Realization of a 5-axis NURBS Interpolation with Controlled Angular Velocity

LIU Yuana, LI Huia,*, WANG Yongzhangb

a School of Astronautics, Harbin Institute of Technology, Harbin 150001, China
b School of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China

Received 24 March 2011; revised 15 June 2011; accepted 29 August 2011

Abstract

5-axis machine tool plays an important role in high-speed and high-precision computer numerical control (CNC) machining of workpieces with complex shapes. A non-uniform rational B-spline (NURBS) interpolation format for 5-axis machining is proposed to adapt to the high speed machining (HSM). With this interpolation format, angles between orientation vectors are chosen as parameters of orientation B-spline constructed by an open controller to achieve reasonable orientation vectors in real-time interpolation process. Coordinated motion between linear axes and rotary axes is achieved by building a polynomial spline which relates interpolation arc lengths of position spline to angles of orientation spline. Algorithm routine of this interpolation format and its realization methods in the supported controller are discussed in detail. Finally, performance of the proposed NURBS interpolation format is demonstrated by a practical example.

Keywords: CNC system; sculptured surface; motion control; NURBS interpolation; multi-axis machining

1. Introduction

Nowadays, the method of producing sculptured surface on workpieces is of great importance to the industries of aerospace, automobile, steamboat, cutter and die-making, etc. [1]. Machining processes of these workpieces are widely carried out by 5-axis computer numerical control (CNC) machine tools. The traditional milling method, which uses piecewise linear segment to approximately approach the profile curve segment to be milled, however, results in low efficiency and poor surface quality [2-3].

As a solution, curve interpolation function for 5-axis machining has been becoming a hotspot with advantages of decreased NC files, reduced cutting time, better accuracy, and improved surface finish. In reality, this technique is only accomplished by a few noted companies such as SIEMENS, FANUC, DELTATAU and DMG [4-7]. In research, a number of researchers have long devoted themselves to the curve interpolation for NC machining. Langeron, et al. [8] gave a command format in which position vectors and orientation vectors in a tool path were separately interpolated by the non-uniform B-spline. However, their work is only located at the computer aided manufacture (CAM) stage without considering real-time interpolation. Chen, et al. [9-10] adopted two non-uniform rational B-spline (NURBS) curves to fit position vectors and orientation vectors respectively. Fleisig, et al. [11] interpolated position vectors with a amended quintic polynomial spline, and interpolated orientation vectors with quintic spherical Bezier spline.

The rotary axis control which is a key factor influencing the interpolation properties is not fully considered in the above research. To obtain a better coordinated motion of rotary axes, orientation spline con-

*Corresponding author. Tel.: +86-451-86413440. E-mail address: lihuia@hit.edu.cn
Foundation item: National CNC Major Project of China (2009zx04002-051)
structured here takes angular dimensions between adjacent orientation vectors as the curve parameter, and arc lengths of position spline will be related to angles of orientation spline. Taking open modular architecture controllers as reference, a configurable motion controller has been developed by us [12]. The 5-axis NURBS interpolation format is realized in this controller authentically.

2. Format of NURBS Interpolation and Translations of NC Programs

2.1. Format of NURBS interpolation

Machining of machine parts requires generating tool paths, which determines the motions of tool with respect to the part to be cut [13-14]. 3-axis tool paths are represented by a series of Cartesian position vectors, each defining the location of the tool with respect to the part. For 5-axis tool paths, orientation of the tool must also be defined. Consequently, a 5-axis tool path is represented by a set of vectors: position vectors and orientation vectors (unit vectors). The position vector $P$ and the orientation vector $T$, correspond to a certain machining position.

