A Real-time NURBS Surface Interpolator for 5-axis Surface Machining

WANG Yong-zhang, CHEN Liang-ji
(School of Mechanical Engineering, Harbin Institute of Technology, Harbin 150001, China)

Abstract: A real-time non-uniform rational B-spline (NURBS) surface interpolator is proposed and 5-axis machining method with a flat end cutter is discussed. With the Taylor expansion and the coordinate transformation, the algorithms of NURBS interpolation, cutter effective machining radius, cutter offsetting and inverse kinematics are deduced and implemented, respectively. Different from the conventional free form surface machining, the proposed interpolator can real-time generate the motion commands of computer numerical control (CNC) machines with CNC feedrate, rather than that of CL. An example part surface is demonstrated and the results of simulation show that the proposed method can be applied in actual 5-axis surface machining.

Key words: free form surface; real-time interpolator; NURBS; 5-axis machining

Most commercial computer-aided design and manufacturing (CAD/CAM) systems are usually equipped with functions to provide users the capability of defining three-dimensional complicated curves and surfaces. Among the techniques used for representing and designing specific curves and surfaces, the NURBS [1-7] is one that currently attracts a lot of attention because NURBS offers a common mathematical form for representing and designing both standard analytical shapes and free-form curves (surfaces). By changing the values of weights, knot vectors and control points, a wide variety of shapes can be represented and designed using NURBS. However, conventional CNC systems provide only line and circular interpolators, that is, only motion along straight lines or circular paths are supported. In order to perform surface machining, the tool paths, which are also known as the cutter location (CL) paths, are typically approximated with piecewise linear or circular segments by CAD/CAM systems. As a result, the corresponding NC codes are generated and the motion controller can be applied to guide the CNC machines to execute accurate cutting. This offline type of approximation approach may result in several disadvantages [25]: (1) to satisfy the machining accuracy, the file size of the NC program is often relatively large. These machining codes will not only involve heavy data transmission loads between the CAD/CAM systems and CNC machines, but also consume a large portion of CNC memory. So in terms of the total machining time and machining cost, the conventional approach is neither efficient nor economical; (2) the cutter needs to accelerate...
and decelerate at each segment, which leads to the velocity discontinuity at the junction of two connected segments and causes a deterioration in machining accuracy. These drawbacks indicate that the conventional approach may not meet the requirements of high speed, high accuracy machining required in today’s manufacturing industry\cite{3].

In order to overcome the disadvantages of conventional CNC machining methods, interpolators, which can directly generate complex curves, need to be added into CNC systems to substitute linear interpolation methods\cite{8,12]. Behnam, et al.\cite{18} realized a real-time interpolator for planar cubic parametric curves in CNC system. Lo\cite{9,10} and Lin \cite{11} had presented real-time surface interpolation method for 5-axis ball-end free-form surface milling respectively. Although these methods had their advantages on free-form surface machining as compared with linear machining method, there existed some disadvantages on using CAD/CAM resources because these interpolators cannot machine the surfaces designed in CAD/CAM systems directly. In order to receive surfaces data from CAD/CAM systems, CNC system must be provided with the function of NURBS curves and surfaces generation. Thus an algorithm of NURBS curves and surfaces interpolation needs to be developed. A number of NURBS interpolators had been proposed by several investigators\cite{27}. Cheng, et al.\cite{3} had proposed a real-time NURBS curve motion command generator for CNC machines. Zhang and Greerway\cite{4} realized a NURBS curve motion interpolator for a 6-axis robot using an open architecture controller system as a test bed. Zhiming, et al.\cite{17} developed a NURBS curve interpolator for CNC machining based on the geometric properties of the tool path. Before free-form surface machining, their methods had to discretize the offsetting surface of part surface into CL curves and then feed these curves into interpolators to get the CL points. On the other hand, most of them had concentrated their attentions on 3-axis ball-end machining, but for 5-axis NURBS surface interpolator, little has been done. 5-axis CNC machine tools are widely used in machining dies, molds, turbine blades, and aerial parts. These parts usually have complex geometry and are represented by parametric or sculptured surfaces. As compared to 3-axis machining, the 5-axis machining with flat-end cutter offers many advantages such as higher productivity and better machining quality. However, in the existing 5-axis machining approaches, the motion commands generated by the CAM system are loaded to CNC machine that adopts a real-time interpolator. The CNC interpolator can convert the cutter path to motion trajectories of the five separate axes in order to coordinate their motion in 5-axis machining. Being similar to 3-axis machining, many 5-axis CNC interpolators still adopted the linear and the spline segments to approximate part surfaces represented by NURBS\cite{13-18}. Matthias, et al.\cite{13} had implemented algorithm of high accuracy spline interpolation for 5-axis machining based on spline segments approximation. Elber and Fish\cite{18}, however, approximated the part surface with ruled surfaces and then milled the surface in 5-axis CNC machines. The data fed to the CNC interpolators are the CL paths (position and orientation) and the CL velocity, which had been offline computed by CAD/CAM systems. CL denotes the center of the tool bottom and is usually not the location where the cutting takes place. In contrast, the cutter contact (CC) location, which denotes the intersection point of the cutter and the sculptured surface, is our main concern. For this reason, the current approaches, as illustrated in Fig. 1 (a), cannot achieve high efficiency and quality because both the generation and the control of the cutter path are based on the CL path, rather than the CC path.

