Journal of Manufacturing Processes
Vol. 2/No. 1
2000

Ruled Surface Machining on Five-Axis CNC Machine Tools

Rong-Shine Lin, Dept. of Mechanical Engineering, National Chung Cheng University, Chia-Yi, Taiwan
Yoram Koren, Dept. of Mechanical Engineering and Applied Mechanics, The University of Michigan, Ann Arbor, Michigan

Abstract

To achieve high precision and high productivity in machining sculptured surfaces, a new architecture for a five-axis CNC interpolator for machining ruled surfaces was developed and demonstrated on a milling machine. The objective of the five-axis interpolator is to continuously maintain the milling cutter axis in parallel with the straight lines of the ruled surface. The cutter position and orientation are calculated at each sampling period of the interpolator, and corresponding axial position commands are generated by an inverse kinematics algorithm. This real-time approach produces precise surfaces and requires substantially less machining time compared to the conventional off-line approach. Two new g-codes are also given in this paper for the new interpolator to produce part surfaces in CNC milling machines.

Keywords: *Five-Axis Machining, Five-Axis CNC, Ruled Surface, Computer Control*

Introduction

Parts containing ruled surfaces, such as conic and cylindrical surfaces, can be machined either by ball-end cutters on three-axis milling machines or by flank milling on five-axis milling machines. Using flank milling enables a tremendous increase in productivity as well as improvement of the surface finish; however, the conventional methods for five-axis machining utilize the off-line approach by which the CAD system divides the curve into a set of line segments that approximates the curve at the desired tolerance.[1-7] These line segments are further processed by post processors to produce the g-code that contains the computer numeric control (CNC) commands needed to control the machine. The g-code is fed into the CNC real-time interpolator whose major function is to generate coordinated reference inputs for the servocontrollers that control the machine tool axes.

Conventional off-line approaches for five-axis machines typically assume a constant tool orientation along each segment. The constant orientation algorithm causes severe roughness around the end-points along the surface because the orientation changes are abrupt at these points. More advanced approaches assume a linear change in the tool orientation between each successive end-point. The linear orientation algorithm produces a better surface but still interpolates the orientations inaccurately between end-points (because the change of orientation is not necessarily linear), which causes surface errors. An additional drawback of off-line methods is that the cutter accelerates and decelerates at each segment, which increases surface nonuniformity and substantially increases cutting time.[8,9] Chou and Yang[10] provided a mathematical formulation for a Euler-angle-type five-axis machine to track a parametric curve. Their work interpolated the curve correctly, but the required tool orientations were assumed to be given from CAD files that did not fit real-time implementations for CNC systems.

To overcome these drawbacks, this paper develops an algorithm for a real-time five-axis interpolator that continuously controls tool orientation and position. This interpolator can perform accurate production of ruled surfaces with flank cutters. The interpolator, which is contained in the CNC computer, calculates new commands in the same time period used for sampling the control loop feedback (typically 0.5 ms to 5 ms). In each sampling period, the interpolator performs two steps: (1) trajectory planning based on a constant feed rate and (2) inverse kinematics transformation based on the structure of the machine.

The developed real-time interpolator for machining ruled surfaces is shown in *Figure 1*. The input to the interpolator is the NC g-code, which contains the geometric information of the part surface as well as the cutting conditions, such as feed rate, spindle speed, specific tool type, and so on. The trajectory planning portion generates the tool position (X, Y, Z) and orientation (O_x, O_y, O_z) of the cutter in each sampling period based on the surface geometry and

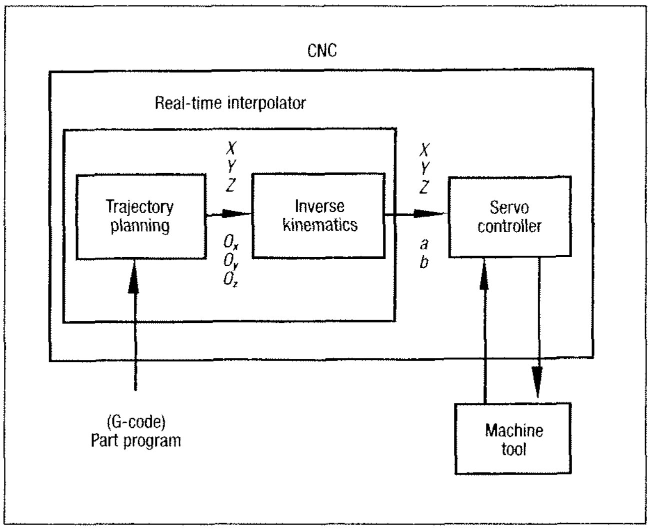

Figure 1
Proposed Real-Time Interpolator for Machining Ruled Surfaces

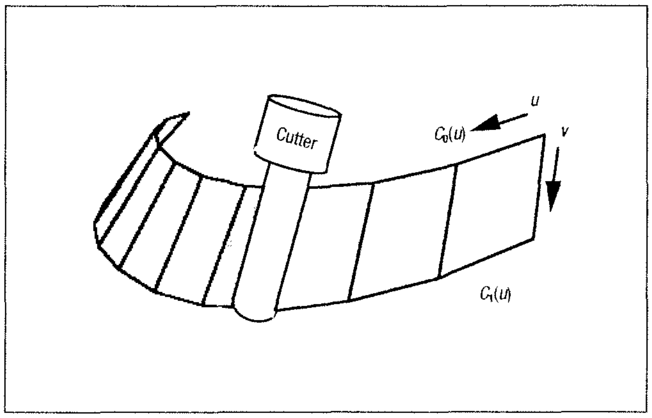

Figure 2
Example of the Ruled Surface and Its Flank Milling

specified feed rate. This is a generic, machine-independent algorithm. Based on the calculated cutter's position and orientation, the reference values (x, y, z, a, b) of the five axes are computed by the inverse kinematics transformation, an algorithm that depends on the structure of each particular machine. The detailed operation of these two algorithms is described in the following two sections. Examples are given in a later section.

