The Third International Conference
on Automation Technology, July 1994

Real-Time Interpolator for 5-Axis Surface Machining

Rong-Shine Lin
and
Yoram Koren

Department of Mechanical Engineering
and Applied Mechanics
The University of Michigan
Ann Arbor, Michigan 48109-2125
U. S. A.

Abstract

In this paper, a new real-time interpolator for surface machining on 5-axis machine tools is proposed. This interpolator produces smoother surfaces and requires less machining time compared with the conventional off-line approach. The proposed interpolator can also handle production of convex surfaces with flat-end cutters. The interpolator calculates in real time a new command in the same time period needed for sampling the control-loop feedback devices. It performs three steps in each sampling period: (1) tool-path planning based on a constant scallop height, (2) trajectory and orientation planning based on a constant feedrate, and (3) inverse kinematics transformation based on the structure of the machine. To use this real-time interpolator for surface machining, the 3-D parametric surface g-codes must be defined. The interpolator was implemented on a 5-axis milling machine and the results are illustrated by examples.

Keywords: CNC, Interpolator, 5-Axis Machining, Surface Machining.

1. Introduction

It is known that 5-Axis machining for molds and dies produces higher metal removal rates and improved surface finish, thereby, eliminating the secondary cleanup of scallops created by 3-axis machining [Sprow 1993, Jensen and Anderson 1992, Vickers and Quan 1989]. However, the conventional methods for 5-axis machining utilize off-line part programming approaches by which the CAD system divides the surface into a set of line segments that approximates the surface at the desired tolerance [Wang 1986, Sambandan 1989, Chou and Yang 1992, Jensen and Anderson 1992, and Renker 1993]. These line segments are further processed by post processors to produce straight-line g-codes which constitute the commands needed to control the machine. In the CNC, these g-codes are fed into the interpolator. Figure 1 shows the above processes.

These off-line approaches for 5-axis machining either assumes a constant tool orientation along each segment, or assumes a linear change in the tool orientation between each successive end-points. The constant orientation algorithm causes severe roughness around the end-points along the surface since the orientation changes are abrupt at these points. The linear orientation algorithm produces a better surface, but still interpolates the orientations inaccurately between end points (since the change of the orientation is not necessarily linear), which causes surface errors. An additional drawback of the off-line methods is that the cutter accelerates and decelerates at each segment, which increases the surface non-uniformity and substantially increases the cutting time [Koren et al., 1993].

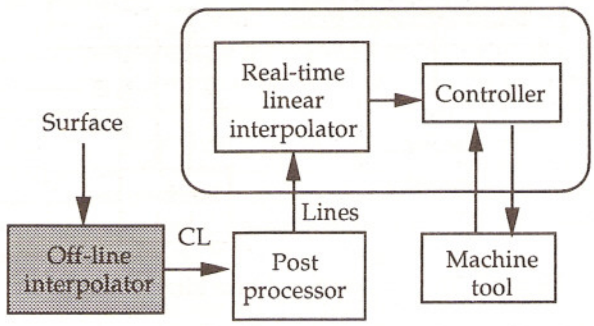


Figure 1: The off-line approach for surface machining

To overcome these drawbacks, we propose an algorithm for the precise real-time 5-axis interpolator for both tool orientation and positioning. This interpolator can handle production of convex surfaces with flat-end cutters.

The interpolator calculates a new command in the same time period needed for sampling the control-loop feedback devices. The interpolator performs three steps in each sampling period: (1) tool-path planning based on a constant scallop height, (2) trajectory planning based on a constant feedrate, and (3) inverse kinematics transformation based on the structure of the machine.

The proposed real-time 5-axis interpolator for surface machining is shown in Fig. 2. The input to the interpolator is a new defined g-code (i.e., an NC instruction), which contains the geometric information of the part surface as well as the cutting conditions, such as the feedrate, the spindle speed, the specific tool, etc. The interpolator begins with tool path planning, which generates curves as tool paths, and subsequently enters the trajectory planning portion for curve interpolation. The trajectory planning portion generates the position (x,y,z) and the orientation (O_x,O_y,O_z) of the cutter based on the surface geometry and the specified constant feedrate in each sampling period. This is a generic, machine independent algorithm. Based on the calculated cutter's position and orientation, the reference values of the five axes can be obtained by the inverse kinematics transformation, an algorithm which depends upon the structure of each particular machine. The detailed operation for each of these algorithms is described in the following sections.