In this paper, a real-time NURBS surface interpolator for 5-axis machining with flat-end cutter is developed and implemented. Fig. 1(b) shows the architecture of the interpolator, which consists of real-time algorithms for the CC path (position and orientation) planning and interpolating, the cutter offsetting, and the coordinate conversion. All the three algorithms are executed on line the machining
1 NURBS Surface Representation

A general form to describe a parametric surface in the 3-D space can be expressed as

\[ S(u, v) = x(u, v) i + y(u, v) j + z(u, v) k \quad 0 \leq u, v \leq 1 \]  

A NURBS surface is the rational generalization of the tensor product non-rational B-spline surface and is defined as follows \cite{1}:

\[ S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j} \]

where \( P_{i,j} \) are 3-D control points; \( w_{i,j} \) are the corresponding weights of \( P_{i,j} \); \((n + 1)\) and \((m + 1)\) are the number of control points in the \( u \) and \( v \) directions, respectively; \( N_{i,p}(u) \) and \( N_{j,q}(v) \) are the so-called normalized B-spline basis functions, where \( p \) and \( q \) are their degrees, respectively. The iterative calculation of \( N_{i,p}(u) \) and \( N_{j,q}(v) \) is the same as discussed in Ref. [1].

Introducing the piecewise rational basis functions

\[ R_{i,p,j,q}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{k,p}(u) N_{l,q}(v) w_{k,l}} \]  

Eq. (2) can be rewritten into the following equivalent form

\[ S'(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,p,j,q}(u, v) P_{i,j} \]  

By fixing \( v = v_0 \) and \( u = u_0 \) in Eq. (4), \( u \) and \( v \) direction is a parametric curve on \( S(u, v) \), i.e., \( S(u, v_0) \) and \( S(u_0, v) \), are obtained respectively.

2 CC Path Real-time Interpolation

The task of a real-time interpolator is to calculate each axis’ motion command of next sampling period (\( T \)) in this sampling one so as to coordinate all axes motion with the desired feedrate (CC velocity here). For 5-axis interpolator, the results of interpolation comprise not only CC points’ position, but also the corresponding cutting tool orientation.

2.1 CC points’ position calculation

Generally, the machining feedrate (CC velocity) along the \( c \)th CC path \( S(u, v_c) \) in \( u \)-direction can be represented as

\[ V_{CC} = \left| \frac{dS(u, v_c)}{dt} \right| \]  

According to the chain derivative rule, we can get

\[ V_{CC} = \left| \frac{\partial S(u, v_c)}{\partial u} \cdot \frac{du}{dt} \right| = \left| \frac{\partial S(u, v_c)}{\partial u} \right| \cdot \frac{du}{dt} \]
where the arithmetic operator \( \mathbb{M} \) denotes the Euclidean norm of a vector in the 3D space. Therefore,

\[
\frac{du}{dt} = \frac{V_{CC}}{|S'(u, v_c)|} \cdot \frac{du}{dt}
\]  

(7)