Trajectory Planning

The trajectory planning algorithm generates the locations and orientations for the cutting tools based on part geometry and feed rate, which are obtained from the part program. The special geometric properties of ruled surfaces are described in this section. Based on these properties, algorithms are presented to calculate tool positions and tool orientations precisely along the ruled surface with machining at a constant feed rate.

Ruled Surfaces

A ruled surface is defined as a surface that through each point of it passes at least one straight line lying entirely on the surface.[11] In other words, a ruled surface is a surface generated by a family of straight lines (for example, a conic surface). This special geometric property can be used to increase the productivity of machining ruled surfaces on five-axis machines. The basic idea of this type of machining is moving the cutter's edge to follow the straight lines of the ruled surface. A ruled surface can be expressed by the following equation:

$$S(u, v) = (1 - v)\, C_0\,(u) + v\, C_1\,(u) \tag{1}$$

where C_0 and C_1 are two boundary curves of the ruled surface, which are known as directrices; u and v are parameters, normally $0 \leq u,\ v \leq 1$. An example of a ruled surface is shown in *Figure 2.*

Position Calculation

In this research, the real-time five-axis interpolator is proposed to be implemented on finish cutting, which removes only a small amount of material. Therefore, the tool motion can be assumed as constant feed rate machining. To simplify the implementation process, the lower boundary curve of the ruled surface is chosen for the tool contact trajectory, $C(u)$, as follows:

$$C(u) = x(u)\hat{i} + y(u)\hat{j} + z(u)\hat{k} \tag{2}$$

where

$$x(u) = a_n u^n + a_{n-1} u^{n-1} + \ldots + a_1 u + a_0$$

$$y(u) = b_n u^n + b_{n-1} u^{n-1} + \ldots + b_1 u + b_0$$

$$z(u) = c_n u^n + c_{n-1} u^{n-1} + \ldots + c_1 u + c_0$$

The interpolator calculates the tool position at each sampling period (ΔT) based on this parametric curve. 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 feed rate. To obtain the constant feed rate (V), 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 is used to obtain the value of Δu that corresponds to the equal trajectory length of the parametric curve. Eq. (3) below shows the function of u in terms of the length ($V \cdot \Delta T$) that the tool moves during one sampling period. If the machining feed rate V is constant, ($V \cdot \Delta T$) is constant as well.

$$\Delta u = u_k - u_{k-1} = \frac{V \cdot \Delta T}{\sqrt{x_{k-1}'^{\,2} + y_{k-1}'^{\,2} + z_{k-1}'^{\,2}}} \tag{3}$$

where

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

The subscripts k and k–1 mean the current sampling time and previous sampling time, respectively. The cutter contact location at the kth sampling period can be obtained by substituting u_k, from Eq. (3), into Eq. (2), as follows:

$$C(u_k) = \begin{cases} x(u_k) = a_n u_k^n + a_{n-1} u_k^{n-1} + \ldots + a_1 u_k + a_0 \\ y(u_k) = b_n u_k^n + b_{n-1} u_k^{n-1} + \ldots + b_1 u_k + b_0 \\ z(u_k) = c_n u_k^n + c_{n-1} u_k^{n-1} + \ldots + c_1 u_k + c_0 \end{cases} \tag{4}$$

The precision of this approach has been analyzed in Shpitalni, Koren, and Lo.[12] It has been shown that the reference positions calculated by this real-time interpolator algorithm result in very small contour errors compared with conventional off-line interpolators. The variations in the velocity commands (feed rates in g-codes) by the real-time approach are negligible for most applications. By contrast, the off-line approach causes very large velocity errors.

In addition, in 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 machining time. In contrast, the real-time approach reads the g-code of the parametric curve directly and generates the reference positions for the tool at the same rate needed by the control loops. This eliminates the acceleration and deceleration effect and consequently results in a substantial reduction in machining time.[8]

To obtain the cutter's location, the cutter contact location, Eq. (4), must be offset a distance of the cutter radius along the surface normal direction. Therefore, the tool center position at the kth sampling can be calculated as follows:[13,14]

$$OT_k = C(u_k) + r \cdot \frac{\frac{\partial S}{\partial v} \times \frac{\partial S}{\partial u}}{\left| \frac{\partial S}{\partial v} \times \frac{\partial S}{\partial u} \right|} \tag{5}$$

where r is the cutter radius and S is the ruled surface.

Orientation Calculation

The key idea of using five-axis machining for ruled surfaces is controlling the milling cutter axis in parallel with the straight lines of the ruled surface. In other words, the cutter not only follows the boundary curve of the ruled surface but also orients the cutter axis in the direction of the straight line where the cutter is located. Elaborated first is the off-line approach that varies 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 as follows:

$$n = n_{k-1} + (n_k - n_{k-1}) \bullet \frac{(u - u_{k-1})}{(u_k - u_{k-1})}, \qquad u_{k-1} \le u \le u_k \tag{6}$$

where n_k and n_{k-1} are the orientations of the two endpoints of the approximated segment and correspond to the parameters u_k and u_{k-1}, respectively. This approach interpolates inaccurate orientations because 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 a ruled surface in terms of the parametric form