To use this new interpolator for surface machining, new g-codes must be defined for programming the part program. In this paper, the 3-D parametric surface g-codes will be given and illustrated by examples.

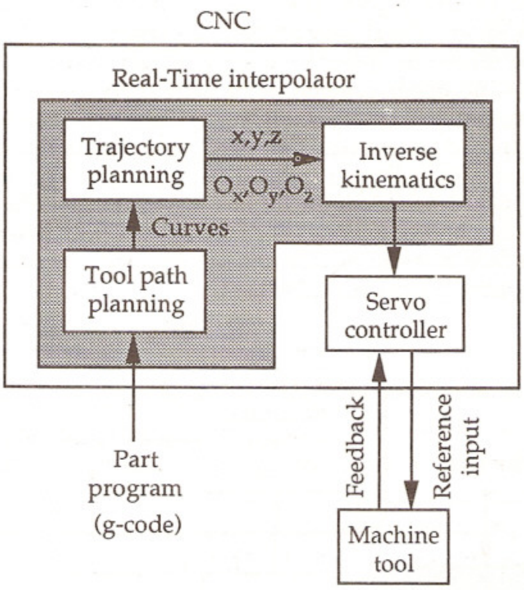


Figure 2: The proposed real-time interpolator

2. Tool Path Planning

The key idea of 5-axis machining is using flat-end cutters to produce part surfaces by controlling the cutter's orientation normal to the surface during cutting. For machining planar surfaces, this machining method produces perfect surface without any remaining scallops on the finished surface. With this method the tool path interval is equal to the diameter of the cutter (Fig. 3) and we assume that the machine is precisely aligned, and the cutters are not worn. The tool path interval is defined as the distance between the parallel tool paths.

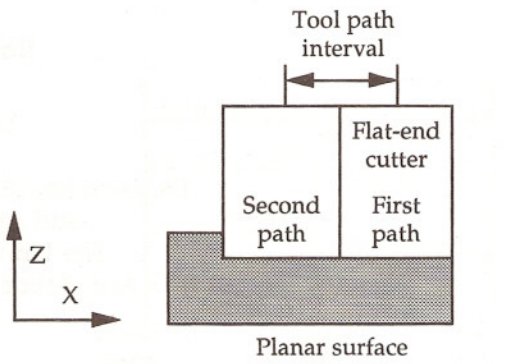


Figure 3: Planar surface machined by a flat-end cutter; the cutter paths are out of the page.

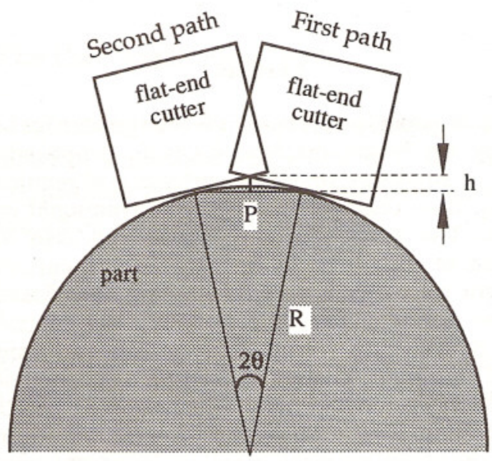


Figure 4: A convex surface machined by a flat-end cutter

For machining curved surfaces, scallops are created on the finished surface. Figure 4 shows the remaining scallops h and the tool path interval P of a convex surface machined by a flat-end cutter where the tool motion is into and out of the page. R is the local radius of curvature of the part surface. A tool path interval that is too large will result in a rough surface; one that is too small will increase the machining time, making the process thereby inefficient. By requiring that the scallop height remains at a given constant value, the tool path interval can be calculated. The calculation of tool path interval for machining a convex surface by a flat-end cutter is shown below. From Fig. 4, an algebraic equation can be obtained,

$$\frac{h}{R}=\frac{1}{\cos\theta}-1 \tag{1}$$

By substituting $\cos\theta=\sqrt{1-\left(\frac{P}{2R}\right)^2}$ into above equation and rearranging,

$$P = \frac{2R}{R+h}\sqrt{2Rh + h^2} \tag{2}$$

Equation (2) is the tool path interval that the cutter can slide without exceeding the allowable surface finish value. This equation is valid when $P \le$ cutter diameter.