When an NC program is programmed using a CAM software, generation methods of tool path will be determined at first. Then path planning will be carried out with the initial inclination angle and swivel angle considered. Interference checking and cutter location modifying are demanded throughout the whole process. The cutter center point coordinates and the corresponding tool-axis unit vectors can then be calculated. With these vectors, an NURBS curve will be created to fit the tool path and angles of machine tool rotary axes will be calculated. Thus, the NC code expressed by Fig. 1 is generated (taking a typical tilting rotary table type 5-axis CNC machine tool which has three linear axes $X$, $Y$ and $Z$, and two rotary axes $B$ and $C$ as an example).

![Fig. 1 Format of the proposed NURBS interpolation.](image)

In Fig. 1, the program segment begins with G05.6, which is regarded as the starting flag of NURBS interpolation. $P$ presents degree of the NURBS curve. The coordinators of position NURBS spline control point are denoted by $X_-$, $Y_-$, and $Z_-$, and the following $R_-$ shows weighted value of position curve. $B_-$ and $C_-$ express angles of the rotary axes. Knot vector of the NURBS curve is presented by $K_-$. $F_-$ is the feed velocity command. For convenience of calculation of coordinated spline behind, the number of control points of position spline is defined to be more than the number of angles of rotary axes by two.

2.2. Translations of NC programs

As mentioned above, the format here is realized in the open controller developed by us in the early days, and the NC program will be translated by the task generator module when the NC program is transferred into the controller. Special data structures (described in C++) for this module are developed to store information in the NC program. The data structures are as follows.

```cpp
typedef enum {LINE, CSCURVE, NURBS} // defining motion segment type
typedef struct NURBS_struct {
    double traverse_rate, start_rate, end_rate;
    double *tparx, *tpary, *tparz; // orientation vectors
    int P; // degree of the position spline
    double *nodevt; // knot vector of the position spline
    double *con1pt, *con2pt, *con3pt; // control points of the position spline
    double *rval; // weight value corresponding to control point
    double *tnodevt; // knot vector of the orientation spline
    double *tcon1pt, *tcon2pt, *tcon3pt; // control points of the orientation spline
    double *SepAngle; // included angle of adjacent orientation vectors
    double *ArcLength; // arc length corresponding to each knot (taking initial point of the position spline as calculating start)
    double *coordvt; // knot vector of the coordinated spline
    double *coord1pt, *coord2pt, *coord3pt; // control points of the coordinated spline
    double *machB, *machC; // angles of the orientation axes
    ...
} NURBS;
typedef struct singleStep2_struct {
    SINGLESTEP_TYPE singleStep_type;
    m_struct m_struct;
    line line_struct;
    spspline spspline_struct;
    NURBS NURBS_struct;
    ...
} singleStep2;
```

The information in the “singleStep_deque” deque is converted into the axis group module for interpolation after further calculation of orientation spline and coordinated spline.
3. Generation of Orientation Spline and Coordinated Spline

3.1. Generation of Orientation spline

We suppose that angles of the orientation axes stored in array “machB” and array “machC” in the NURBS structural body separately are \( \{b_0, b_1, \ldots, b_n\} \) and \( \{c_0, c_1, \ldots, c_n\} \). The corresponding orientation vectors (unit vector) \( q_i (i=0, 1, \ldots, n) \) in one machining path can be achieved by

\[
q_i = \begin{bmatrix}
\sin b_i \cos c_i \\
\sin b_i \sin c_i \\
\cos b_i
\end{bmatrix}
\]

To achieve continuous angles of machine tool rotary axes in a machining process, a curve should be constructed through these orientation vectors compared to the position NURBS spline. We choose the non-uniform cubic B-spline curve as an interpolation curve for orientation vectors mentioned above. In the previous research, the chord between two adjacent orientation points is commonly taken as the knot value of one knot vector calculated is supposed to be in array “machB” and array “machC” in the NURBS structural body separately are \( \{0, b_1, \ldots, b_n\} \) and \( \{0, c_1, \ldots, c_n\} \). The angle parameters of the orientation B-spline curve. In the previous research, the chord between two adjacent orientation points is commonly taken as the knot value of one knot vector calculated is supposed to be in array “machB” and array “machC” in the NURBS structural body.

