods developed for constant parameter tool path generation techniques. This non-constant parametric tool pass interval adjustment method uses the radius of curvature of the intersection curve of the part surface and a plane perpendicular to the cutting plane, referred to as the tool pass interval plane, as shown in Fig. 14. The tool pass interval, l, on a non-flat surface, as depicted in Fig. 15, can be calculated by approximating the actual surface curve between A and B as a circular arc [10,11], and observing that,

$$(\rho+R)^2 + (\rho+h)^2 - R^2 = 2(\rho+R)(\rho+h)\cos\phi \tag{6}$$

where ρ is the radius of curvature and,

$$\cos\phi = \sqrt{\left[1 - \left(\frac{l}{2\rho}\right)^2\right]} \tag{7}$$

Solving for l yields,

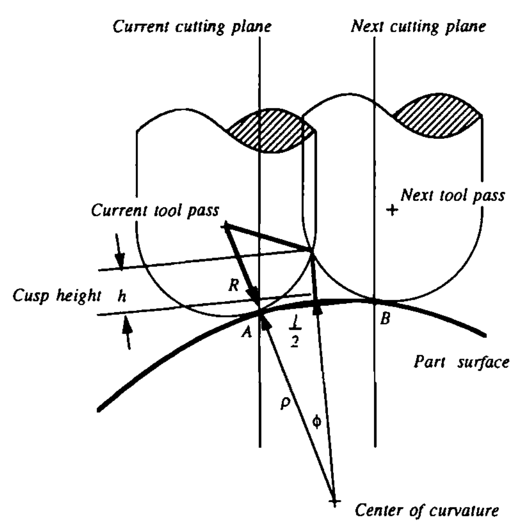

Fig. 15. Non-constant tool pass interval calculation.

$$l = \frac{\rho\sqrt{\{4(\rho+R)^2(\rho+h)^2 - [\rho^2+2Rh+(\rho+h)^2]^2\}}}{(\rho+R)(\rho+h)} \tag{8}$$

When the radius of curvature of the part surface is flat the tool pass interval l becomes,

$$l = 2\sqrt{(2Rh - h^2)} \tag{9}$$

For non-constant parameter tool generation, the line on which the tool pass interval is measured is not generally parallel to the normal vector of cutting planes. So, to define the next cutting plane, the tool pass interval l is projected onto the cutting plane normal, as shown in Fig. 16.

Note that since a circular arc approximation is used to find the tool pass interval l, ΔOCD forms an isosceles triangle. Thus, $\angle DOC$ can be calculated by the property of isosceles triangles,

$$\angle DOC = \cos^{-1}\left(\frac{\rho l}{2}\right) \tag{10}$$

Denoting the angle between the cutting plane normal and the surface normal at the cutter contact point O as $\angle AOB$, the projection magnitude of the tool pass interval can be calculated by,

$$l' = l\cos(\angle AOB + \angle DOC = \pi) \tag{11}$$

where l' is the projection magnitude of l. On a flat surface, l lies exactly on the flat surface and $\angle DOC$ is equal to $\pi/2$ then,

$$l' = l\cos\left(\angle AOB - \frac{\pi}{2}\right) \tag{12}$$

The tool pass interval is calculated at all CC points of a tool pass by using the non-constant parameter tool pass interval adjustment method combined with the magnitude projection formulation. The resulting tool pass intervals at all CC points are compared and the smallest one is applied to define the next cutting plane.

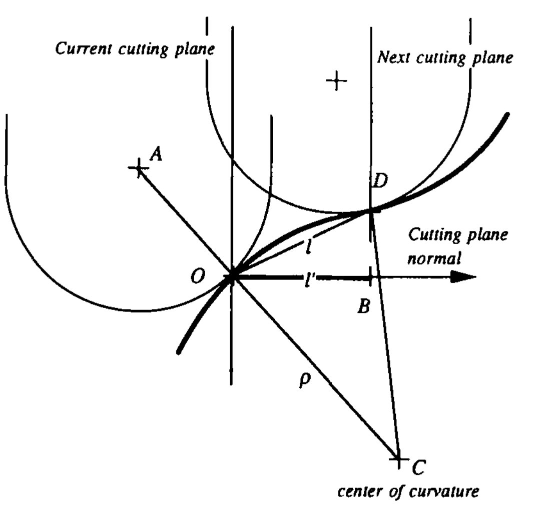

Fig. 16. Projecting tool pass interval onto the cutting plane normal.