To simplify the planning process, one of the boundary curve of a surface is chosen for the first tool path. Along this path, the tool path intervals are calculated by Eq. (2). Among these tool path intervals, the minimum one is selected for finding the next tool path by offsetting the previous path with this minimum interval. Therefore, the determined tool paths can produce the part surface within the allowable scallops.

3. Trajectory Planning

The trajectory planning algorithm generates the locations and orientations for the cutting tools based on the part geometry and feedrate, which are obtained from the part program. The part surface described here is represented by a parametric form S (u,v). The n^{th} order parametric surface is shown below,

$$S(u,v) = \begin{cases} x(u,v) = \sum_{j=0}^{n}\sum_{i=0}^{n} a_{ij}u^i v^j \\ y(u,v) = \sum_{j=0}^{n}\sum_{i=0}^{n} b_{ij}u^i v^j \quad 0 \le u,v \le 1 \\ z(u,v) = \sum_{j=0}^{n}\sum_{i=0}^{n} c_{ij}u^i v^j \end{cases} \tag{3}$$

where a_{ij}, b_{ij}, and c_{ij} are the coefficients of the cubic polynomials.

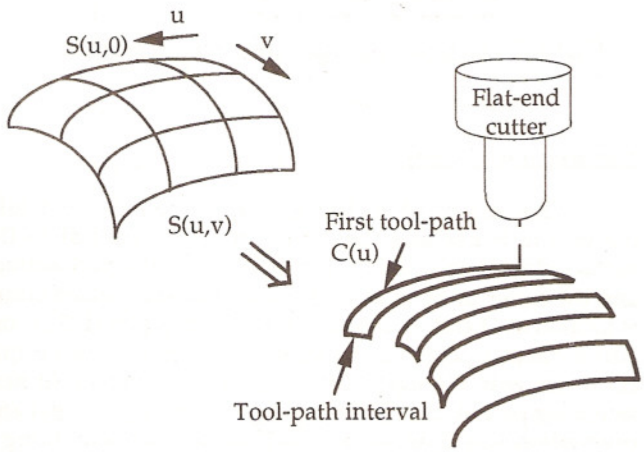


Figure 5: The parametric surface and its tool paths

Position Calculation

To produce smooth part surfaces, the machining feedrate (V) must be constant when the cutter tracks the cutting trajectory. To keep the constant feedrate, the cutting tool has to move a constant distance relative to the workpiece in each sampling period of the interpolator. One of the boundary curve C (u) of above parametric surface is chosen to be the first tool path, C(u) = S (u,0), (Fig. 5).

$$C(u) = x(u)\hat{i} + y(u)\hat{j} + z(u)\hat{k} \tag{4}$$

$$\text{where} \begin{cases} x(u) = a_n u^n + a_{n-1}u^{n-1} + \cdots + a_1 u + a_0 \\ y(u) = b_n u^n + b_{n-1}u^{n-1} + \cdots + b_1 u + b_0 \\ z(u) = c_n u^n + c_{n-1}u^{n-1} + \cdots + c_1 u + c_0 \end{cases}$$

Based on this parametric curve, the interpolator calculates the tool position at each sampling period (T). However, the constant change in the parameter Δu in the parametric domain does not guarantee the constant length in the Cartesian domain, and consequently does not guarantee a constant feedrate. To obtain the constant feedrate, the conversion between the parameter u and the constant distance that the tool moves at each sampling period has to be derived, A solution based on Taylor's expansion [Koren et al., 1993] is used to obtain the value of Δu that corresponds to equal trajectory length of the parametric curve. Equation (5) shows the function of u in terms of the length VT that the tool moves during one sampling period T. If the machining feedrate V is constant, VT is constant as well.

$$\Delta u = u_k - u_{k-1} = \frac{VT}{\sqrt{x_{k-1}'^2 + y_{k-1}'^2 + z_{k-1}'^2}} \tag{5}$$

$$\text{where } x' = \frac{dx}{du},\ y' = \frac{dy}{du},\ \text{and } z' = \frac{dz}{du}$$