\[
S(u_1, u_2) = \int_{u_1}^{u_2} \sqrt{(x'(u))^2 + (y'(u))^2 + (z'(u))^2} \, du
\]

To calculate this integral equation, a composite Simpson rule using four subintervals is adopted [16]. Firstly, dividing the parameter interval \([u_1, u_2]\) into two equal subintervals \([u_1, (u_1+u_2)/2]\) and \([(u_1+u_2)/2, u_2]\), and then halving subintervals \([u_1, (u_1+u_2)/2]\) and \([(u_1+u_2)/2, u_2]\) one more time, the culminating subintervals will be \([u_1, 3(u_1+u_2)/4]\), \([(3u_1+u_2)/4, (u_1+u_2)/2]\), \([(u_1+u_2)/2, u_2]\), \((u_1+u_2)/4]\) and \([u_1+3u_2)/4, u_2]\).

With a presupposition

\[
f(u) = \sqrt{(x'(u))^2 + (y'(u))^2 + (z'(u))^2}
\]

a Simpson expression approximating the integral equation is performed as follows:

\[
S(u_1, u_2) = \frac{b - a}{12} [f(a) + 4f \left( \frac{3a + b}{4} \right) + 2f \left( \frac{a + b}{2} \right) + 4f \left( \frac{a + 3b}{4} \right) + f(b)]
\]

In the task generator module, arc length between the adjacent knot values \(u_i\) and \(u_{i+1}\) will be calculated by using Eq. (5). These arc length values will be stored in the array “ArcLength” in the NURBS structural body.

Further, a cubic polynomial spline curve will be constructed in which angle-parameter \(\lambda\) of orientation spline curve will be taken as the function value, and arc length \(S\) of the position spline curve will be parameter of the polynomial spline curve. The interpolation polynomial spline curve is expressed as follows [17]:

\[
\lambda(S) = \frac{(S_j - S)^3}{6h_{j+1}} M_j + \frac{(S - S_j)^3}{6h_{j+1}} M_{j+1} + \frac{h_{j+1}^2}{6} M_j - \frac{h_j^2}{6} M_{j+1}
\]

\[
\lambda(S) = \frac{S - S_j}{h_{j+1}} M_j + \frac{S_j - S}{h_{j+1}} \left( \lambda_j - \lambda_{j+1} \right) - \frac{h_j^2}{6} (M_{j+1} - M_j)
\]

\(j = 0, 1, \ldots, n+1\)

where \(h_j\) equals \(x_{j+1} - x_j\), and \(M_j\) is coefficient of the spline.

3.2. Generation of coordinated spline

In the real-time interpolation process executed by the NURBS controller, knot values of the position NURBS spline curve will be calculated at first. As methods in Refs. [8]-[10], taking parameters of position curve and parameters of orientation curve as the same values will make the angular velocity not conformed to reality and therefore the effective feed velocity will be uncontrolled. This effect will be the greatest where there is a small feed relative to the angular velocity [11]. Fleisig, et al. takes an orientation reparameterization spline to solve this problem. However, with their reparameterization spline, chords of the position spline are related to angles of orientation spline in which chords cannot exactly reflect distance of the interpolated path. Here, we take arc length of the NURBS position curve as the parameters related to angle parameters of the orientation B-spline curve.

Arc length \(S\) of one NURBS curve, from knot value \(u_1\) to knot value \(u_2\), can be expressed by

\[
S(u_1, u_2) = \int_{u_1}^{u_2} \sqrt{(x'(u))^2 + (y'(u))^2 + (z'(u))^2} \, du
\]

To calculate this integral equation, a composite Simpson rule using four subintervals is adopted [16]. Firstly, dividing the parameter interval \([u_1, u_2]\) into two equal subintervals \([u_1, (u_1+u_2)/2]\) and \([(u_1+u_2)/2, u_2]\), and then halving subintervals \([u_1, (u_1+u_2)/2]\) and \([(u_1+u_2)/2, u_2]\) one more time, the culminating subintervals will be \([u_1, 3(u_1+u_2)/4]\), \([(3u_1+u_2)/4, (u_1+u_2)/2]\), \([(u_1+u_2)/2, u_2]\), \((u_1+u_2)/4]\) and \([u_1+3u_2)/4, u_2]\).