$S(u, v) = (1 - v)\, C_0\, (u) + v\, C_1\, (u)$,
$0 \le u, v \le 1$

$$C_0(u) = \begin{cases} x(u) = 2.205u^3 - 7.425u^2 + 0.22u + 5 \\ y(u) = -2.205u^3 - 0.81u^2 + 8.051 \\ z(u) = 20 \end{cases}$$

(continued)

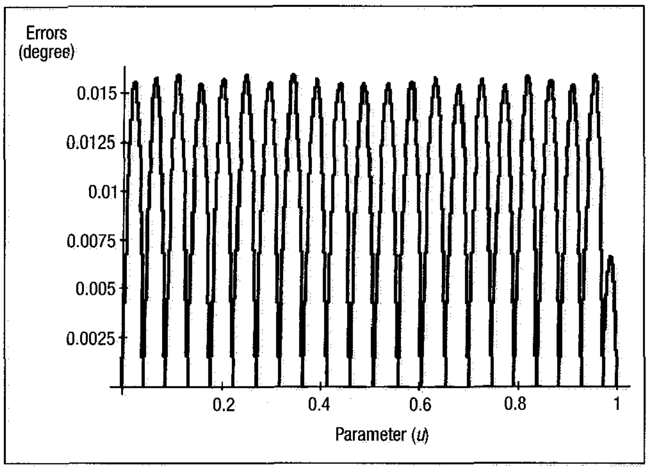

Figure 3
Orientation Errors Caused by Inaccurate Off-Line Interpolation

$$C_1(u) = \begin{cases} x(u) = 6.615u^3 - 22.275u^2 + 0.66u + 15 \\ y(u) = -6.615u^3 - 2.43u^2 + 24.045u \\ z(u) = 0 \end{cases} \quad (7)$$

were simulated and shown in *Figure 3*. The orientation error is defined as the surface normal error between the desired boundary and the interpolated result. The orientation error caused by the off-line approach is due to the linear interpolating results of Eq. (6). The lower boundary curve $C_1(u)$ in this example was approximated by 22 segments. The maximum contour error for this off-line approximation was 0.01 mm. The simulation result shows that the maximum orientation error is 0.016 degrees, which will subsequently cause a position error to 43 μm. This is not acceptable for high-precision machining systems (see the section on translation motions for details).

In contrast, the real-time interpolator calculates precisely the tool's orientation at each sampling period based on surface geometry. For each cutter's position, $u = u_k$, the cutting tool orientation can be calculated by the first derivative of Eq. (1) with respect to v, as shown below:

$$[O_x, O_y, O_z] = C_1(u_k) - C_0(u_k) \quad (8)$$

where $[O_x, O_y, O_z]$ is the tool orientation at the specific tool position $u = u_k$. As the same example shown in Eq. (7), the real-time interpolating result for the maximum orientation error caused by the real-time interpolation is 0.00018 degrees. This calculation was based on a CNC system with a feed rate of 20 mm/sec and a CNC sampling time of 0.01 sec. The orientation error brings a position error of 0.49 μm, which is much more accurate than that of the off-line approach. In the off-line approach, one possibility to increase the interpolating accuracy is increasing the number of approximated straight lines; however, this will cause very large feed rate variations, which results in a rough surface and makes machine vibration.[8]

The six variables $[x, y, z, O_x, O_y, O_z]$ of Eqs. (5) and (8) 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 feed rate; 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.

Inverse Kinematics Transformation

The conversion of the trajectory planning outputs into control references is performed by inverse kinematics in the real-time interpolator. The inverse kinematics transformation depends on the structure of the machine. Because the cutting tool is symmetric, only five degrees of freedom are needed to reach any point with orientations in space. In principle, there are a considerable number of configurations with respect to a five-axis arrangement; however, in practice most five-axis machines have a structure based on a three-axis Cartesian machine with the addition of two rotary axes. One of the simplest ways of constructing a five-axis machine is installing rotary and tilting tables on the top of the *x*-*y* table of the machine and having the *z*-axis move up and down. In this case, there are only six kinds of configurations, which are shown in *Figure 4*. In the structures shown in *Figures 4e* and *4f*, the rotary axis (*A*) is installed in the same direction as the *Z*-axis on the *X*-*Y* table and does not add a degree of freedom. With these structures, the tool does not have five degrees of freedom and five-face machining in one setup is impossible. Therefore, only the four configurations shown in *Figures 4a* through *4d* are commonly used in five-axis machines. Alternatively, the two rotation axes may be added on the spindle axis (*Z*-axis). Again, only four configurations out of the possible six are

