Five-Axis Surface Interpolators

Yoram Koren (1), Dept. of MEAM, The University of Michigan, USA
Rong-Shine Lin, Dept. of Advanced Manufacturing Engineering, Chrysler Corporation, USA

Received on January 10, 1995

Abstract

A new interpolator architecture for surface machining on five-axis CNC machines was developed and tested. This interpolator can handle quick and precise production of both convex surfaces by using end-mills and ruled surfaces by using flank-mills. The main ideas are (1) Decomposing the surface into curves, (2) continuously maintaining the cutter axis perpendicular to convex surfaces (end mills) or parallel to ruled surfaces (flank mills) by calculating corresponding axial reference positions at each sampling period of the interpolator. This real-time approach produces smoother surfaces and requires substantially less machining time compared to conventional off-line approaches.

Keywords: Numerical Control, Interpolator, Machine Tools

1. Introduction

There are surfaces that must be produced on 5-axis machines, and there are surfaces that may be produced either on 5-axis machines or on 3-axis machines using ball-end cutters. Five-axis surface machining has several advantages compared to three-axis surface machining, including higher metal removal rates, better surface finish, and more precise part surface in one setup [Sprow 1993, Jensen and Anderson 1992, Vickers and Quan 1989]. Five-axis machining ,however, requires programs that convert the surface into a sequence of machine axial commands. Some advanced Computer-Aided Design Computer-Aided Manufacturing (CAD/CAM) systems utilize off-line part programming approaches for 5-axis machining 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 which generates the axial commands.

Most off-line approaches for 5-axis machining assume a constant tool orientation along each segment. The more advanced methods assume 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 in the orientation is not necessarily linear), which causes surface errors. An additional major 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].

To overcome these drawbacks, we developed a new interpolator architecture for 5-axis CNC machine tools that generates precise tool orientations and tool positions in real time and continuously controls 5-axis motions. This real-time interpolator can handle both production of convex surfaces by using end mills and production of ruled surfaces by using flank mills. The interpolator calculates a new command in the same time period needed for sampling the control-loop feedback devices. It eliminates the acceleration and deceleration steps during machining; therefore, it produces smoother surfaces and requires less machining time compared to conventional off-line approaches.

The proposed real-time 5-axis interpolator for surface machining is shown in Fig. 1. The input to the interpolator is a new defined NC g-code, which contains the geometric information of the part surface as well as the cutting conditions, such as feedrate, spindle speed, specific tool, etc. The 3-D parametric surface g-codes will be given below and illustrated by examples.

For end mills, the interpolator begins with tool path planning, which generates curves from given surfaces, and subsequently executes the trajectory planning for the interpolation of each curve. The tool position (x,y,z) and orientation (O_x,O_y,O_z) of each interpolation step are calculated in real-time. The inverse kinematics portion generates the position values of each axis, which will be the reference values transmitted to the controller. For flank mills, the interpolator begins with trajectory planning and then goes through the inverse kinematics. The detailed descriptions for these portions of the interpolator are discussed in the following sections. The implementation of the proposed 5-axis interpolator on CNC systems is also discussed.

2. End-Mills

2.1 Tool Path Planning

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

For machining curved surfaces, scallops are created on the finished surface. Figure 2 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 paper. The local radius of curvature of the part surface is R. 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 [Lin and Koren; 1994]:

$$P = \frac{2R}{R+h}\sqrt{2Rh + h^2} \qquad (1)$$

Equation (1) expresses the maximum tool path interval that will keep the scallops bellow the allowable surface finish value.

To simplify the planning process, one of the boundary curves of a surface is chosen for the first tool path. Along this path, the tool path intervals are calculated by Eq.(1). 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.

2.2. Trajectory Planning

The trajectory planning algorithm generates the locations and orientations for the cutting tool based on the part geometry and given feedrate. The tool trajectory C(u) can be determined from tool-path planning and is described below

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

To produce smooth part surfaces, the machining feedrate (V) must be kept 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 (typical sampling periods are 1 ms to 10 ms).

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 (3) shows the function of u in terms of the length VT that the tool moves during one sampling period T.

$$\Delta u = u_k - u_{k-1} = \frac{VT}{\sqrt{x'^2_{k-1} + y'^2_{k-1} + z'^2_{k-1}}} \qquad (3)$$

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

The subscripts 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.(3) into Eq.(2).