With a presupposition

\[
f(u) = \sqrt{(x'(u))^2 + (y'(u))^2 + (z'(u))^2}
\]

a Simpson expression approximating the integral equation is performed as follows:

\[
S(u_1, u_2) = \frac{b - a}{12} [f(a) + 4f \left( \frac{3a + b}{4} \right) + 2f \left( \frac{a + b}{2} \right) + 4f \left( \frac{a + 3b}{4} \right) + f(b)]
\]

In the task generator module, arc length between the adjacent knot values \(u_i\) and \(u_{i+1}\) will be calculated by using Eq. (5). These arc length values will be stored in the array “ArcLength” in the NURBS structural body.

Further, a cubic polynomial spline curve will be constructed in which angle-parameter \(\lambda\) of orientation spline curve will be taken as the function value, and arc length \(S\) of the position spline curve will be parameter of the polynomial spline curve. The interpolation polynomial spline curve is expressed as follows [17]:

\[
\lambda(S) = \frac{(S_j - S)^3}{6h_{j+1}} M_j + \frac{(S - S_j)^3}{6h_{j+1}} M_{j+1} + \frac{h_{j+1}^2}{6} M_j - \frac{h_j^2}{6} M_{j+1}
\]

\[
\lambda(S) = \frac{S - S_j}{h_{j+1}} M_j + \frac{S_j - S}{h_{j+1}} \left( \lambda_j - \lambda_{j+1} \right) - \frac{h_j^2}{6} (M_{j+1} - M_j)
\]

\(j = 0, 1, \ldots, n+1\)

where \(h_j\) equals \(x_{j+1} - x_j\), and \(M_j\) is coefficient of the spline.

4. Curve Real-time Interpolation and Inverse Kinematics Transformation

4.1. Curve real-time interpolation operation

When the above-mentioned non-real-time work is done, button “autorun” becomes effective, and then the axis group module will carry out real-time path inter-
polation operation. For the position curve \( p(u) \), the task of real-time interpolation is to calculate the parameter value \( u_{k+1} \) in next interpolation period based on the corresponding parameter value \( u_k \) of position spline in the current period. The first order approximation interpolation algorithm which is based on Taylor’s expansion method is adopted to calculate parameter \( u_{k+1} \) with an equation \([18-19]\):

\[
\begin{equation}
\begin{aligned}
u_{k+1} &= u_k + \frac{V(u_k)T}{\| \frac{dp(u)}{du} \|_{u=u_k}}
\end{aligned}
\end{equation}
\]

where \( T \) is the controller’s interpolation period, \( V(u_k) \) the instantaneous feed velocity according to authorized velocity processing methods as mentioned in Refs. \([20]-[22]\).

With the calculated position parameter \( u_{k+1} \), the position vector in the next period can be achieved by substituting \( u_{k+1} \) into expression \( p(u) \). The calculated position vector is supposed to be \( p_k\times (x_{k+1}, y_{k+1}, z_{k+1}) \).

To achieve the corresponding orientation vector in the following interpolation period, arc length related to parameter \( u_{k+1} \) from the position curve beginning will be gained adopting Simpson rule described in Section 3.2. And the resulted arc length is supposed to be \( S_{k+1} \). By substituting \( S_{k+1} \) into the polynomial spline curve function \( A(S) \) shown by Eq. (6), the angle parameter is calculated as \( \lambda_{k+1} \). Then, the orientation vector \( q_{k+1}(g_{k+1}, v_{k+1}, w_{k+1}) \), corresponding to \( p_{k+1} \), will be achieved by substituting \( \lambda_{k+1} \) into orientation B-spline curve function \( q(\lambda) \).