The subscript k and k-1 mean the current sampling time and previous sampling time respectively. The cutter location at the k^{th} sampling period can be obtained by substituting u_k, from Eq. (5), into Eq. (4)

The precision of this approach has been analyzed in [Koren et al., 1993]. It has been shown there that the reference positions calculated by this real-time interpolator algorithm results in very small contour errors compared with conventional off-line interpolators. The variations in the velocity commands by the real-time approach are negligible for most applications (e.g., less than 0.1%). By contrast, the off-line approach causes very large velocity errors. In addition, the machining by the conventional off-line interpolator, the machine has to follow a sequence of short straight lines, which causes acceleration and deceleration at the beginning and the ending of each line. This results in a substantial increase in the machining time.

In contrast, the real-time approach reads the g-code of the parametric surface directly and generates the reference positions for the tool in the same rate needed by the control loops. This eliminates the acceleration and deceleration effect and consequently results in substantial reduction in the machining time.

Orientation Calculation

The key idea of 5-axis machining is using flat-and cutters to produce part surfaces by controlling the cutter's orientation normal to the surface during cutting. In other words, the cutter not only follows the determined tool paths, but also orientates the cutter axis in the direction of surface normal. The surface normal direction $[O_x,O_y,O_z]$can be calculated by follows [Faux and Pratt, 1981],

$$[O_x,O_y,O_z]_k = \frac{\left.\frac{\partial S}{\partial v}\times\frac{\partial S}{\partial u}\right|_{,u=u_k}}{\left.\left|\frac{\partial S}{\partial v}\times\frac{\partial S}{\partial u}\right|\right|_{,u=u_k}} \qquad 0 \le v \le 1 \qquad (6)$$

where $[O_x,O_y,O_z]_k$ is the tool orientation at the specific tool position $u = u_k$.

However, the conventional off-line approaches vary cutter orientations linearly between the intermediate points of the curve approximated by linear segments. With this off-line approach, the interpolated orientation n between the intermediate points is

$$n = n_{k-1} + (n_k - n_{k-1}) \bullet \frac{(u - u_{k-1})}{(u_k - u_{k-1})} \qquad (7)$$

where $u_{k-1} \le u \le u_k$, n_k and n_{k-1} are the orientations of the two end-points of the approximated segment and correspond to the parameters u_k and u_{k-1}, respectively. This approach interpolates inaccurate orientations since the changes in the tool orientations are not necessarily linear. The orientation errors subsequently affect the position accuracy (discussed further in the section below on inverse kinematics transformation). As an example, the orientation errors caused by the off-line interpolation of the parametric curve

$$\begin{cases} x(u) = 10u^3 + 10u^2 + 10 \\ y(u) = 10u^2 + 10u \end{cases} \qquad 0 \le u \le 1 \qquad (8)$$

were simulated and shown in Fig. 6. The curve in this example was approximated by 52 segments with maximum contour error of 1 μm. The maximum orientation error is 0.017 degrees that will subsequently cause a position error to 0.045 mm (see section below for details).

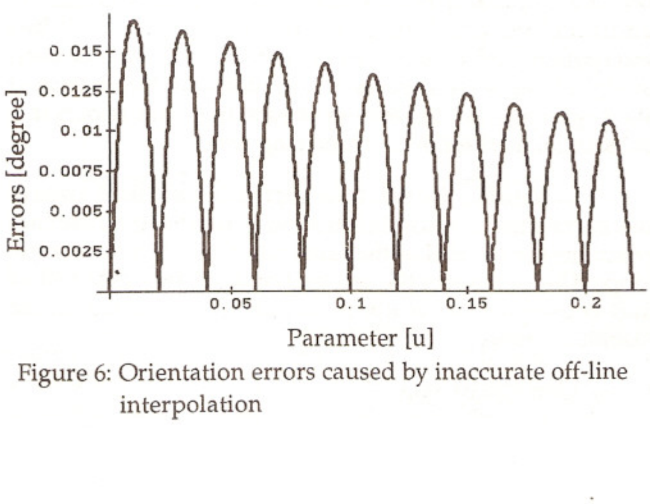


Figure 6: Orientation errors caused by inaccurate off-line interpolation

4. Inverse Kinematics Transform

The six variables $[x,y,z,O_x,O_y,O_z]$ of Eqs. (4) and (6) are the solutions of the cutter's position and orientation at each sampling time. The derivation so far is based on the part surface geometry and a specific machining feedrate; therefore, these solutions are machine independent. Subsequently, these six variables have to be transformed by inverse kinematics into five reference inputs for the controllers of the five-axis machine.

