NURBS Interpolation Technology in CNC System Based on STEP-NC

DU Juan[1),2)], TIAN Xi-tian[1)], ZHANG Zhen-ming[1)], ZHU Ming-quan[1)], LI Jian-ke[1)]

1) (School of Electromechanics, Northwest Polytechnical University, Xi'an 710072)

2) (School of Electromechanics, Taiyuan University of Science & Technology, Taiyuan 030024)

Abstract STEP-NC is a new interface standard for data exchanging and sharing between CAD/CAM and CNC, and the CNC based on STEP-NC will be the next generation of CNC controller, which not only holds linear and circular interpolation but also possesses the capability of spline interpolation. A universal NURBS (non-uniform rational B-spline) based interpolator was designed and the interpolation technique based on constant arc increment and interpolation algorithm were inverstigated arc increment. The validity and reliability of algorithm was tested by an instance simulation and machining.

Keywords STEP-NC, non-uniform rational B-spline (NURBS), spline interpolation, constant arc increment interpolation

基于 STEP-NC 的 CNC 系统中 NURBS 插补技术研究

杜 娟[1),2)] 田锡天[1)] 张振明[1)] 朱名铨[1)] 李建克[1)]

1)(西北工业大学机电工程学院，西安 710072) 2)(太原科技大学机电工程学院，太原 030024)

摘 要 STEP-NC 是一个用来实现 CAD/CAM 与 CNC 系统间数据交换的接口标准，基于 STEP-NC 的 CNC 系统是未来数控技术发展方向之一，该系统不但具有直线和圆弧插补功能，而且还具有样条曲线插补功能。为此设计了一个统一的基于 NURBS 样条曲线插补的通用插补器，并开发了一种基于等弧长的插补技术和插补算法。最后通过仿真和实例加工验证了该算法的有效性和可靠性。

关键词 STEP-NC 非均匀的有理 B 样条 样条插补 等弧长插补

中图法分类号：TP391.7 **文献标识码**：A **文章编号**：1006-8961(2007)01-0171-06

1 Introduction

Currently, most computer numerical control (CNC) machines are programmed in G & M codes (formalized by ISO6983), which is a low level language specifying mainly the cutter motion in terms of position and feed rate. Since it delivers only limited information to CNC (excluding the valuable information, such as part geometry and process plan implicated in the NC code), it makes CNC nothing but an executing mechanism completely unaware of the motions being executed, and makes it impossible to realize data exchange and sharing between CAD/CAM and CNC system[1]. Furthermore, ISO 6983 only support linear and circular interpolation and does not support the spline data, which makes it incapable of controlling five or more axis milling.

Currently, a new and comprehensive language for CNC, called STEP-NC is under development by ISO TC184 SC1. STEP-NC data model not only contains a detailed geometrical description of the part but also

基金项目：国家 863/CIMS 主题资助项目(2003AA411041)；航空科学基金项目(02H53056)

收稿日期：2005-01-07；**改回日期**：2005-10-24

第一作者简介：杜娟(1973～)，女。现为西北工业大学博士研究生，太原科技大学讲师。主要研究方向为 CAD/CAM 及其集成化制造技术。已发表学术论文 16 篇。E-mail：lgzdj1999@163.com

contains the detailed technology description, furthermore, it support spline data. Since STEP-NC data model is an extension of STEP application in numerical control machine, its geometrical description is harmonized with STEP data format. In STEP data model free curve and surface are described with uniform NURBS. Because NURBS not only can accurately express all standard quadratic parametric curses, such as line, circle and ellipse and so on, but also can express other complicated parametric curves, it is applied significance to develop a universal interpolation technology based on NURBS for intending CNC system based on STEP-NC data model.