4.2. Inverse kinematics transformation operation

The orientation vector \( q_{k+1} \) calculated in the axis group module cannot be adopted by the servo system directly. They should be transformed into the joint space to generate motion command of axes. This process is finished by an inverse kinematics transform module developed as a personalized module for the open controller. This module integrates a variety of inverse kinematics transform algorithm which is essential for 5-axis CNC machining. The inverse kinematics transform operation is previously carried out in CAM stage of NC file programming.

Here is an example of inverse kinematics with a typical tilting rotary table type 5-axis CNC machine tool which has three linear axes, \( X, Y \), and \( Z \), and two rotary axes, \( B \) and \( C \). The motion command of rotary axes is supposed to be \( B_{k+1} \), \( C_{k+1} \) in joint space of the CNC machine tools:

\[
\begin{align}
B_{k+1} &= \arctan \left( \frac{\sqrt{g_{k+1}^2 + v_{k+1}^2}}{w_{k+1}} \right) \quad w_{k+1} > 0 \\
B_{k+1} &= 90^\circ \quad w_{k+1} = 0 \quad (8) \\
B_{k+1} &= 180^\circ + \arctan \left( \frac{\sqrt{g_{k+1}^2 + v_{k+1}^2}}{w_{k+1}} \right) \quad w_{k+1} < 0
\end{align}
\]

\[
\begin{align}
C_{k+1} &= \arctan \left( \frac{v_{k+1}}{g_{k+1}} \right) \quad g_{k+1} \geq 0, v_{k+1} \geq 0 \\
C_{k+1} &= 180^\circ - \arctan \left( \frac{v_{k+1}}{g_{k+1}} \right) \quad g_{k+1} \leq 0, v_{k+1} \geq 0 \\
C_{k+1} &= 180^\circ + \arctan \left( \frac{v_{k+1}}{g_{k+1}} \right) \quad g_{k+1} \leq 0, v_{k+1} \leq 0 \\
C_{k+1} &= 360^\circ - \arctan \left( \frac{v_{k+1}}{g_{k+1}} \right) \quad g_{k+1} \geq 0, v_{k+1} \leq 0 \\
\end{align}
\]

For Eqs. (8)-(9), to make these two expressions have result values when \( g_{k+1} \) is equal to zero, there is an equation as follows:

\[
\arctan \left( \frac{v_{k+1}}{g_{k+1}} \right) = 90^\circ \quad (10)
\]

5. Experiment and Results

The motion controller with NURBS interpolation function proposed in this paper is equipped on a 5-axis CNC milling machine tool, XKV715, jointly developed by the Qier Machine Tool Group Company Limited and the CNC Research Center of Harbin Institute of Technology. The machine tool which has three linear axes, \( X, Y \), and \( Z \), and two rotary axes, \( B \) and \( C \), is shown in Fig. 2. The NURBS controller is running as an application program in an industrial computer, AXIOMTEK workstation (Pentium IV 3.0 GHZ, RAM 1 GB), with a preestablished interpolation period 0.002 5 s.

Fig. 2 5-axis machine tool equipped with the proposed motion controller.

A multi-axis tool path to be machined with the proposed CNC system, which is shown in Fig. 3, is taken as an example for analyzing the NURBS interpolation algorithm practically.

Fig. 3 A multi-axis tool path to be machined with the proposed CNC system.
Figures 4(a)-4(b) show the given position NURBS spline curve and the orientation B-spline curve constructed with generation methods of orientation curve described in this paper respectively. The points which have marks of pentagram in the two figures are control points for the two curves respectively. The line through control points is the control polygon.

To observe fluctuation of velocity and acceleration conveniently, the controller will interpolate this tool path using a constant feed velocity of 3600 mm/min (60 mm/s) in real time. The multi-axes tool path in Fig. 3 is machined with an NC program shown in Fig. 5.