Depends on users’ requirements, the desired feedrate command can be a function of time or simply a constant. Note that, in general, users can choose the desired feedrate command based on various factors, such as the cutting tool used, the workpiece material properties, the required accuracy and the machining conditions. To implement a real-time interpolator is to determine successive values of \( u \). A computationally efficient solution of Eq. (7) can be classified into two categories—Taylor’s first or second-order expansion and the high-order Runge-Kutta method. If the approximation accuracy is of interest, the high-order Runge-Kutta method can be used. In this study, the Taylor’s approximation method is chosen to realize the NURBS surface interpolator. Hence, the second-order approximation on \( du/dt \) is given by

\[
u_{k+1} = u_k + T \left( \frac{du}{dt} \bigg|_{t = kT} + \frac{T^2}{2} \frac{d^2u}{dt^2} \bigg|_{t = kT} \right)
\]  

Higher Order Terms

(8)

where \( u_k = u(t_k) \) denotes the value of \( u \) at the \( k \)th sampling instant \( t_k = kT \). If \( T \) is small enough, its first-order approximation, which is usually adequate, can be expressed as

\[
\nu_{k+1} = u_k + T \frac{du}{dt} \bigg|_{t = kT}
\]  

(9)

Accordingly, by substituting Eq. (7) into Eq. (9), we will have

\[
u_{ls} = \nu_0 + \frac{T V_{CC}}{|S'(u, v_c)|} | \bigg|_{u = \nu_k}
\]  

(10)

Substituting the computed ( \( \nu_{k+1}, v_c \) ) into Eq. (4), we will yield the next CC point’s position vector at time of \( t_{k+1} \).

2.2 Cutting tool orientation calculation

In \( S \) axis machining, it is easier to define the tool orientation based on the local surface property than that based on the machine global coordinate system [15]. A local coordinate system (LCS) is defined to analyze the cutting operation of a flat end mill.

As shown in Fig. 2, a flat-end mill is contacting with a part surface \( S(u, v) \) at a CC point \( C \) that is computed in the previous sampling period, and LCS is represented by \( X_C \), \( Y_C \) and \( Z_C \) axes. The \( X_C \) axis is always lying in the current cutting direction, and the \( Y_C \) axis is in the surface normal direction. The \( Z_C \) axis is determined by the cross product of the \( X_C \) and the \( Y_C \) axis. In \( S \) axis machining, the tool is first rotated by an inclination angle \( \theta_C \) about the \( Z_C \) axis, then a tilt angle \( \beta_C \) about the \( Y_C \) axis, while the CC point \( C \) is the pivot, as shown in Fig. 2. Traditionally, the cylindrical tool is inclined to an angle \( \theta_C \) at a large effective cutter radius and results in a small scallop height. Consequently, without exceeding the scallop height limitation, the inclination angle can be chosen as small as possible so that a large path interval can be utilized to improve the machining efficiency (or shorten the path length). This, however, increases the possibility for

![Fig. 2 Tool coordination system and local coordination system](image-url)
rear gauging. The problem of tool interference detection and tool orientation optimization will be discussed in our future work. In this paper, to explain the proposed $\mathcal{F}$-axis NURBS surface interpolation method, we initialize the tool orientation ($\alpha_c$, $\beta_c$) as $(2\theta_0, 0^\circ)$.

2.3 Cutting tool path interval distance calculation

(1) Analysis of effective cutting shape in $\mathcal{F}$-axis machining

As shown in Fig. 2, a flat-end mill cutter is contacting with $S(u, v)$ at $C$ and the radius of the cutter is $R_T$. In order to get the effective cutting shape, an instant cutter coordinate system (TCS or $X_T Y_T Z_T$) is defined on the center $O$ of the bottom edge of the cutter. The $X_T$ axis is along $\overrightarrow{OC}$, the $Y_T$ axis is in the centro-symmetric axis direction of cylindrical miller and $Z_T$ is determined by the cross product of the $X_T$ and the $Y_T$ axis. On the $X_T Y_T Z_T$ plane, $\theta$ is the angle from $Z_T$ to a point $P$ on the cutter edge. In the $X_T Y_T Z_T$ system, the cutter edge ($E_T(\theta)$) can be expressed by the following equation

$$E_T(\theta) = \{ R_T \sin \theta \ 0 \ R_T \cos \theta \}^T$$

At first, we assume that TCS is coincided with LCS. Through the translation and rotation transformation, the cutter edge in LCS can be written as

$$E_L(\theta) = \text{ROT}(-\beta_c) \text{ROT}(-\alpha_c) \text{TRANS}([l - R_T \ 0 \ 0]) E_T(\theta)$$