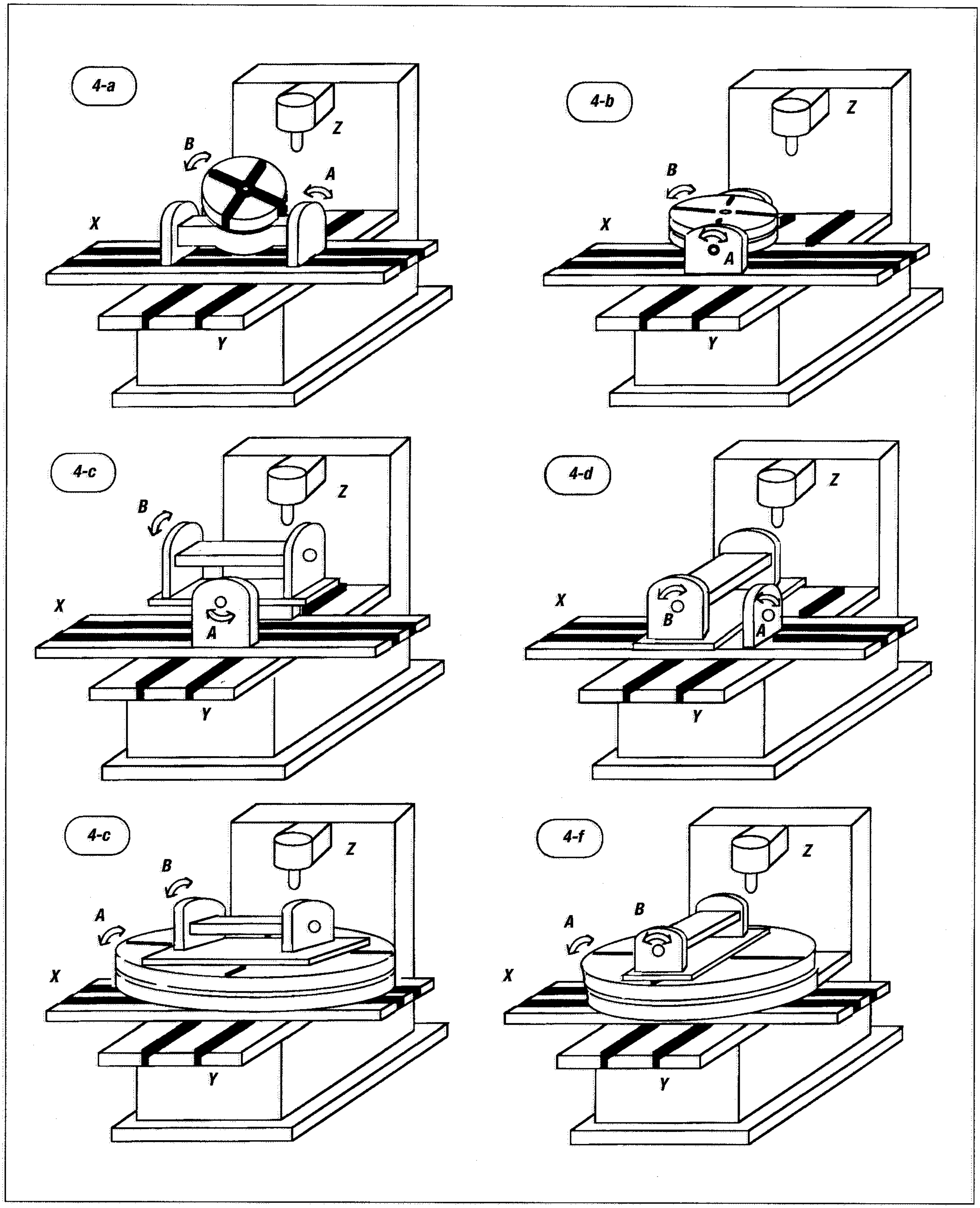

Figure 4
Six Kinds of Configurations for Five-Axis Machines.
(a) through (d) are configurations usually appearing in tilt tables; (e) and (f) cannot machine five faces in one setup.

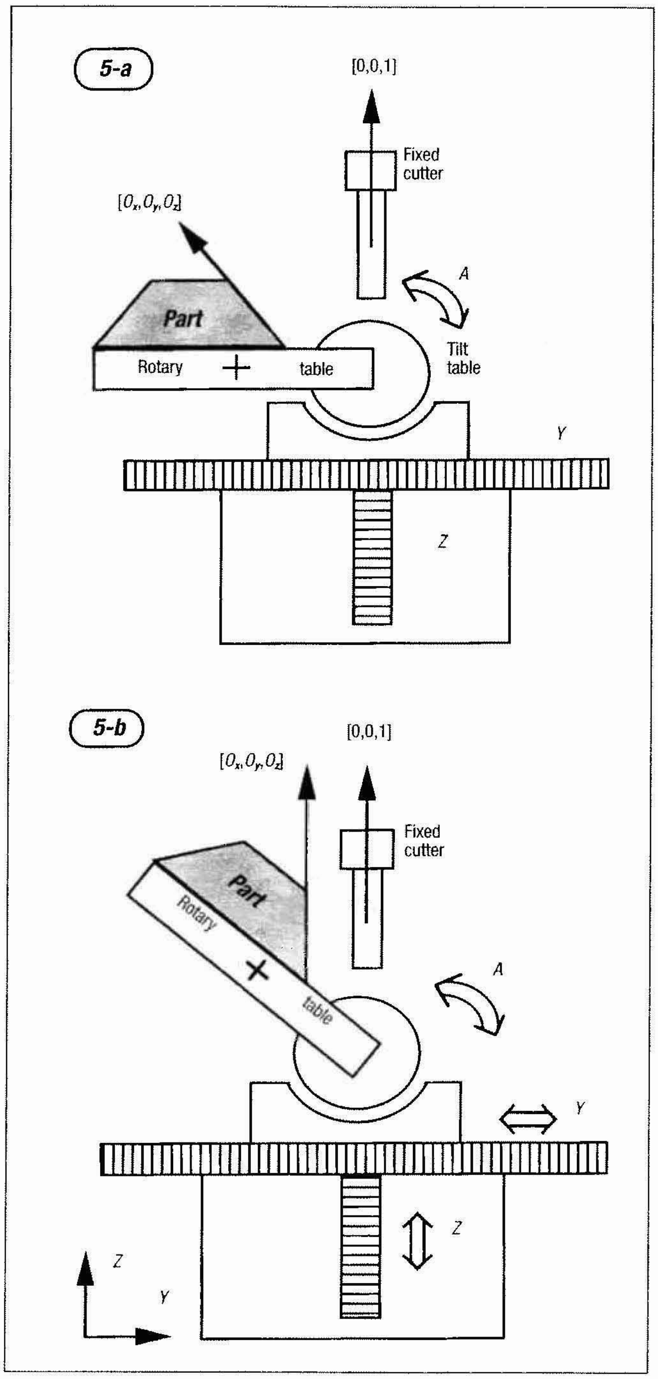

Figure 5
Table Motion for Controlling the Orientation.
(a) is before cutting; (b) is during cutting

acceptable. In this paper, only the four configurations in *Figures 4a* through *4d* are discussed; however, the derivation procedure can be applied to the case of adding rotating axes on the *Z*-axis.