5. Implementation and Results

Fig. 17 illustrates an overview of the software implementation of the non-constant parameter NC tool path generation system. The algorithm has been coded in the C language using the graphics library on a Silicon Graphics workstation. Two application examples are presented to demonstrate the capabilities of the algorithm. First, a simple wave-shaped surface is used to illustrate the gouge-elimination capability. The second example involves the tool path generation on a fan-shaped surface.

Example 1. Gouge Detection and Elimination

This example features NC tool path generation on a surface which contains a concave area where the radius of curvature is smaller than the tool radius. A bi-cubic Bézier surface defined by the 4 × 4 control point matrix:

$$\begin{matrix} (1,1,20) & (10,1,-50) & (25,1,10) & (40,1,-30) \\ (1,7,20) & (10,7,-50) & (25,7,10) & (40,7,-30) \\ (1,25,20) & (10,25,-50) & (25,25,10) & (40,25,-30) \\ (1,40,20) & (10,40,-50) & (25,40,10) & (40,40,-30) \end{matrix}$$

is to be machined with the following machining parameters:

Ball-end tool radius: 15
Machining tolerance: 0.01
Cusp tolerance: 15
Cutting plane normal vector: (0,1,0)

The computation time for this example including all graphic output, was 8 seconds on an SGI personal Iris. In this example, a very large cusp height was assigned to the tool path generation program (cusp height must be less than or equal to the tool radius). Only two tool passes were generated for this surface, since the intent of this example is to demonstrate the gouge elimination capabilities. Results are shown in Figs. 18 and 19. In Fig. 18 the direct gouge-elimination module has been disabled, and a serious gouge is

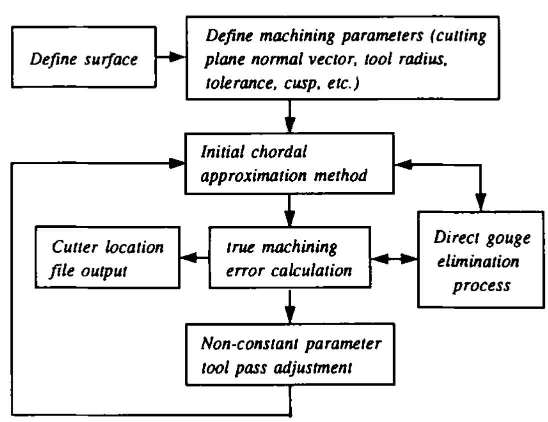

Fig. 17. Overview of the non-constant parameter tool path generation algorithm.

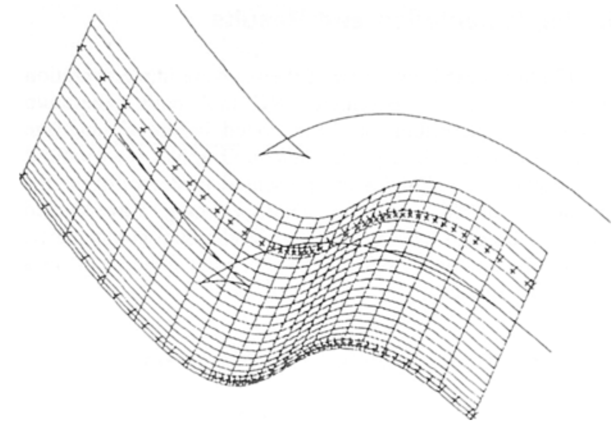

Fig. 18. Tool path generated without gouge elimination.

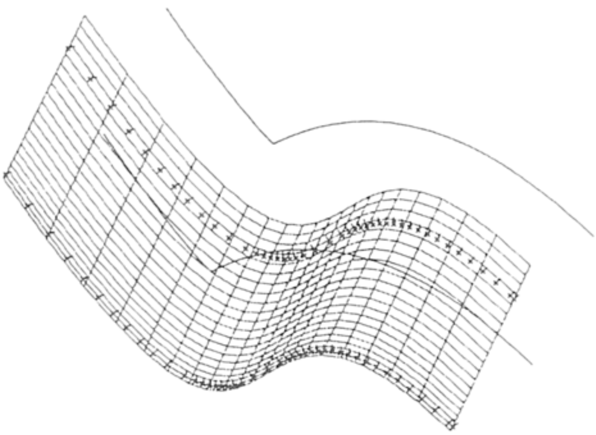

Fig. 19. Tool path generated with direct gouge elimination.

evident at the concave region. Fig. 19 shows the results of including the gouge elimination portion of the algorithm. This example also shows that long tool motions occur at smooth areas and short tool motions exist at highly curved areas.