(12)

where ROT($\bullet$) and TRANS($\bullet$) represent rotation and translation matrix respectively. Substitute Eq. (11) into Eq. (12), we can get

$$E_L(\theta) = \begin{bmatrix}
R_T(\sin \theta - 1) \cos \alpha_c \sin \beta_c - R_T \cos \theta \sin \beta_c \\
R_T(\sin \theta - 1) \cos \alpha_c \sin \beta_c - R_T \cos \theta \sin \beta_c \\
R_T \sin \theta \sin \beta_c + R_T \cos \theta \cos \beta_c
\end{bmatrix}$$

(13)

As shown in Fig. 3, $E_L(\theta)$ at $C$ is projected along the cutting direction $X_C$ onto the $Y_C-Z_C$ plane as the effective machining shape $E_{\text{eff}}(\theta)$. Apparently, $E_{\text{eff}}(\theta)$ on $Y_C-Z_C$ plane can be found from Eq. (13) as

$$E_{\text{eff}}(\theta) = \begin{bmatrix}
0 \\
R_T(\sin \theta - 1) \sin \alpha_c - R_T \cos \theta \sin \beta_c \\
R_T(\sin \theta - 1) \cos \alpha_c \sin \beta_c + R_T \cos \theta \cos \beta_c
\end{bmatrix}$$

(14)

According to Eq. (14), we can get the curvature formulation of the effective machining shape $E_{\text{eff}}(\theta)$ as follows

$$k_{\text{eff}}(\theta) = \frac{1}{R_{\text{eff}}(\theta)}$$

$$= \frac{\sin \alpha_c}{R_T \cos \beta_c + (\tan \beta_c - \cos^2 \alpha_c \cos \beta_c) \cos \theta}$$

(15)

While in $\mathcal{F}$-axis machining, the effective cutting radius of the CC point is our main concern. As can be seen from Fig. 2, in the tool coordinate system, the angle $\theta$ corresponding to the CC point $C$ is $\pi/2$. Substituting $\theta = \pi/2$ into Eq. (15), the effective cutting radius can be obtained as

$$R_{\text{eff}} = \frac{1}{k_{\text{eff}}(\pi/2)} = \frac{R_T \cos \beta_c}{\cos \alpha_c}$$

(16)

(2) Calculation of path interval distance

As shown in Fig. 3, the scallop height $h$ is a function of the effective cutter radius $R_{\text{eff}}$, the curvature radius along $Z_C$ axis $R_{Z_C}$, and the distance of path interval $\Delta l$. On the other hand, $h$ is usually given as the allowable error $\delta$, so $\Delta l$ is to be determined in real-time machining.
face and -1 for a concave surface. Calculation of \(R_{Zc}\) can be referred to Refs. [13, 17].

(3) Calculation of parameter increment \(\Delta u\) corresponding to \(\Delta l\)

The parameter increment \(\Delta u\) along the path interval direction can be calculated by using geometric analysis of the local area. As shown in Fig. 4, \(k_c\) is unit vector of \(Z_c\) axis, which is defined by \(i_c\) and \(j_c\), and \(k_c\) is equivalent to \(i_c \times j_c\), where \(i_c\) and \(j_c\) are unit vectors of \(X_c\) and \(Y_c\) axes respectively. On tangential plane, projecting \(S_u \Delta u\), \(S_v \Delta v\) and \(k_c \Delta l\) to \(X_c\) and \(Z_c\) respectively, the following two equations about \(\Delta u \) and \(\Delta v\) can be founded:

\[
(S_u \cdot i_c) \Delta u + (S_v \cdot i_c) \Delta v = (k_c \cdot i_c) \Delta l = 0 \tag{18}
\]

\[
(S_u \cdot k_c) \Delta u + (S_v \cdot k_c) \Delta v = (k_c \cdot k_c) \Delta l = \Delta l \tag{19}
\]