The inverse kinematics transformation depends upon the structure of the machine. Since the cutting tool is symmetric, we need only five degrees of freedom to reach any point with orientations in space. The 5-axis machine structure shown in Fig. 7 is used to demonstrate the derivation of the solution for the inverse kinematics, in this paper, where a tilting and a rotary tables are installed upon the x-axis table. However, the derivation procedure can be applied to the other 5-axis configurations.

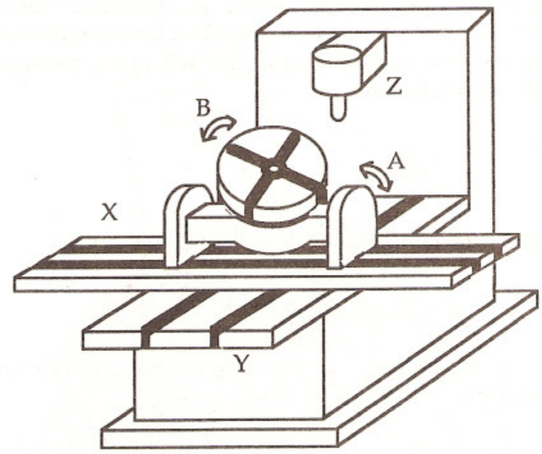


Figure 7: The demonstrated 5-axis milling machine

The Rotation Motions

The main idea behind the five-axis end-mill machining is keeping the cutting tool axis normal to the surface during machining. Figure 8 shows the part surface and the cutter position at the k^{th} sampling time where cutter is located at O and its axis is in the direction [0,0,1]. P_0 is the cutter contact point on the surface and its surface normal coincides with the fixed cutter axis at this moment. In next sampling period, the cutter's next position P_1 (p_x,p_y,p_z) and orientation $[O_x,O_y,O_z]$ are calculated while maintaining a constant feedrate by Eqs. (4) and (6). To maintain the cutter normal to the surface during machining, the surface normal $[O_x,O_y,O_z]$ has to be rotated to the direction of [0,0,1] during the next sampling, as shown in Fig. 9 for a 2-D case. For the general 3-D case, two rotations are required. For the structure shown in Fig. 7, the first rotation is of axis A at angle a about the X-axis and the second rotation is of angle b about the Z-axis. The equations of motions can be obtained as follows,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix} \bullet \begin{bmatrix} \cos(b) & -\sin(b) & 0 \\ \sin(b) & \cos(b) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (9)$$

where a is the rotating angle and b is the tilting angle. Equation (9) represents three scalar equations. The angle a and b are obtained by solving the first two equations as follows

$$\begin{cases} a = \tan^{-1}\left(\dfrac{O_x}{O_y}\right) & 0 \le a \le 2\pi \\ b = \tan^{-1}\left(\dfrac{\sqrt{O_x{}^2 + O_y{}^2}}{O_z}\right) & 0 \le b \le \dfrac{\pi}{2} \end{cases} \quad (10)$$

By substituting Eq. (10) into the third scalar equation of Eq. (9), we obtain $O_x^2 + O_y^2 + O_z^2 = 1$ which is always true because O_x, O_y, and O_z are the directional cosines of an unit vector in Cartesian coordinates.