Our five-axis end-mills machining controls the cutter not only to follow the determined tool paths, but also to orientate the cutter axis in the direction of surface-normal. The surface-normal direction $[O_x, O_y, O_z]$ can be calculated by [Faux and Pratt, 1981],

$$\left[O_x, O_y, O_z\right]_k = \frac{\left.\frac{\partial S}{\partial v} \times \frac{\partial S}{\partial u}\right|_{u=u_k}}{\left|\frac{\partial S}{\partial v} \times \frac{\partial S}{\partial u}\right|_{u=u_k}} \qquad 0 \le v \le 1 \qquad (4)$$

where $[O_x, O_y, O_z]_k$ is the tool orientation at the specific tool position $u = u_k$. The six variables $[x, y, z, O_x, O_y, O_z]$, Eqs.(2) and (4) are the solutions of tool position and orientation and must be determined in real-time at each sampling time.

2.3. Inverse Kinematics Transformation

The derivation so far is based on the part surface geometry and the 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 solution of inverse kinematics depends upon the structure of a machine. In this paper, we used the structure of one tilt table and one rotary table (see Fig. 3) are put on top of a three axis machine.

The solution of the inverse kinematics based on the decoupling approach [Lin and Koren, 1994] and is shown below,

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{O_x}{O_y}\right) \\ \tan^{-1}\left(\frac{\sqrt{O_x^2 + O_y^2}}{O_z}\right) \end{bmatrix} \qquad \begin{matrix} 0 \le a \le 2\pi \\ 0 \le b \le \frac{\pi}{2} \end{matrix} \qquad (5)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix} \bullet \left\{ \begin{bmatrix} \cos(b) & -\sin(b) & 0 \\ \sin(b) & \cos(b) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} P_x + R_x \\ P_y + R_y \\ P_z + R_z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \right\} + \begin{bmatrix} OT_x \\ OT_y \\ OT_z \end{bmatrix} \qquad (6)$$

The detailed coordinate system is shown in Fig.3, where 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. (5) and (6), provide the five reference inputs for the five servo-controllers that control the machine.

3. Flank Mills

Parts containing ruled surfaces, described by Eq.(7), can be machined by using flank milling on 5-axis machines that enables a tremendous increase in productivity and surface finish.

$$S(u,v) = (1-v)\, C_0(u) + v\, C_1(u) \qquad (7)$$

In the proposed real-time interpolator (see Fig. 1), flank milling needs only trajectory planning and inverse kinematics transformation. The only difference between flank-mills and end-mills is the orientation calculations. The key idea of flank machining is controlling the tool axis in parallel with the ruled surface. For each cutter's position, $u = u_k$, the cutting tool orientation can be calculated by the first derivative of Eq.(7) with respect to v, as shown below

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

where $[O_x, O_y, O_z]$ is the tool orientation at specific tool position $u = u_k$. By substituting Eq. (8) into Eqs.(5) and (6), the five reference positions can be obtained.

4. 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.

A parametric cubic surface is shown in Eq. (9).

$$S(u,v) = x(u,v)\hat{i} + y(u,v)\hat{j} + z(u,v)\hat{k} \qquad (9)$$

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 (x direction) of Eq. (9) we get

$$x(u,v) = \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} \begin{bmatrix} a & b & c & d \\ e & h & i & j \\ k & l & n & o \\ p & q & r & s \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \qquad (10)$$

Therefore, we define a g-code surface, 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__. (11)

Likewise, g54 and g55 are defined as the coefficients in y- and z-direction, respectively. The new g-codes, g53 to g55, represent a parametric cubic surface, which contains the geometric information needed for the surface interpolator.

5. Implementation

The 5-axis CNC system on which the proposed real-time interpolators were implemented has a sampling period of 10 ms, and a resolution of BLU=0.01 mm. This system consists of a 5-axis milling machine controlled by an Intel i486/33MHz personal computer, I/O interfacing boards, and sensors, such as rotary encoders, linear encoders, and tachometers. The block diagram of this CNC system is shown in Fig. 4. Every user of this system may write his/her own interpolation, control, and compensation software. The input for the system is a g-code part program. According to the specific g-codes, the interpolator generates the reference positions for the controller. The controller compares the references with the actual position and velocity feedbacks, which are read from sensors, to generate the control signals. By means of the I/O interfacing boards, these signals are transmitted to the milling machine, thus driving the tool to desired positions.

The software program for this CNC system is written in C++ programming language. This program consists of a user input interface, a module for reading/decoding g-code part programs, interpolators, controllers, and hardware interfaces. The program flow chart is shown in Fig. 5. In this program, users can easily select the input parameters, such as the part programs, controllers, and sensor types, from the graphic pull-down window menu. First, the machine will be initialized according to the chosen system parameters. Then, the part-program is read. The part program specifies what kind of interpolator is required. Each interpolator is programmed as a software subroutine. In accordance with the interpolation algorithm, the reference positions are generated, and sent to the servo-controllers. The program reads the actual table positions and velocities from sensors, compares them with the references, and transmits the control signals to the I/O interfaces.