The structure shown in *Figure 4a* is chosen to demonstrate the derivation of the solution for the inverse kinematics. Subsequently, a general solution for the inverse kinematics of the most common four configurations, *Figures 4a* through *4d*, is also presented in this paper.

Rotation Motions

The main idea behind five-axis machining for ruled surfaces is keeping the cutting tool axis in parallel with the straight lines of the ruled surfaces during machining. In each sampling period, the straight line's position $[x, y, z]$ and orientation $\lfloor O_x, O_y, O_z \rfloor$ are calculated while maintaining a constant feed rate, based on Eqs. (4) and (8). To obtain the maximum contact between the cutter and the surface during five-axis machining, the ruled surface straight line orientation $\lfloor O_x, O_y, O_z \rfloor$ has to be rotated to the direction of [0, 0, 1] that is the fixed cutter's direction, as shown in *Figure 5a* for a 2-D case. For the general 3-D case, two simultaneous rotations along the *A* and *B*-axis are required. For the structure shown in *Figure 4a*, 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 resultant motions at these rotations are shown in *Figure 5b*. The equations of motions can be obtained as follows:

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

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

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

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

Translation Motions

By the rotations demonstrated in *Figures 5a* and *5b*, the correct tool orientation was achieved, but not the correct position because the rotation moved the part out of the paper plane (that is, the X direction in *Figure 5*), if both a and b axes are rotated simultaneously. To machine the part surface on five-axis machines, both the tool orientation and position must be controlled. For easy illustration in the 2-D plane, an example of a cylindrical surface machining is selected to show the change in the tool position because of the motion of the rotary table. In *Figure 6a*, T_0 is a position on the cutter that contacts with the part surface at the current sampling time; T_c is the fixed cutter center position; and P_c is the part center position. T_0 is lying in the line $\overline{T_cP_c}$ at this moment. P_1 (*Figure 6a*) is the required tool contact point on the part surface at the next sampling time, which can be calculated by Eq. (4). The requirement for this particular machining is that the contact points between the tool and the part surface must be always lying in the line $\overline{T_cP_c}$. Therefore, to complete the machining for the next sampling period, P_1 must be rotated to P_1' (*Figure 6b*) and linearly translated back to T_0 by the distance $\overline{T_0P_1}$. The distance $\overline{T_0P_1}$ is called the pull-back distance. This translation makes $\overline{T_cP_1}$ and $\overline{P_cP_1}$ and collinear, which satisfies the requirement for simultaneous motions. For the 3-D case based on the table configuration of *Figure 4a*, the pull-back distance can be calculated by the following equation:

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

where the detailed coordinate system is shown in *Figure 7*. P, R, W, and OT are the coordinate systems for the part surface, rotary table, tilt table, and cutter center, respectively. The five variables $[X, Y, Z, a, b]$, shown in Eqs. (10) and (11), are the solutions for the inverse kinematics transform that provides the reference inputs for the servocontrollers to accurately control the five-axis machine with the configuration of *Figure 4a*. From Eq. (11), it is shown that the tool