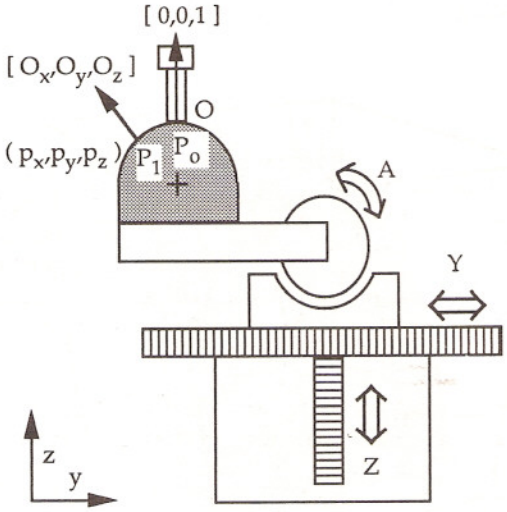


Figure 8: Illustration of machine motion

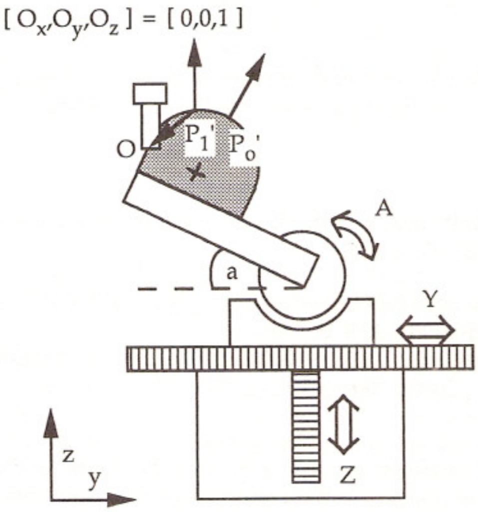


Figure 9: Rotating the vector [O_x, O_y, O_z] to [0,0,1].

The Translation Motions

By the rotations demonstrated in Fig. 9 the correct tool orientation was achieved, but not the correct position because the rotation moved the part away from the cutter where P_1 has been moved to P_1'. Therefore, to complete the machining for the next sampling period, P_1' must be translated back to O and denoted as P_1''. Figure 10 shows the completed cutting motion that the cutter locates at P_1'' and the cutter axis matches with surface normal. The distance $\overline{OP_1'}$ is called the pull-back distance. For the 3-D case based on the table configuration of Fig. 7, the pull-back distance can be calculated by the following equation

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix} \bullet \left\{ \begin{bmatrix} \cos(b) & -\sin(b) & 0 \\ \sin(b) & \cos(b) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} P_x + R_x \\ P_y + R_y \\ P_z + R_z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \right\} + \begin{bmatrix} OT_x \\ OT_y \\ OT_z \end{bmatrix} \quad (11)$$

where the detailed coordinate systems is shown in Fig. 11. P, R, T, and OT are the coordinate systems for the part surface, the rotary table, the tilt table, and the cutter center, respectively. The five variables [X,Y,Z,a,b], shown in Eqs. (11) and (10), provide the five reference inputs for the five servo-controllers that control the machine.

From Eq. (11), we see that the tool position accuracy depends on the precision of interpolated values a and b. In the previous example shown in Fig. 6, the maximum orientation error of 0.017 degrees will cause the position error of 45 μm. This position error is obtained by substituting the physical values (a=0.017 degree; T_y=96 mm; T_z=120 mm; R_x=8 mm; and b = P_y = P_z = R_y = R_z = T_x = OT_x = OT_y = OT_z =0) of a five-axis machine into Eq. (11).

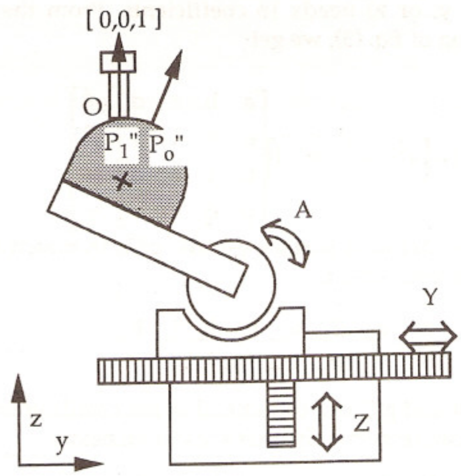
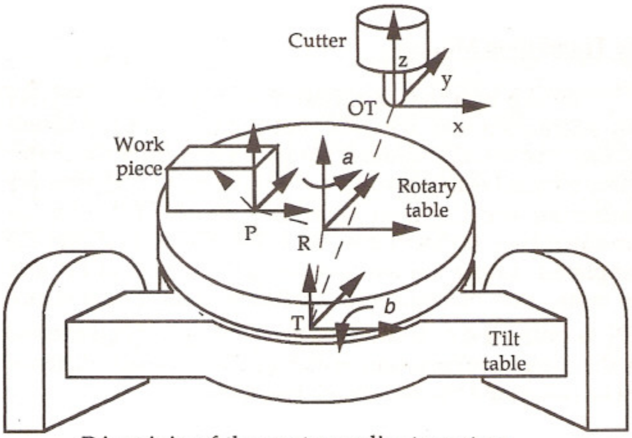


Figure 10: Completed cutting motion: from P_0 to P_1.