Example 2. Fan-shaped Surface

This example demonstrates tool path generation on a fan-shaped Bézier surface which is defined by the control point matrix,

$$\begin{matrix} (10,1,0) & (15,3,8) & (20,5,10) & (25,2,0) \\ (5,8,8) & (11,10,16) & (23,12,18) & (35,10,10) \\ (1,23,10) & (15,25,18) & (25,29,20) & (45,25,8) \\ (-5,32,0) & (19,38,6) & (42,40,4) & (55,34,-2) \end{matrix}$$

The machining parameters are specified as,

Ball-end tool radius: 0.1875

Maximum machining tolerance: 0.001

Maximum cusp height: 0.004

Cutting plane normal vector: (1, 0, 0)

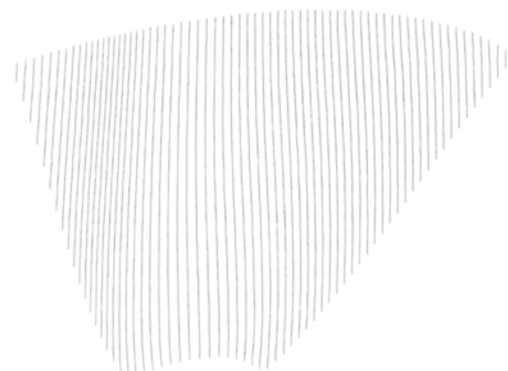

Fig. 20. Tool path generated by non-constant parameter method.

The computation time for this example is 34 seconds on an SGI Personal Iris computer.

The CL-data generated on this surface, as plotted in Fig. 20, was post-processed into the format of a Dyna Mechtronics DM4400 milling machine, and the part was milled from a block of wax. Uniform smoothness of the part surface can be observed from the tool path plot (Fig. 20) and the photograph of the actual milled product (Fig. 21). It should be noted that density of tool passes is greater on the left portion of the part where a relatively steep surface slope exists. This phenomenon illustrates the subtle effect of the tool pass interval adjustment method.

The same surface and machining parameters were also entered into a commercially available CAM software package for tool path generation via a typical constant parameter curve method. The tool path, as plotted in Fig. 22, was also post-processed and the milled result from this yielded the part shown in Fig. 23.

The results summarised in Table 1 indicate that both methods generate the same number of tool passes for the given machining parameter setting, but the non-constant parameter approach yields 476 fewer CL points than the constant parameter method. As expected, on the milled part produced by the constant parameter method, the cusp heights vary from one end to the other; the roughest portion meets

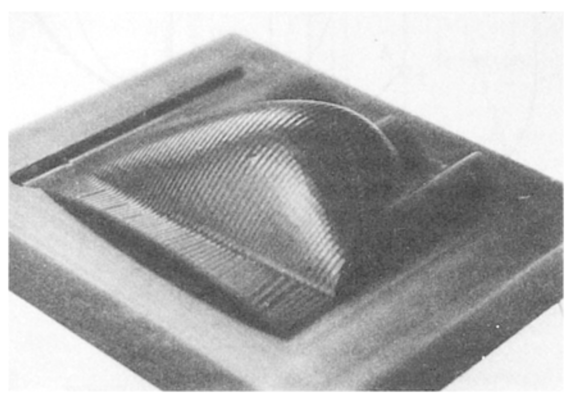

Fig. 21. Part milled by non-constant parameter tool path.

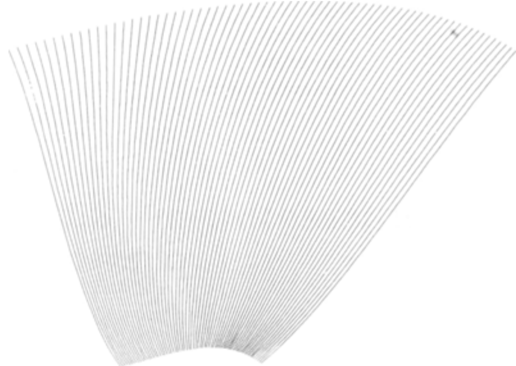

Fig. 22. Tool path generated by constant parameter curve method.