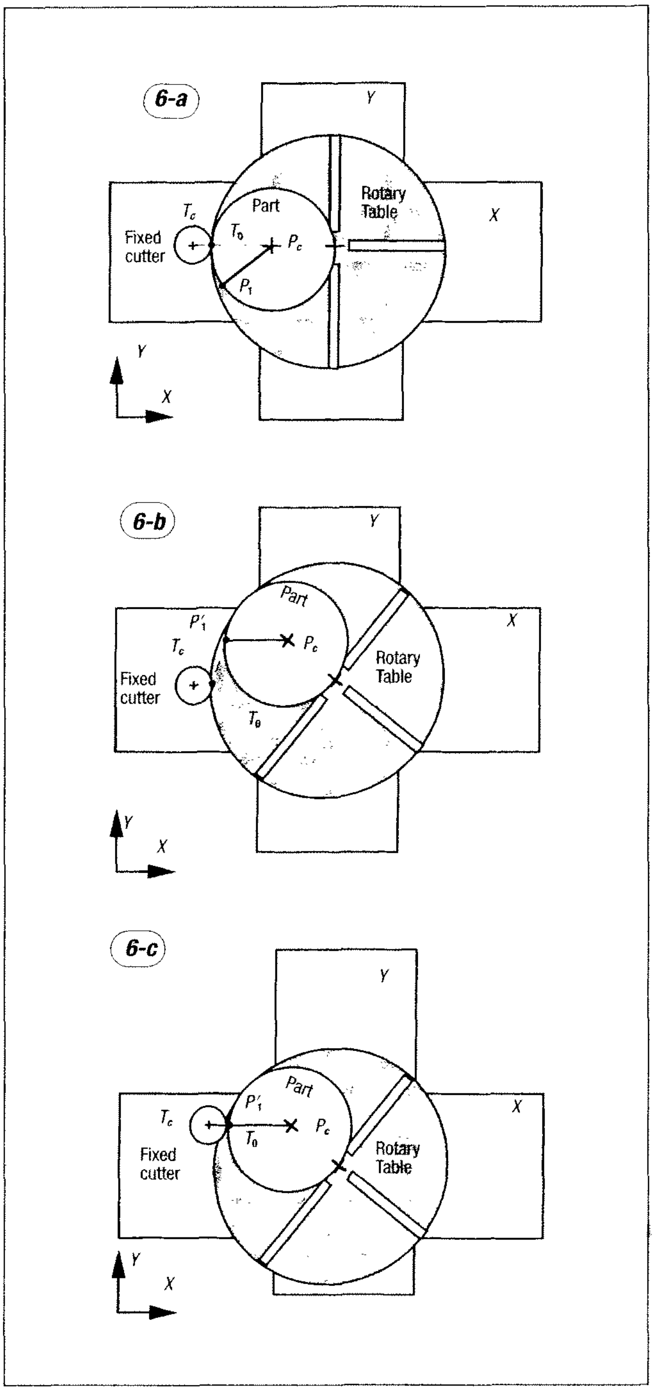

Figure 6
Illustration of Pull-Back Distance Due to Rotation Motion. (a) before cutting, (b) rotating table to desired orientation, (c) result after rotations and translations

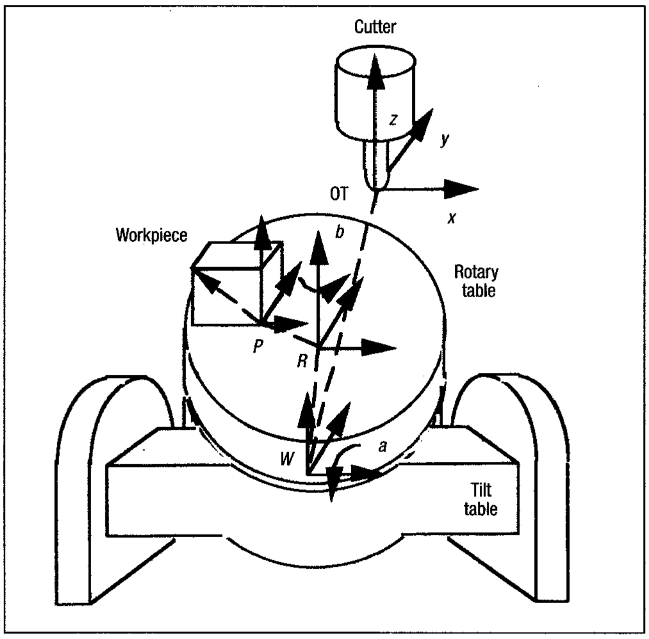

Figure 7
Coordinate Systems for Eq. (11)

Table 1
Virtual Rotation Angles for Each Case of *Figures 4a* Through *4d*

	α	α_x	α_y	α_z
Fig. 4a	0	0	0	0
Fig. 4b	0	0	0	-90
Fig. 4c	-90	0	-90	-90
Fig. 4d	90	90	0	0

position accuracy depends on the precision of interpolated tool orientation values a and b. In the previous simulation results of the off-line approach, as shown in *Figure 3*, the maximum orientation error of 0.016 degrees will cause the position error of 43 µm. This position error is obtained by substituting the physical values (a = 0.016 degrees; 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). In contrast, the real-time interpolation approach causes the maximum orientation error of 0.00018 degrees for the same example. Substantially, this brings only 0.49 µm position errors from Eq. (11).

For the other configurations, shown in *Figures 4b* through *4d*, the solutions can be obtained by using the same method. A general inverse kinematics solution for rotating tables of these configurations is given below [Eq. (13)]. This general solution is based on the configuration of *Figure 4a*. Therefore, to use this solution form, the configurations in *Figure 4b* through *4d* must be virtually rotated with respect to the certain rotation axis to become the same configuration of *Figure 4a*. *Table 1* gives the angle values for each case where α_x, α_y, and α_z are virtually rotating angles with respect to the X, Y, and Z axes, respectively; a is chosen for the same value of the virtually rotating angle whose axis is in the same direction of the A-axis.

The values for each case must be substituted into Eq. (12) to obtain the values of E, F, and G, as follows:

$$\begin{bmatrix} E \\ F \\ G \end{bmatrix} = \begin{bmatrix} \cos(\alpha_z) & -\sin(\alpha_z) & 0 \\ \sin(\alpha_z) & \cos(\alpha_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos(\alpha_y) & 0 & \sin(\alpha_y) \\ 0 & 1 & 0 \\ -\sin(\alpha_y) & 0 & \cos(\alpha_y) \end{bmatrix}$$

$$\bullet \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_x) & -\sin(\alpha_x) \\ 0 & \sin(\alpha_x) & \cos(\alpha_x) \end{bmatrix} \bullet \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} \qquad (12)$$

The values of E, F, and G are then substituted into the following equations, which are based on Eq. (10):

$$a = \tan^{-1}\left(\frac{\sqrt{E^2 + F^2}}{G} \right) + \alpha \quad \text{and}$$

$$b = \tan^{-1}\left(\frac{E}{F} \right) \qquad (13)$$

Equation (13) represents the general solution of the rotating tables shown in *Figures 4a* through *4d*.

Examples

In this section, two examples are demonstrated for machining ruled surfaces using the developed real-time five-axis interpolator with flank milling. Two new g-codes have been defined for the conic surface in *Figure 8* and the ruled surface in *Figure 10*. These two parts were successfully produced on a five-axis milling machine.