P is origin of the part coordinate system
R is the center of the rotary table
T is the center of the Tilt Table
OT is a fixed coordinate system at the tool center

Figure 11: The coordinate systems for Eq. (11).

5. G-codes for Parametric Surfaces

In order to use the real-time surface interpolator, new g-code formats for CNC machine tools have to be defined. The g-code contains the surface geometry information and machining conditions. The parametric cubic form is adopted for the surface representation in this g-code. Although high-order polynomials can describe complex surfaces, they require a large number of coefficients whose physical significance is difficult to grasp. In addition, it has been found that cubics are a good compromise in most applications, and most design methods are implemented using cubic parametrisation [Faux and Pratt, 1981].

A surface S(u,v) with parametric cubic form is shown in Eq. (3) by substituting n = 3. Forty-eight coefficients are needed to design a parametric cubic surface where each direction (x, y, or z) needs 16 coefficients. From the first scalar equation of Eq. (3), we get

$$x(u,v) = \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} \begin{bmatrix} a & b & c & d \\ e & h & i & j \\ k & l & n & o \\ p & q & r & s \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \tag{12}$$

Therefore, we define a surface g-code, g53, to match these coefficients in x-direction,

g53 a__ b__ c__ d__ e__ h__ i__ j__ k__
l__ n__ o__ p__ q__ r__ s__ (13)

Likewise, g54 and g55 can be defined as the coefficients in y- and z-direction, respectively, that are shown below

g54 a__ b__ c__ d__ e__ h__ i__ j__ k__
l__ n__ o__ p__ q__ r__ s__ (14)

g55 a__ b__ c__ d__ e__ h__ i__ j__ k__
l__ n__ o__ p__ q__ r__ s__ (15)

The new g-codes, g53 to g54, represent a parametric cubic surface, which contains the geometric information needed for the surface interpolator. However, a parametric surface, which is defined by two parameters (u and v), can be machined by either direction, u or v. Therefore, two variables (u and v) are added in g53: if u = 1, machining begins from u direction; if v = 1, machining begins from v direction.

6. Examples

The surface g-codes defined a developable surface are shown below where BLU = 0.01 mm.

g53 o 3000 v1
g54 r3000
g55 j-4000 o1000 s3000

Figure 12 shows the defined developable surface, the machining tool paths, and the orientations of the cutter axis where only parts of the cutter axis are shown. The cutter radius is 5 mm and the scallop height remained on the surface is 0.

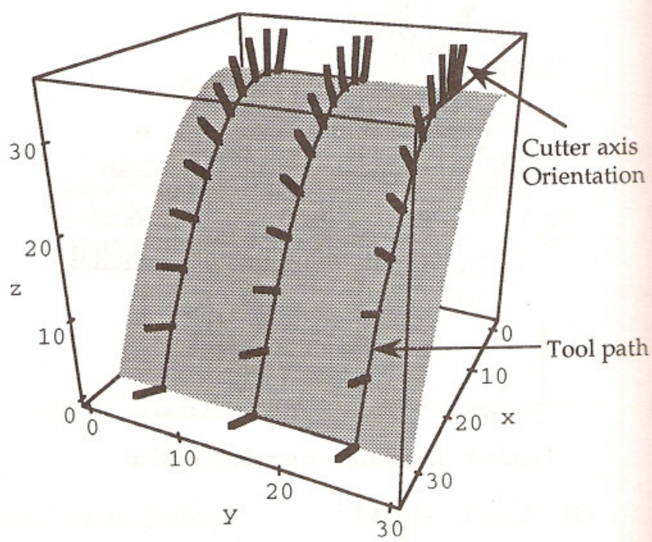


Figure 12: The g-code defined developable surface and the tool paths with tool axis orientations.