Several researchers have developed real time parametric interpolators for curve generation using non-uniform rational B-spline (NURBS). Syh-Shiuh Yeh[2] proposed a speed-controlled interpolation algorithm with a adaptive federate. Since the chord error in interpolation depends on the curve speed and radius of the curvatures, the feed rate in the proposed algorithm is automatically adjusted so that a specified limit on the chord error is met. Wang and Yang[3] have implemented trajectory generation via cubic and quintic spline using chord length and nearly arc length parametrization. The nearly arc length parametrization resulted in smaller feedrate fluctuations compared to chord length parametrization. This was followed by the work of Wang and Wright[4], who recommended using more points in fitting the spine at high curvatures, in order to reduce the feedrate fluctuations due to the nearly arc length parametrization errors. This paper presents a NURBS interpolation technology, using a constant arc increment (Δs) at each step to maintain a stable feedrate. This way, feedrate fluctuations arising from the difference between the spline parameter (typically the chord length) and the arc length are avoided, so constant feedrate are then obtained to high speed machining system.

2 NURBS definition

NURBS is abbreviation of Non-Uniform Rational B-Spline, a NURBS curve is generally expressed as follows,

$$P(u) = \frac{\sum_{i=0}^{n} \omega_i d_i N_{i,k}(u)}{\sum_{j=0}^{n} \omega_j N_{j,k}(u)} = \sum_{j=0}^{n} d_i R_{i,k}(u) \quad (1)$$

$$R_{i,k}(u) = \frac{\omega_i N_{i,k}(u)}{\sum_{j=0}^{n} \omega_j N_{j,k}(u)} \quad u \in [0,1] \quad (2)$$

Where u is a scalar parameter which varies from 0 to 1, and $N_{i,k}(u)$ are order k basis B-spline functions; $d_i(i=0,1,\cdots,n)$ is an ordered list of deBoor points (also called control points); ω_i is an list of weigths-data. Given a knot list $U=[u_0,u_1,\cdots,u_n,\cdots,u_{n+k}]$, $k\geqslant 1$ and $n\geqslant 0$, the associated normalized B-splines $N_{i,k}(u)$ of order k in the formula (1) are defined by

$$N_{i,1}(u) = \begin{cases} 1 & u_i \leqslant u \leqslant u_{i+1} \\ 0 & \text{other} \end{cases} \quad (3)$$

$$N_{i,k}(u) = \frac{u-u_i}{u_{i+k-1}-u_i} N_{i,k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} N_{i+1,k-1}(u) \quad (4)$$

$(i=0,1,\cdots,n,k\geqslant 0)$ and assumpsit $\frac{0}{0}=0$

3 NURBS interpolation algorithm

3.1 Interpolation functions in the CNC based on STEP-NC

Most traditional CNC systems only hold linear and circular interpolation, for machining other kinds of complicated parametric curves engineers have to develop corresponding interpolation algorithms, which certainly debases interpolation efficiency of CNC system. Since NURBS not only can accurately express all standard quadratic parametric curses but also can express other complicated free curves and surface, this paper develops a universal interpolation module based on NURBS. When CNC system machines different curves, by giving proper parameters values for uniform NURBS expression, it can interpolate and machine corresponding curves and surface.

3.2 NURBS data pre-processing

Data pre-processing is a necessary precondition for interpolation computation. It mainly processes the data picked-up from STEP-NC program explainer to form NURBS expressions describing tool trajectory. In this paper we main study cubic NURBS interpolation technology.

STEP-NC data model not only contains machined part abundant technology information but only contains detailed geometrical description. Machined free curves and surfaces are described by uniform NURBS in STEP-NC program, so when the NC program is explained by STEP-NC explainer, some defining NURBS data can be automatically abstracted, including degree of function, control_point_list, knot_list, knot_multiplicities, and weights_list and so on. When these data are input into data pre-processing module of CNC, by Eq. (1) and recursion (3) and (4) NURBS expression describing tool path can be automatically generated. Because the total tool trajectory is composed of multi-spline segments, it is necessary to generate each spline segment expression. In this work we deduct a simple algorithm to create each spline segment expression as follows:

Suppose the parameter $u \in [u_i, u_{i+1}]$, based on local characteristic of cubic NURBS, in this parameter interval there be only four nonzero basis B-spline functions $N_{j,3}(u)\,(j=i,\ i-1,\ i-2,\ i-3)$, and the others are all equal to zero. Therefore, corresponding the ith segment polynomial $\boldsymbol{P}_i(u)$ can be expressed as follows,

$$\boldsymbol{P}_i(u) = \frac{\sum_{j=i-3}^{i} \omega_j \boldsymbol{d}_j N_{j,3}(u)}{\sum_{j=i-3}^{i} \omega_j N_{j,3}(u)} \quad u \in [u_i, u_{i+1}] \tag{5}$$

In Eq. (5) when ω_j, $\boldsymbol{d}_j$, u_i are given certain values $\boldsymbol{P}_i(u)$ is only decided by basis B-spline functions $N_{j,3}(u)$. Based on recursion Eq. (3) and (4) $N_{j,3}(u)$ can be deducted as shown in the Eq. (6).

$$N_{j,3}(u) = \begin{cases} \dfrac{(u-u_j)^3}{(u_{j+3}-u_j)(u_{j+2}-u_j)(u_{j+1}-u_j)} & u \in [u_j, u_{j+1}) \quad (6\text{-}1) \\[2ex] \dfrac{(u-u_j)^2(u_{j+2}-u)}{(u_{j+2}-u_j)(u_{j+3}-u_j)(u_{j+2}-u_{j+1})} + \dfrac{(u_{j+3}-u)(u-u_j)(u-u_{j+1})}{(u_{j+3}-u_j)(u_{j+3}-u_{j+1})(u_{j+2}-u_{j+1})} + \\ \quad \dfrac{(u-u_{j+1})^2(u_{j+4}-u)}{(u_{j+4}-u_{j+1})(u_{j+3}-u_{j+1})(u_{j+2}-u_{j+1})} & u \in [u_{j+1}, u_{j+2}) \quad (6\text{-}2) \\[2ex] \dfrac{(u-u_j)(u_{j+3}-u)^2}{(u_{j+3}-u_j)(u_{j+3}-u_{j+1})(u_{j+3}-u_{j+2})} + \dfrac{(u_{j+3}-u)(u_{j+4}-u)(u-u_{j+1})}{(u_{j+3}-u_{j+1})(u_{j+4}-u_{j+1})(u_{j+3}-u_{j+2})} + \\ \quad \dfrac{(u-u_{j+2})(u_{j+4}-u)^2}{(u_{j+4}-u_{j+1})(u_{j+4}-u_{j+2})(u_{j+3}-u_{j+2})} & u \in [u_{j+2}, u_{j+3}) \quad (6\text{-}3) \\[2ex] \dfrac{(u_{j+4}-u)^3}{(u_{j+4}-u_{j+1})(u_{j+4}-u_{j+2})(u_{j+4}-u_{j+3})} & u \in [u_{j+3}, u_{j+4}) \quad (6\text{-}4) \\[2ex] 0 & u \notin [u_j, u_{j+4}) \quad (6\text{-}5) \end{cases} \tag{6}$$

By substituting $Q_1(u)$, $Q_2(u)$, $Q_3(u)$ and $Q_4(u)$ for sub-Eq. (6-1), (6-2), (6-3) and (6-4) respectively, then the ith segment polynomial $\boldsymbol{P}_i(u)$ can be written as follows,

$$\boldsymbol{P}_i(u) = \frac{\omega_{i-3}\boldsymbol{d}_{i-3}Q_4(u)}{\omega_{i-3}Q_4(u)} + \frac{\omega_{i-2}\boldsymbol{d}_{i-2}Q_3(u)}{\omega_{i-2}Q_3(u)} + \frac{\omega_{i-1}\boldsymbol{d}_{i-1}Q_2(u)}{\omega_{i-1}Q_2(u)} + \frac{\omega_i\boldsymbol{d}_iQ_1(u)}{\omega_iQ_1(u)} \tag{7}$$

So by Eq. (7) we can educe each segment polynomial expression of NURBS.

Notice: when $Q_4(u)$, $Q_3(u)$ and $Q_2(u)$ are being computed in Eq. (7) the i should be replaced by $i-3$, $i-2$ and $i-1$ respectively.