Example 1

The conic surface, shown in *Figure 8*, was machined on a five-axis machine by a flank cutter with the developed real-time interpolator. To machine the conic surface, a new g-code, g51, was defined, as follows:

g51 x__ y__ z__ i__ j__ k__ h__ r__ d__ f__

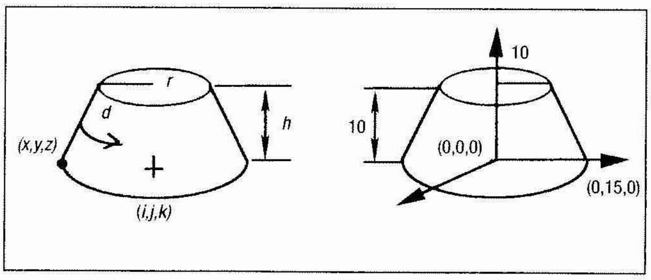

Figure 8
Example of the Conic Surface

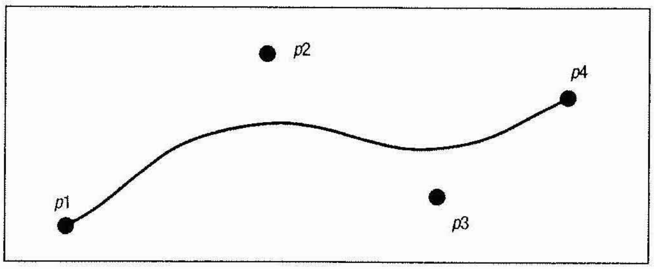

Figure 9
Example of the Bezier Curve and Its Control Points

where (x, y, z) are the initial tool contact positions on the conic surface; (i, j, k) define the center of the bottom circle of the cone; h is the height of the conic surface; r is the radius of the upper circle; d is the angle of the conic surface to be produced; and f is the surface feed rate. The dimensions in this g-code are in BLUs, where BLU equals 0.01 mm for translation and 0.001 degree for rotation. According to the conic surface in this example, fixed coordinates are assumed at the center of the bottom circle; thus, the programmed g-code is as follows:

```
n010 g51 x0 y1500 z0 i0 j0 k0 h1000 r1000
d360000 f100000
```

Example 2

A compound ruled surface was machined with the real-time interpolator on a five-axis machine. A compound ruled surface is a surface constructed by multi-ruled surfaces. In this example, each ruled surface is defined by two Bezier curves,[11] and each Bezier curve is determined by four control points. *Figure 9* shows an example of a Bezier curve with four control points. For a ruled surface, eight control points are needed.

The following new g-code, g52, was assigned to define the eight control points for machining ruled surfaces with flank milling:

```
g52 x__ y__ z__ f__
    x__ y__ z__
    x__ y__ z__
    x__ y__ z__
    x__ y__ z__
    x__ y__ z__
    x__ y__ z__
    x__ y__ z__
```

The first four lines specify the upper ruled surface, and the other four lines define the lower ruled surface. For machining a compound ruled surface, a sequence of g52 codes have to be contained in the part program. *Figure 10* shows the example of the compound ruled surface. The control points of the surface are shown below, where dimensions are in BLUs (BLU = 0.01 mm),

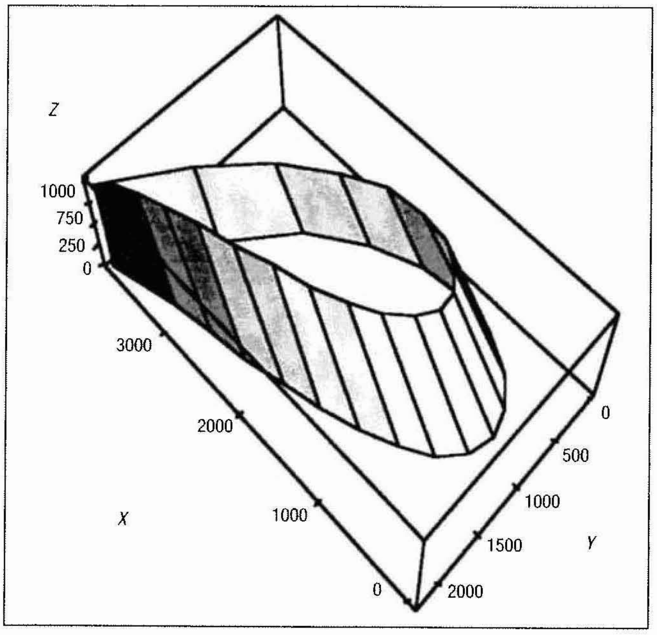

Figure 10
Example of the Compound Ruled Surface

The first ruled surface:
the upper Bezier curve
point1: (3800,2200,1000) point2: (2150,-550,1000)
point3: (500,500,1000) point4: (1050,1100,1000)
the lower Bezier curve
point1: (3800,2200,0) point2: (2150,-1100,0)
point3: (-158,330,0) point4: (360,1100,0)

The second ruled surface:
the upper Bezier curve
point1: (1050,1100,1000) point2: (1600,1650,1000)
point3: (3580,1842,1000) point4: (3800,2200,1000)

the lower Bezier curve

point1: (360,1100,0)   point2: (878,1870,0)
point3: (2810,1870,0)   point4: (3800,2200,0)

The part program to machine this compound ruled surface is shown as follows:

```
n010 g52  x3800  y2200  z1000  f10000
n020      x2150  y-550  z1000
n030      x500   y500   z1000
n040      x1050  y1100  z1000
n050      x3800  y2200  z0
n060      x2150  y-1100 z0
n070      x-158  y330   z0
n080      x360   y1100  z0
n090 g52  x1050  y1100  z1000
n100      x1600  y1650  z1000
n110      x3580  y1842  z1000
n120      x3800  y2200  z1000
n130      x360   y1100  z0
n140      x878   y1870  z0
n150      x2810  y1870  z0
n160      x3800  y2200  z0
```

Conclusions

Ruled surfaces are commonly used in industry for part surface design, such as in automobile and airplane components. To benefit from the advantages of using five-axis machines to produce these parts, developing a precise, real-time interpolator for five-axis CNC machines is required. A real-time five-axis interpolator for machining ruled surfaces with flank cutters has been presented in this paper. The real-time approach interpolates accurate tool orientations and substantially gives precise tool positions. It consists of two algorithms: (1) a generic, machine-independent algorithm for trajectory planning to generate the tool position and orientation at each sampling period, and (2) a machine-dependent algorithm for inverse kinematics transformation that generates corresponding axial position commands.