6. Example

The 3-D surface g-codes of the example are defined below.

g53 o 3000
g54 r3000
g55 j-4000 o1000 s3000

Figure 6 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. With programmed feedrate of10 mm/sec, the machining time is 13.83 sec by using proposed real-time interpolator. The maximum interpolating contour error is 0.1 μm in this case. By contrast, the conventional off-line interpolator requires 303 straight-line segments to approximate the part surface with a maximum interpolating contour error of 1 μm, namely 10 times larger than with our approach. The machining time is 41.88 sec which is 3 times longer than with our method. In the off-line approach, one can improve the contour error by adopting more straight-lines to approximate the surface; however, this increases the machining time and causes very large feedrate variations.

7. Summary

A new interpolator architecture for surface machining on five-axis CNC machines was developed and tested. The interpolator calculates the cutter position and orientation at each sampling period of the interpolation and generates corresponding axial position commands by machine-dependent inverse kinematics algorithm. This real-time approach produces smoother surfaces and requires substantially less machining time compared to conventional off-line approaches because it eliminates the acceleration and deceleration steps.

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, pp. 15-22.

Faux, I. and Pratt, M., 1981, Computational Geometry for Design and Manufacturing, N Y, 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, ASME Winter Annual Meeting, pp. 83-92.

Lin, R. and Koren, Y., 1994, "Real-Time Five Axis Interpolator for Machining Ruled Surfaces," ASME Winter Annual Meeting, DSC-Vol. 55-2, pp. 951-960.

Mortenson, Michacle E., 1985, Geometric Modeling, Wiley Press, New York.

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", ASME the 15th Design Automation Conference, DE Vol 19-1, 143-150.

Sprow, Eugene E., 1993, "Set Up to Five-Axis Programming", Manuf. Engi., November pp. 55-60.

Vickers, G. W. and Quan, K. W., 1989, "Ball-Mills Versus End-Mills for Curved Surface Machining", ASME Journal of Eng. 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.

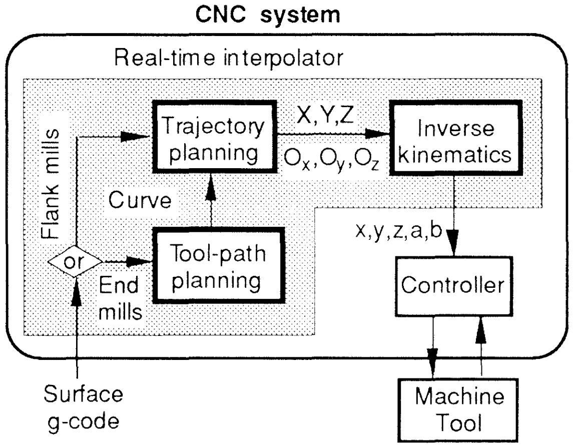

Figure 1: The proposed real-time 5-axis interpolator.

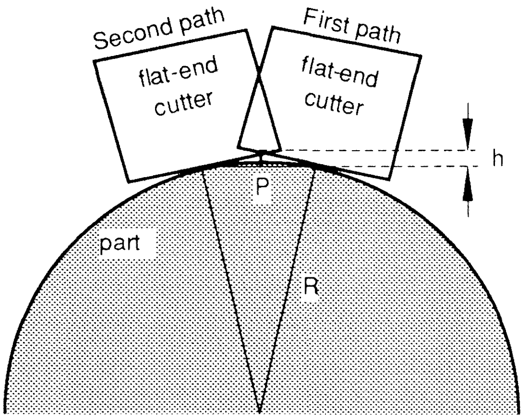

Figure 2: A convex surface machined by a flat-end cutter.

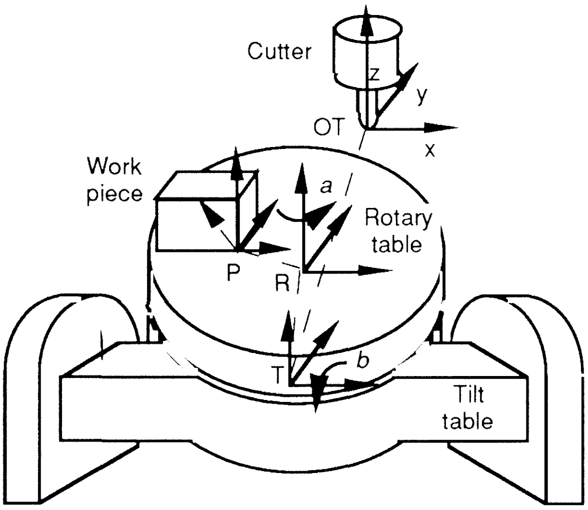

Figure 3: The coordinate systems for Eq. (6).