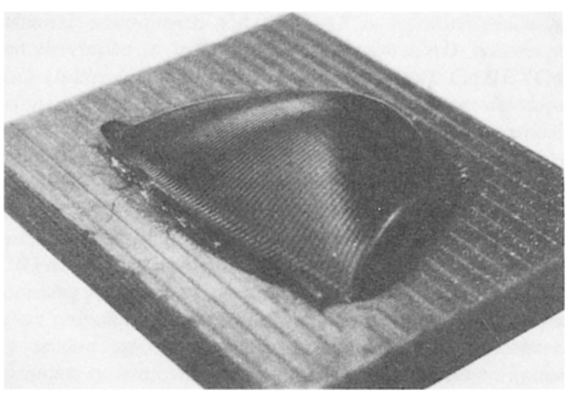

Fig. 23. Part milled by constant parameter curve tool path.

Table 1. Comparison of results from non-constant parameter and constant parameter tool path generation methods.

	Number of tool passes	Number of CL points
Non-constant parameter tool path generation	68	1496
Constant parameter tool path generation	68	1972

the specified cusp height tolerance while the finest portion is unncessarily smooth. The part produced by the non-constant parameter approach has a uniform surface smoothness that meets the specified cusp height tolerance without sacrificing machining efficiency.

6. Conclusion

The non-constant parameter approach for tool path generation is independent of the parametric coordinate system, and therefore provides a high degree of flexibility in planning tool path direction. Thus this approach is very well suited to most sculptured surface machining applications. The implementation of this algorithm demonstrates its efficiency and flexibility when compared to the constant parameter curve method of a commercially available CAM system. However, this algorithm requires more computational effort in locating cutting curves than the constant parameter curve method does. Work is continuing on improving the computational efficiency of this technique and incorporating capabilities for multiple surfaces.

Acknowledgement

The authors are grateful for the financial support of this work provided by the Iowa Center for Emerging Manufacturing Technology. Thanks are also due to Mr Patrick Bergan for his technical advice in post-processing CL-data, producing wax models, and constructive suggestions for paper preparation.

References

1. C.-H. Chang and M. A. Melkanoff, *NC Machine Programming and Software Design*, Prentice–Hall, New Jersey, 1989.
2. M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*, Prentice–Hall, New Jersey, 1987.
3. G. C. Loney and T. M. Ozsoy, "NC machining of free form surfaces", *Computer-Aided Design*, **19**(2), pp. 85–90, March 1987.
4. B. K. Choi, C. S. Lee, J. S. Hwang and C. S. Jun, "Compound surface modelling and machining", *Computer-Aided Design*, **20**(3), pp. 127–136, April 1989.
5. D. A. Wysocki, J. H. Oliver and E. D. Goodman, "Gouge detection algorithms for sculptured surface NC generation", *ASME Computer-Aided Design and Manufacture of Cutting and Forming Tools*, PED **40**, pp. 39–44, 1989.
6. M. E. Mortenson, *Geometric Modeling*, John Wiley & Sons Inc., New York, 1987.
7. J. E. Bobrow, "NC machine tool path generation from CSG part representations", *Computer-Aided Design*, **17**(2), pp. 69–76, 1985.
8. J. Pegna and F.-E. Wolter, "Designing and mapping trimming curves on surfaces using orthogonal projection", *ASME Advances in Design Automation, Vol. 1, Computer Aided and Computational Design*, DE-Vol. 23-1, pp. 235–245, September 1990.
9. R. L. Fox, *Optimization Methods for Engineering Design*, Addison–Wesley Publishing Company, New York, 1971.
10. B. K. Choi and C. S. Jun, "Ball-end cutter interference avoidance in NC machining of sculptured surfaces", *Computer-Aided Design*, **21**(6), pp. 371–378, July/August, 1989.
11. Y. S. Suh and K. Lee, "NC milling tool path generation for arbitrary pockets defined by sculptured surfaces", *Computer-Aided Design*, **22**(5), pp. 371–378, June 1990.

Notation

$\mathbf{C}$	cutting curve
$\mathbf{C}^t$	cutting curve tangent
$\mathbf{CC}_0, \mathbf{CC}_1, \ldots$	cutter contact points
d	chordal deviation
$\angle ABC$	triangle
Δw	incremental step in parameter w
$\langle ABC$	angle
ϵ	a small quantity
l	chord length
$\mathbf{n}_s, \mathbf{n}_p, \ldots$	normal vectors
$\mathbf{P}, \mathbf{P}_r, \mathbf{P}_e, \mathbf{P}_1, \mathbf{P}_2, \ldots$	space point

$\mathbf{Q}$	parametric equation of a surface	ϕ	angle
R	radius of a ball-end milling tool	ρ	curvature
$\mathbf{TC}_0$, $\mathbf{TC}_1$, ...	tool center points	h	cusp height
u, v, u_s, u_e, w, t	parameters	T	machining tolerance