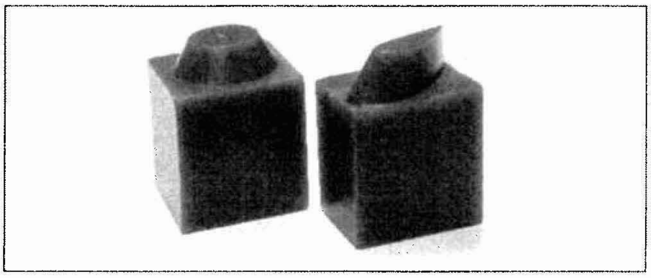

Figure 11
Demonstrated Part Surfaces Produced by Real-Time Approach on a Five-Axis Machine

Two demonstrated ruled surfaces have been successfully produced from a five-axis milling machine implemented with the developed real-time interpolator, as shown in *Figure 11*. Using the real-time five-axis interpolator, the machining time required to produce the first demonstrated example (conic surface) is 65 seconds. Only one g-code instruction is needed to describe this conic surface. If this part is produced by conventional three-axis machines, 99 g-code instructions are required. It took about 7 minutes to finish this part surface with the machining feed rate of 400 mm/min. To produce the second demonstrated example (general ruled surface), the machining time is 60 seconds by using the real-time five-axis interpolator. It requires 16 g-code instructions for this approach, as shown in the previous section. If the part is produced by a three-axis machine, hundreds of g-codes are needed.

The real-time five-axis interpolator developed in this research assumes that the part surface is gouge-free. A gouging problem occurs when the cutter radius is larger than the radius of a concave surface curvature or when the cutter collides with the other part of surface. Several algorithms were proposed to detect and avoid gouging problems.[2-5] However, these algorithms fit only the off-line approach because they require a very large computation load. It is assumed, therefore, that to obtain a gouge-free surface the part surface will be examined off-line before machining it with the real-time interpolator.

References

1. Y. Lee and T. Chang, "Machined Surface Error Analysis for 5-Axis Machining," *Int'l Journal of Production Research* (v34, n1, 1996), pp111-135.

2. Y. Lee, "Admissible Tool Orientation Control of Gouging Avoidance for 5-Axis Complex Surface Machining," *Computer Aided Design* (v29, n7, 1997), pp507-521.

3. Y. Wang and X. Tang, "Five-Axis NC Machining of Sculptured Surfaces," *The Int'l Journal of Advanced Mfg. Technology* (v15, 1999), pp7-14.

4. H. Renker, "Collision-Free Five-Axis Milling of Twisted Ruled Surfaces," *Annals of the CIRP* (v42/1/1993), pp457-461.

5. C. Jensen and D. Anderson, "Accurate Tool Placement and Orientation for Finish Surface Machining," ASME Winter Annual Mtg. (1992), pp127-145.

6. K. Sanbandan and K. Wang, "Five-Axis Swept Volumes for Graphic NC Simulation and Verification," ASME 15th Design Automation Conf. (DE v19-1, 1989), pp143-150.

7. W. Wang and K. Wang, "Real-Time Verification of Multi-Axis NC Programs with Raster Graphics," *Proc.* of IEEE Int'l Conf. on Robotics and Automation (1986), pp166-171.

8. R. Lin and Y. Koren, "Real-Time Interpolators for Multi-Axis CNC Machine Tools," *CIRP Journal of Mfg. Systems* (v25, n2, 1996), pp145-149.

9. D. Yang and T. Kong, "Parametric Interpolator Versus Linear Interpolator for Precision CNC Machining," *Computer Aided Design* (v26, n3, 1994), pp225-233.

10. J. Chou and D. Yang, "On the Generation of Coordinated Motion of Five-Axis CNC/CMM Machines," *Journal of Engg. for Industry* (v114, Feb. 1992), pp15-22.

11. Michacle E. Mortenson, *Geometric Modeling* (New York: Wiley Press, 1985).

12. M. Shpitalni, Y. Koren, and C.C. Lo, "Real-Time Curve Interpolator," *Computer Aided Design* (v26, n11, 1994), pp832-838.

13. Y. Koren and R. Lin, "Five-Axis Surface Interpolators," *Annals of the CIRP* (v44/1/1995), pp379-382.

14. I. Faux and M. Pratt, *Computational Geometry for Design and Mfg.* (New York: John Wiley & Sons, 1981).

Authors' Biographies

Rong-Shine Lin received his PhD in mechanical engineering and applied mechanics from the University of Michigan in 1994. He worked for Chrysler Corporation as a tool and process engineer in the Dept. of Advanced Manufacturing Engineering from 1994 to 1995. He was an assistant professor of industrial engineering at Rutgers University from 1995 to 1997. Currently, he is an assistant professor in the Dept. of Mechanical Engineering at National Chung Cheng University (Taiwan). His research interests include five-axis CNCs, CAD/CAM/CNC, and rapid prototyping

Yoram Koren received a BSc and an MSc in electrical engineering and a DSc (1970) in mechanical engineering from the Technion-Israel Institute of Technology. He is the Paul G. Goebel Professor of Mechanical Engineering and Applied Mechanics at the University of Michigan. He is the author of many publications on automated manufacturing and is the inventor of work covered by three US patents in robotics.