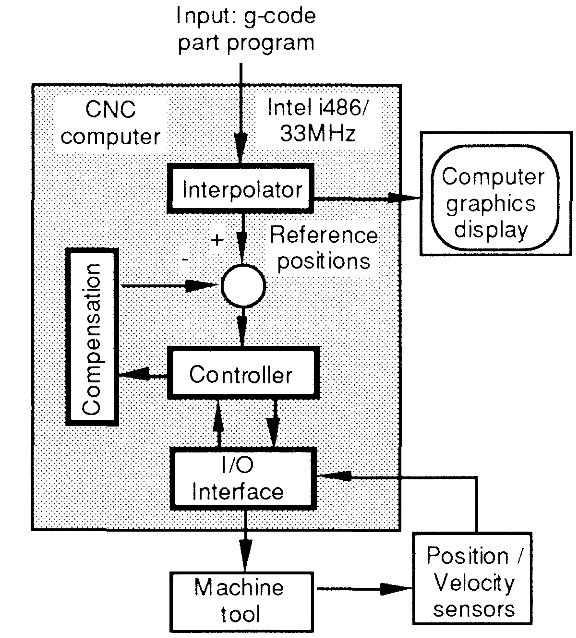

Figure 4: The structure of the implemented CNC system.

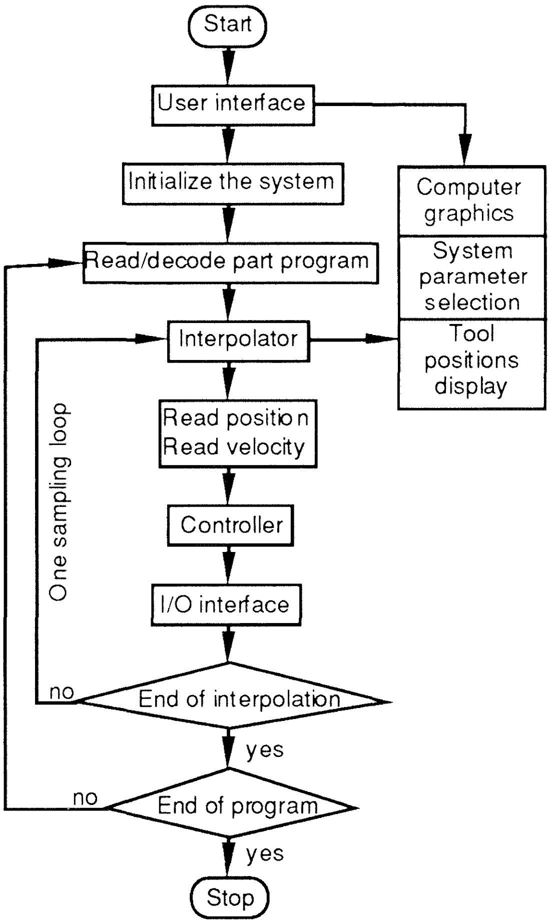

Figure 5: The flow chart of the CNC software program.

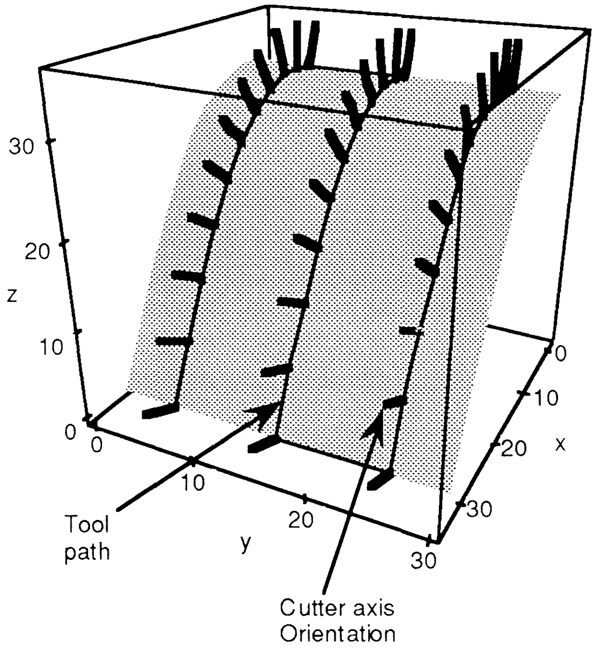

Figure 6: The g-code defined developable surface and the tool paths with tool axis orientations.