Another example of surface g-codes define a convex surface (BLU = 0.01 mm),

g53 b-2000 c2000 h3000 i-3000 p-9000 q13000 r1000 v1
g54 d-9000 j13000 o 1000
g55 a-6000 b10000 c-4000 d4000 e9000 h-15000 i6000
j-7000 o1000 p-4000 q5000 s2000

The convex surface and its tool paths machined by a cutter of 10 mm radius are shown in Fig. 13. In order to see tool paths clearly, the maximum scallop height is chosen for 1 mm.

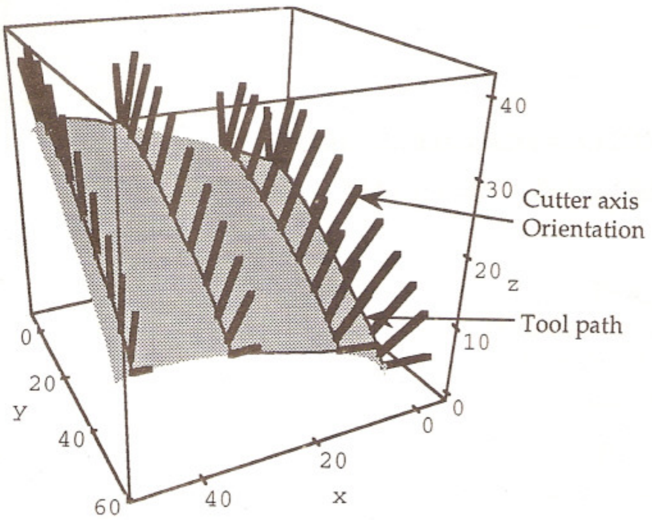


Figure 13: The g-code defined convex surface and the tool paths with tool axis orientations.

7. Discussions and Conclusions

In this paper, the algorithm for a new real-time 5-axis surface interpolator is proposed. The interpolator is limited to convex surface machining. For concave surface machining, the tool gouging problems are also involved, which requires additional calculations, and therefore the method cannot be implemented in real-time.

We have shown that the proposed real-time 5-axis surface interpolator has interpolated more accuracy the tool orientations and tool positions than conventional off-line interpolators. The real-time interpolator calculates the precise tool orientation and position at each control sampling based on the specified feedrate; therefore, it eliminates the acceleration and deceleration steps caused by off-line interpolator during machining. Subsequently, it produces smoother surface and requires substantially less machining time.

References

Chou, J. and Yang, D., 1992 "On the Generation of Coordinated Motion of Five-Axis CNC/CMM Machines", *Journal of Engineering for Industry*, Vol. 114, Feb., pp. 15-22.

Faux, I. and Pratt, M., 1981, Computational Geometry for Design and Manufacturing, New York, John Wiley & Sons.

Jensen, C. G. and Anderson, D. C., 1992, "Accurate Tool Placement and Orientation for Finish Surface Machining", *ASME Winter Annual Meeting*, pp. 127-145.

Koren, Y., Lo, C.C., and Shpitalni, M., 1993, "CNC Interpolators: Algorithm and Analysis", PED-Vol. 64, *Manufacturing Science and Engineering, ASME winter annual meeting*, Nov. pp. 83-92.

Renker, H., 1993, "Collision-Free Five-Axis Milling of Twisted Ruled Surfaces", Annals of the *CIRP* Vol. 42/1/1993, pp. 457-461.

Sanbandan, K. and Wang, K., 1989, "Five-Axis Swept Volumes for Graphic NC Simulation and Verification", *ASME the 15th Design Automation Conference*, DE Vol. 19-1, pp. 143-150.

Sprow, Eugene E., 1993, "Set Up to Five-Axis Programming", *Manufacturing Engineering*, November pp. 55-60.

Vickers, G. W. and Quan, K. W., 1989, "Ball-Mills Versus End-Mills for Curved Surface Machining", *ASME Journal of Engineering for Industry*, Vol. 111 Feb. pp. 22-26.

Wang, W. and Wang, K., 1986, "Real-Time Verification of Multiaxis NC Programs with Raster Graphics", Proceedings of *IEEE International Conference on Robotics and Automation*, pp. 166-171.