Solving Eqs. (18) and (19), we can find

\[
\Delta v = \frac{\Delta u}{S_v \cdot k_c} \tag{20}
\]

3 Algorithm of Cutter Offsetting

To realize \(S\) axis machining, the CC data discussed in Section 2 must be converted to the CL data, and then are fed to the CNC machine tools. As shown in Fig. 2, \(O\) and \(T\) are the CL point vector and the unit vector of tool orientation, respectively. \(T\) is usually set to a fixed angle off the \(Y_c\) axis during machining. Geometrically, this angle can be further decomposed into an inclination angle \((\alpha_c)\) and a tilt angle \((\beta_c)\) as mentioned in Section 2. Based on the surface and cutter geometry \((C, R_T, X_c, Y_c, \alpha_c \text{ and } \beta_c)\), the CL data (including \(O\) and \(T\)) are defined.

As shown in Fig. 2, the flat-end cutter’s \(O\) and \(T\) in TCS can be expressed as follows:

\[
(O)_T = [0 \quad 0 \quad 0]^T \quad \quad (T)_T = [0 \quad 1 \quad 0]^T \tag{21}
\]

Through the same mathematical manipulation as Eq. (12), \(O\) and \(T\) in LCS can be written as

\[
(O)_C = [-R_T \cos \alpha_c \cos \beta_c, \quad R_T \sin \alpha_c \cos \beta_c \cos \beta_c]^T \tag{22}
\]

and

\[
(T)_C = [\sin \alpha_c \cos \beta_c \cos \alpha_c \sin \beta_c \cos \beta_c]^T \tag{23}
\]

The purpose of cutter offsetting is to calculate the expressions of \(O\) and \(T\) in the workpiece coordinate system (WCS, or \(X_w\)-\(Y_w\)-\(Z_w\)) as shown in Fig. 2. According to the relationship between LCS and WCS, we can do the following transformation to solve the problem:

\[
(O)_W = \text{TRANS}(C) \cdot \text{ROT}(\text{WCS} \leftrightarrow \text{LCS})(O)_C \tag{24}
\]

and

\[
(T)_W = \text{ROT}(\text{WCS} \leftrightarrow \text{LCS})(T)_C \tag{25}
\]

where \(\text{ROT}(\text{WCS} \leftrightarrow \text{LCS})\) denotes the compound rotation matrix from LCS to WCS and is determined by Eq. (26)

\[
\text{ROT}(\text{WCS} \leftrightarrow \text{LCS}) = [i_c, \quad j_c, \quad k_c] = \begin{bmatrix}
(i_{c_x} & j_{c_x} & k_{c_x}) \\
(i_{c_y} & j_{c_y} & k_{c_y}) \\
(i_{c_z} & j_{c_z} & k_{c_z})
\end{bmatrix} \tag{26}
\]

Here, \(i_c, j_c\) and \(k_c\) is the unit vector of \(X_c, \ Y_c\) and \(Z_c\) axis respectively; \(i_c\) equals to \(S_u(u_{k+1}, v_c, \alpha_c)\), \(j_c\) equals to \(S_v(u_{k+1}, v_c, \alpha_c)\) and \(k_c\) is equivalent to \(S_u \times S_v \times S_{\alpha_c}\); \(k_c\) is determined by \(k_c = i_c \times j_c\).

Through Eq. (22) to (25), the tool’s CL motion command in each sampling period can be determined.

4 Inverse Kinematics Transformation

WCS is assumed to be identical to the machine coordination system (MCS) as shown in Fig. 2 and Fig. 5. The machine tool configuration is assumed to be a spindle-tilt type of \(S\) axis CNC machine. During the \(S\) axis CNC machining, the cutting tool is equipped on the spindle that is being rotated.
According to Eqs. (27), the two spindle rotational vectors from the pivot point to the cutter tip. A discussion in Section 3 and where calculation of \( T \) can be obtained