3.3 Constant arc increment interpolation

In CNC machine tool, the task of interpolation is to compute a sequence of reference points that constitute the desired tool path, and tool moves along those points to approach theory profile. The NURBS interpolation is to calculate parameter value u for each interpolation point during machining.

By Eq. (1) we can see each point vector function can be computed by parameter u, so it is conceivable to compute interpolation point through parameter u, that is to say, using constant increment Δu to calculate next interpolation point of NURBS during pre interpolation period. Due to NURBS function is not linear with parameter u, the chord length between two consecutive interpolation points is different. When the tool passes different chord length in the same interpolation period, it is inevitable to result in feedrate fluctuating, which in return affects machining precision of part. On the other hand, it is difficult to optimize Δu value. If the Δu is given too small value there will be a mass of interpolation points, which may affect machining efficiency, while as a large value can result in obvious machining error. To avoid feedrate fluctuation we develop an interpolation algorithm based on constant arc length increment, which could satisfy invariable machining speed.

3.3.1 Estimation of arc length

Ideally, the total of length the composite NURBS is calculated by numerically integrating the arc length for each segment, and summing the segment length as,

$$L = \sum_{i=1}^{N} s_i = \sum_{i=1}^{N} \int_0^{s_i} \mathrm{d}s \tag{8}$$

Where, L is the total composite NURBS length and s_i is the ith NURBS segment. Since it is difficult to calculate prototype function of integral in Eq. (8), so in this work L is evaluated by the chord length. The evaluation error mainly depends on discrete method, so this paper employs distance that is the product of feedrate and interpolation period to break up NURBS. Detailing is as follows,

(1) Discretization of the NURBS

Normally, knot_multiplicities is given 4 for the cubic NURBS curve, so the parameter u is defined between $u = u_3$ and $u = u_{n+1}$ and the value i in Eq. (7) belongs to $i \geqslant 3$ and $i \leqslant n$. For a two dimensional case, $\boldsymbol{P}_i(u)$ and $\boldsymbol{d}_j$ can be written as,

$$\boldsymbol{P}_i(u) = \begin{bmatrix} P_{x,i}(u) \\ P_{y,i}(u) \end{bmatrix}, \quad \boldsymbol{d}_j = \begin{bmatrix} d_{x,j} \\ d_{y,j} \end{bmatrix}$$

Suppose F is the planned tool feedrate and T is the interpolation period, and M_i is count of discrete points of the ith segment NURBS, then the chord length l_i between two consecutive points $P_i(u_i)$ and $P_i(u_{i+1})$ and M_i can be expressed as, respectively,

$$l_i = \sqrt{(P_{x,i}(u_{i+1}) - P_{x,i}(u_i))^2 + (P_{y,i}(u_{i+1}) - P_{y,i}(u_i))^2} \quad (i = 3,4,\cdots,n) \tag{9}$$

$$M_i = \mathrm{round}\left(\frac{l_i}{F \cdot T}\right) \quad (i = 3,4,\cdots,n) \tag{10}$$

(2) Calculation of interpolation points

Based on above l_i and M_i, the chord length increment Δl_i for the ith segment NURBS becomes as,

$$\Delta l_i = \frac{l_i}{M_i} \quad (i = 3,4,\cdots,n) \tag{11}$$

Considering $k = 1,2,\cdots,M_i$, to be the chord increment counter, the corresponding point $(x_{i,k}, y_{i,k})$ on the ith segment NURBS is computed as,

$$\begin{aligned} x_{i,k} &= \frac{\sum_{j=i-3}^{i} \omega_j d_{x,j} N_{j,3}(k \cdot \Delta l_i)}{\sum_{j=i-3}^{i} \omega_j N_{j,3}(k \cdot \Delta l_i)} \\ y_{i,k} &= \frac{\sum_{j=i-3}^{i} \omega_j d_{y,j} N_{j,3}(k \cdot \Delta l_i)}{\sum_{j=i-3}^{i} \omega_j N_{j,3}(k \cdot \Delta l_i)} \end{aligned} \tag{12}$$