The NC program in Fig. 5 is transferred into a simulation controller generated from the NURBS motion controller. A series of pre-set data is collected when the controller is running. Fig. 6 shows the feed velocity for the tool path in Fig. 3. The fluctuation quantity of feed velocity is approximated to near constant feed velocity within a range of 0.06 mm/s. It is no greater than 0.1% error for the feed velocity of 60 mm/s. The angular velocity is achieved through the coordinated polynomial spline curve and is not a constant. The angular velocity changes smoother than that with linear interpolation methods.

To observe fluctuation of velocity and acceleration conveniently, the controller will interpolate this tool path using a constant feed velocity of 3600 mm/min (60 mm/s) in real time. The multi-axes tool path in Fig. 3 is machined with an NC program shown in Fig. 5.

The NC program in Fig. 5 is transferred into a simulation controller generated from the NURBS motion controller. A series of pre-set data is collected when the controller is running. Fig. 6 shows the feed velocity for the tool path in Fig. 3. The fluctuation quantity of feed velocity is approximated to near constant feed velocity within a range of 0.06 mm/s. It is no greater than 0.1% error for the feed velocity of 60 mm/s. The angular velocity is achieved through the coordinated polynomial spline curve and is not a constant. The angular velocity changes smoother than that with linear interpolation methods.
As shown in Figs. 7-8, velocity and acceleration of the individual axes are continuous. There are no uncontrolled spikes and they are all in an acceptable fluctuation range.

The tool path in Fig. 3 is machined with the double-NURBS (DNURBS) interpolation method referred in Ref. [1] and with the proposed NURBS interpolation method respectively. The interpolation error of rotary axes is weighed with the angular deviation. The calculated angular deviation values within the circle of curvature in each interpolation point are collected to form error charts separately for $B$ axis and $C$ axis as shown in Fig. 9.

It is obvious that with the proposed NURBS interpolation method, the machine tool has a higher accuracy characteristic in rotary axes interpolation. Furthermore, the NURBS interpolation is able to greatly enhance the machining performance and improve the overall machining accuracy.

Figure 10 shows photographs of the successfully machined sample workpieces for tool path in Fig. 3. Figures 10(a)-10(b) show the machining part and the end product respectively.

6. Conclusions

An NURBS interpolation format for 5-axis machining is proposed and it is realized in the open motion controller developed by us.

1) The human-computer interaction process, code analysis process, orientation curve generation, and coordinated curve construction are all implemented by non-real-time process. While, real-time interpolation for position curve and orientation curve, as well as
inverse kinematics transformation operation are executed in real-time process. The hard real time performance of CNC motion controller is met adequately.

2) Machining accuracy is guaranteed through choosing angles between orientation vectors as parameters of orientation and relating arc length values of position curve to angles of orientation curve.

3) Compared with the DNUURBS interpolation method, the machining accuracy is markedly increased through adopting more reasonable controlling method of rotary axes.

4) The proposed NURBS interpolation method shows a great promise in the future through actual processing and result analysis.

References


Biographies:

LIU Yuan received M.S. and Ph.D. degrees from Harbin Institute of Technology in 2007 and 2010 respectively, and then became a teacher there. His main research interests are open architecture CNC system, CAD/CAM of aerocraft, flight dynamics and control of spacecraft. E-mail: dr.yuanliu@gmail.com

LI Hui received M.S. degree from Harbin Institute of Technology in 2004, and then became a teacher there. His main research interests are CAD/CAM of aerocraft and reliability assessment of spacecraft. E-mail: lihuis@hit.edu.cn

WANG Yongzhang is a professor and doctoral tutor of Harbin Institute of Technology. Now, his main research interests include NC technology, application of computer in design, machining and controlling of modern manufacturing systems. E-mail: nc_mach@hit.edu.cn