\[
(T)_M = (T)_w = \begin{bmatrix}
T_x \\
T_y \\
T_z
\end{bmatrix} = \begin{bmatrix}
\cos \theta_C \cos \theta_B & \sin \theta_C & 0 \\
- \sin \theta_C \cos \theta_B & \cos \theta_C & 0 \\
0 & 0 & 1
\end{bmatrix} \cdot \begin{bmatrix}
\cos \theta_B & 0 & \sin \theta_B \\
0 & 1 & 0 \\
- \sin \theta_B & 0 & \cos \theta_B
\end{bmatrix}_{\text{ROT}}^{/ \theta_c}
\]

where calculation of \( (T)_w \) and \( (O)_M \) has been discussed in Section 3 and \( l \) is the tool length measured from the pivot point to the cutter tip. According to Eqs. (27), the two spindle rotational angle \( \theta_B \) and \( \theta_C \) can be determined by Eqs. (29) and (30).

\[
\theta_B = \text{sgn}(T_x) \arccos(T_z) \quad (29)
\]

\[
\theta_C = \arctan(T_y / T_x) \quad (30)
\]

In order to control the mill cutter to machine the part surface along the CC paths in 5-axis CNC machining, the reference pivot points \( (P_S)_M \) of the CNC machine tool need to be guided to the real-time calculating result of Eq. (28) with the calculated spindle orientation \( (\theta_B, \theta_C) \) of Eqs. (29) and (30), as shown in Fig. 5.

5 Computer Implementation Procedures for the Proposed Method

Based on the discussions in Sections 2, 3 and 4, the procedures for implementing the proposed 5-axis real-time NURBS surface interpolator are summarized as follows:

1. Using the isoparametric CC path planning method, and choosing one of the surface parameters \( (e.g. u) \) as the feeding direction and the other one \( (v) \) as the path interval direction. Thus the two boundary curves \( S(u, 0) \) and \( S(u, 1) \) are chosen as the first and the last CC path, respectively.

2. Inputting the given part surface (NURBS representation), feedrate command (CC velocity), flat-end cutter radii, machining scallop height limit and the initial tool orientation.

3. Fixing \( v = v_c \), as the \( c \)th CC path \( S(u, v_c) \), and then performing the CC path interpolation method to compute \( u_{c+1} \) using Eq. (10).

4. Substituting the computed \( (u_{c+1}, v_c) \) from Step (3) into Eq. (4), generating the next CC position command and then calculating the next CL data with Eqs. (24) and (25). Simultaneously, using Eqs. (28), (29) and (30) to find the spindle pivot point location \( P_S \) and the two spindle rotational angle \( \theta_B \) and \( \theta_C \).

5. For each CC point, \( S(u_{c+1}, v_c) \) calculating the radius of curvature in \( v \) direction and the flat-end tool’s effective machining radius to find the parameter increment \( \Delta u_{c+1} \) along the path interval direction using Eqs. (17) and (20).

6. Repeating from Step (3) to (5) until the end of the current path \( S(u, v_c) \). At the end of the path, choosing the minimum one of the parameter increments obtained along the path, i.e. \( \Delta v = \min \{ \Delta u_c \} \), and then setting the next CC path \( S(u, v_{c+1}) \) at \( v_{c+1} = v_c + \Delta v \).
Repeating Step (3) to (6) until the side spatial parameter reaches the other boundary \((v = 1)\) and completing the part surface machining.

6 Example

The 5-axis real-time interpolator proposed in this paper has been implemented using Visual C++ and was executed on a personal computer with Pentium IV 1.7 GHz CPU. The control points of a designed part surface are first input from surface design module and then the surface is generated and shown in Fig. 6. The parameters of the end cutter are: \(l = 150 \text{ mm} \) and \(R_T = 10 \text{ mm}\). The desired feedrate along a CC path is 100 \(\text{mm/s}\) and the sampling period of the CNC system is set as 1.0 ms. The CC path is scheduled to be in the \(u\)-direction, while the path interval is in the \(r\)-direction. The tool orientation \((\alpha_C, \beta_C)\) is initialized as \((20^\circ, 0^\circ)\). The allowable machining error is 0.001 mm. In the following sections, several experiments on machining the NURBS surface as depicted in Fig. 6 are given to evaluate the performance of the proposed method as compared with the conventional method, where the feedrate is along a CL path and the ball-end cutter radius is 10 mm.