(3) Calculation of the total arc length

By the Eq. (12) the arc length between two successive points can be expressed as,

$$s_{i,k} \cong \sqrt{(x_{i,k} - x_{i,k-1})^2 + (y_{i,k} - y_{i,k-1})^2} \tag{13}$$
$$(i = 3,4,\cdots,n;\ k = 1,2,\cdots,M_i)$$

Hence, the total arc length of NURBS can be evaluated as,

$$L = \sum_{j=1}^{N} \int_0^{s_i} \mathrm{d}s \cong \sum_{i=3}^{n} \sum_{k=1}^{M_i} s_{i,k} \tag{14}$$

3.3.2 Interpolation based on constant arc increment

To avoid feedrate fluctuation, in this work we propose a interpolation algorithm based on the constant arc increment Δs. The path increment Δs is determined in such a way that, supposing F is the highest value of tool feedrate and T is the interpolation period, and by the above calculated the total arc length L, then the total number of interpolation steps M and resulting in a path increment Δs at each interpolation step are calculated as follows[5],

$$M = \mathrm{round}\left(\frac{L}{F \cdot T}\right) \tag{15}$$

$$\Delta s = \frac{L}{M} \tag{16}$$

Once Δs is worked out, by which the parameter u corresponding to next interpolation point can be figured out. consequently, the x axis increment Δx and the y axis increment Δy are all can be computed to servo sub-system. Noting that, it is necessary to determine which segment NURBS the current interpolation point locates before computing parameter u of next interpolation point. If B_i is the number of interpolation steps of the ith segment NURBS, and ΔL_i is the arc length of the ith segment, then the following equation holds,

$$B_i = \frac{\Delta L_i}{\Delta s} = \frac{\sum_{k=1}^{M_i} s_{i,k}}{\Delta s} \quad (i = 3, 4, \cdots, n) \tag{17}$$

When the interpolation point is at the ith segment NURBS, the parameter u can be calculated by the following equations,

$$\Delta s = \sqrt{\Delta x^2 + \Delta y^2} \tag{18}$$

$$\Delta x = x_{i,k+1} - x_{i,k} = \frac{\sum_{j=i-3}^{i} \omega_j d_{x,j} N_{j,3}(u)}{\sum_{j=i-3}^{i} \omega_j N_{j,3}(u)} - x_{i,k}$$
$$\Delta y = y_{i,k+1} - y_{i,k} = \frac{\sum_{j=i-3}^{i} \omega_j d_{y,j} N_{j,3}(u)}{\sum_{j=i-3}^{i} \omega_j N_{j,3}(u)} - y_{i,k} \tag{19}$$

Since the NURBS expression and the previous position $(x_{i,k}, y_{i,k})$ are known, the problem is to find the new chord parameter u, so that the fixed arc increment Δs is realized. Combining Eqs. (18) and (19), the new chord parameter can be obtained by solving the root of the below sixth-order polynomial:

$$F(u) = a_6 u^6 + a_5 u^5 + a_4 u^4 + a_3 u^3 + a_2 u^2 + a_1 u + a_0 \tag{20}$$

The solution of Eq. (20) is obtained through Newton iterative method. In the Eq. (20) coefficients $a_6, a_5, a_4, a_3, a_2, a_1$ and a_0 are all real numbers, and basing on the above interpolation principle a unique real number root must be existence in the Eq. (20). Suppose the parameter $u_{i,k}$ corresponds to the point $(x_{i,k}, y_{i,k})$ and the parameter $u_{i,k-1}$ corresponds to the point $(x_{i,k-1}, y_{i,k-1})$, then in order to reduce time of iterative calculating the initial proximate value u_0 is defined as follows: $u_0 = u_{i,k} + (u_{i,k} - u_{i,k-1})$, where $u_0 \in (u_{i,k}, 1)$, and the iterative error $\varepsilon = |u_{i+1} - u_i| < 10^{-3}$.

Although $F(u)$ has a relatively high order, the experiment proved the method have properties of rapid convergence and calculation stability, making this approach feasible for real-time implementation. Furthermore, before interpolation the main of computation has been finished, and it is only to calculate Eq. (19) and (20) during real-time interpolation, which can fully meet the needs of interpolation speed.

4 Analysis of interpolation error

From the above interpolation technology we can see that the interpolation points are all along at tool path during the machining, hence, position error is zero. The interpolation error is mainly caused by approximate calculation of arc length. In this paper we use an effective method to discrete the NURBS, and by this method the density of interpolation point is considerable dense, making the desired machining precision of part can be absolutely realized.

5 Simulation results

In order to validate the correctness of the proposed method, an algorithm instance programmed by Visual C++ and Open GL was demonstrated through simulation. The attribute data of NURBS surface instance and corresponding manufacturing data were firstly extracted from the new NC program by STEP-NC explainer. Their values are as following:

the 3-D control points[4][4][3]
= {{-1.0f, -1.0f, 0.0f}, {-0.5f, 1.0f, 0.0f}, {1.0f, -1.0f, 0.0f}, {1.5f, 1.0f, 0.0f}}, {{-1.0f, -1.0f, -1.0f}, {-0.5f, 1.0f, -1.0f}, {1.0f, -1.0f, -1.0f}, {1.5f, 1.0f, -1.0f}}, {{-1.0f, -1.0f, -2.0f}, {-0.5f, 1.0f, -2.0f}, {1.0f, -1.0f, -2.0f}, {1.5f, 1.0f, -2.0f}}, {{-1.0f, -1.0f,

$-3.0f\}$, $\{-0.5f, 1.0f, -3.0f\}$, $\{1.0f, -1.0f, -3.0f\}$, $\{1.5f, 1.0f, -3.0f\}\}$,

weights $\omega_{i,j}$ =

$\{\{1,1,1,1\},\{1,1,1,1\},\{1,1,1,1\},\{1,1,1,1\}\}$,

knot list $\boldsymbol{U}[8] = \boldsymbol{V}[8] = \{0,0,0,0,1,1,1,1\}$,

the orders $k = l = 3$, the knot multiplicities is 4.

The NURBS surfaces machining trajectory formed by employing the above interpolation method is shown in Fig. 1.

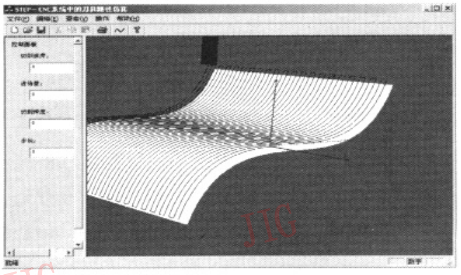

Fig. 1 The tool path of the NURBS

6 Conclusions

This paper has presented an interpolation method based on constant arc length increment and developed a data pre-processing module. By an instance test the interpolation algorithm has been proved to avoid feedrate fluctuating during machining, consequently, higher machining precision and quality of surface can be obtained. Furthermore, in the fact CNC machining, it is necessary to take into account tool compensation, and acceleration and deceleration control at the start and the end of machining. These issues are to be discussed in the other papers.

References

1 Suh S H, Cheon S U. A framework for intelligent CNC and data model[J]. International Journal Advance Manufacturing Technology, 2002, **19**:727 ~ 35.

2 Yeh S S, Hsu P L. Adaptive-feedrate interpolation for parametric curves with a confined chord error[J]. Computer-Aided Design, 2002, **34**(4): 229 ~ 237.

3 Wang F C, Yang D C H. Nearly arc-length parameterized quintic-spline interpolation for precision machining[J]. Computer-Aided Design, 1993, **25**(5): 281 ~ 288.

4 Wang F C, Wright P K. Open architecture controllers for machine tools Part 2: A real time quintic spline interpolator[J]. Journal of Manufacturing Science and Engineering, 1998, **120**(2): 425 ~ 432.

5 Kaan E, Yusuf A. High speed CNC system design Part I: jerk limited trajectory generation and quintic spline interpolation[J]. Machine Tools & Manufacture, 2001, **41**(9): 1323 ~ 1345.