6.1 Simulation experiments along a CC path

Inputting the cutter and machining parameter into the simulation system, the CC paths along \(u\)-direction can be easily generated with the algorithm discussed in Section 5. Choosing the first CC path \(i.e., S( u, 0)\) as the subject investigated, the following simulation results can be found.

(1) CC and CL feedrate

The feedrate experiment results on machining the CC path using the conventional method with constant feedrate along the CL path are shown in Fig. 7. As the CL path is approximated by many straight line segments without exceeding the limitation of machining error, the feedrate cannot be strictly constant from this segment to next one, that is, the feedrate is discontinuity at the junction of two connected segments as demonstrated in Fig. 7. The fluctuation of the CL velocity results in a variable CC feedrate, which leads to nonuniform unsatisfactory machining. Note that as shown in Fig. 7, in order to keep a constant CL feedrate, the CC velocity increases when the CC path is concave decreases when convex.

Fig. 7 CC and the CL feedrate in conventional method

Experiment results on machining the CC path using the proposed method with constant CC feedrate along the CC path are shown in Fig. 8. As the next CC points on part surface is real-time calculated and the distance of any two adjacent points is nearly equivalent, the CC velocity is basically kept constant and consecutive. As can be seen from Fig. 8, variation of the CL feedrate is continuous and arrives at an extremum when the curvature is the maximum. In addition, one interesting observation is that in order to keep a constant CC feedrate along a CC path, the CL velocity decreases when the CC path is concave, and the CL feedrate increases when convex.

(2) Chordal error

From the CC path interpolation algorithm discussed in Section 2.1, we know that all the CC points lie on the machined surface. This means the
proposed method not involving the cumulated errors, but only the chordal errors. So the chordal error, which is the function of the desired CC feedrate, sampling interval and curvature, is our main concern. As shown in Fig. 9, the proposed method can achieve a higher machining accuracy compared with the conventional one not only due to the interpolation principle but also due to the fact that the cutting occurs at the CC point, not the CL point.

(3) Time of interpolation calculation

The proposed interpolator is a real time method, which needs to calculate the motion commands of each axis in a given sampling interval. In each sampling period, the CNC machine tool has to complete interpolation and controlling task. Consequently, the time occupied by the interpolator is only a fraction of the sampling period. Fig. 10 demonstrates the time of interpolation calculation is about 0.4 ms and the algorithm is very stable. So proposed NURBS surface interpolator in this paper can primarily meet the real-time requirement of interpolation calculation.

6.2 Inverse kinematics

In Section 4, the algorithm of each axis motion command had been discussed. When the five independent axes are synchronously controlled to their own position, the cutter can be contacted with the machined surface at the CC point. Fig. 11 (a) and (b) demonstrate the motion of translational and rotational axes respectively.

7 Conclusion

Instead of decomposing a free-form surface represented by NURBS surface into a set of straight lines inside the CAD/CAM system (off-line machining method), a real-time interpolator for 5-axis machining with a flat-end miller is presented in this paper. Difference from the conventional 5-axis surface machining, the input to the proposed interpolator is not G-code but the machined surface, the cutter geometry, and some other machining process information. The cutter location points are real-time generated by the interpolator. In addition, the proposed method maintains a constant CC feedrate along a cutting path to improve machining efficiency and quality. The results
of simulation indicate that the proposed NURBS surface interpolator can real-time generate the motion commands for each axis servo controller and achieve higher machining accuracy compared with the conventional 5-axis surface interpolator.

References


Biographies:

WANG Yong zhang Born in 1941. He is a professor and doctoral tutor of Harbin Institute of Technology (HIT). He has completed 22 scientific research items and 5 monographs and has published more than 70 academic papers in recent years. Now, his research interests include NC technology, application of computer in machining and design/machining/controlling of modern manufacturing systems. Tel: (0451) 86413812, Email: nc_mach@hit.edu.cn

CHEN Liang jǐ Born in 1978. He received bachelor’s and master’s degree from HIT in 2001 and 2003 respectively, and then became a Ph. D. candidate of HIT in 2003. And now, his research interests include CAD/CAM for complex surfaces and 5-axis CNC machining technology of free form surfaces. Tel: (0451) 86413812, Email: chenjiaxidj@163.com