

# Approximation Method of Tool Path Oriented to Five-axis CNC Machining with NURBS Curve

Chen Liangji, Li Huiying

**Abstract**—For the case of the advancement of Non-Uniform Rational B-Spline (NURBS) curve to represent a complex freeform shape, the NURBS method has been applied in Computer Numerical Control (CNC) machine tools to express the tool path curve. In order to achieve higher approximation accuracy in five-axis CNC machining, the approximation principle was firstly proposed based on the time division and the tool path curve was approximated by a group of chord lines. To avoid the federate fluctuation, the federate correction method is also proposed. The relation methods for realizing the approximation of tool path curve was also presented in this paper. The given example and the cutting result show that the proposed methods are feasibility of being applied to CNC trajectory approximation to achieve high-speed and high-accuracy multi-axis machining.

**Index Terms**—Five-axis machining; CNC; NURBS curve; Tool path

## I. INTRODUCTION

Five-axis computer numerical control (CNC) machine tools have been more and more applied in performing a cutting task of complex work-pieces such as molds, dies, etc. to meet the requirements of high speed and high accuracy. To realize the multi-axis CNC machining, the work-pieces have to be firstly designed using computer-aided design (CAD) system, and then cutter contacting path is converted to the cutter location (CL) path and tool path by computer-aided manufacturing (CAM) system [1~4].

The controlling algorithm of CNC system is a type of location and federate controlled method. The most important and functional kernel module is tool path approximation method, which is also the so-called interpolation method in applied mathematics. In order to guide the CNC machine to perform cutting along the CL path or tool path, the path is traditionally divided into a set of line or circular segments (also called NC codes). These segments will approximate the original tool path to a desired accuracy if sufficient numbers of segments are used. To obtain more approximation accurate and to reduce the contour error, the number of the segments should be increased. However, these methods could lead to several drawbacks [5~9].

Based on the above observation and analysis, it is evident that accurate high-speed manufacturing is hard to achieve if the conventional line and circular interpolators are used in CNC systems. Therefore, developing a new type of interpolator for CNC machine tools is of importance [10]. Since non-uniform rational B-spline (NURBS) uses fewer data to represent the designed curves and if both the

CAD/CAM and CNC systems adopt NURBS as the geometric representation, a considerable amount of time spent on transferring data between CAD/CAM and CNC machine can be saved.

Because of the advantages of NURBS method [11], a tool path approximation method is proposed in this paper for machining a complex curves (or surfaces). And the relation algorithm of NURBS method is also introduced and presented to help realizing the controlling software of the multi-axis CNC system. Consequently, a trial cutting is performed in our research work to validate that the methods in this paper is succeed and applicable and the approximation theory can dramatically increase the machining accuracy and decrease the machining time.

## II. NURBS REPRESENTATION

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

$$\mathbf{r}(u) = x(u)\mathbf{i} + y(u)\mathbf{j} + z(u)\mathbf{k} \quad (0 \leq u \leq 1) \quad (1)$$

where  $u$  is an arbitrary parameter,  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  is the unit vector of axis orientation of  $x, y, z$  respectively.

A  $p$ -degree NURBS curve with parameter  $u$  can be defined as follows:

$$\mathbf{r}(u) = [x(u) \ y(u) \ z(u)]^T = \frac{\sum_{i=0}^n w_i B_{i,p}(u) \mathbf{C}_i}{\sum_{i=0}^n w_i B_{i,p}(u)} \quad (0 \leq u \leq 1) \quad (2)$$

where  $\mathbf{C}_i$  is the  $i^{\text{th}}$  3-D control point;  $w_i$  is the corresponding weight factor of  $\mathbf{C}_i$ ;  $(n+1)$  is the number of control points;  $B_{i,p}(u)$ , B-spline basis function with degree of  $p$ , can be calculated by the following formula:

$$\begin{cases} B_{i,0} = \begin{cases} 1 & (u_i \leq u < u_{i+1}) \\ 0 & (\text{otherwise}) \end{cases} \\ B_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} B_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}(u) \\ \begin{matrix} 0 \\ 0 \end{matrix} = 0 \quad (\text{prescribed}) \end{cases} \quad (3)$$

where  $u_i (i=0,1,2,\dots,n+p+1)$  is corresponding parametric value of control point which can be formed a non-decrease serial  $\mathbf{K}$ .  $\mathbf{K}$  is also called the knot vector and can be expressed as the following style.

$$\mathbf{K} = \{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_n, \underbrace{1, \dots, 1}_{p+1} \}$$

Chen Liangji, School of Mechanical Engineering, Tianjin Polytechnic University, Tianjin, China

Li Huiying, School of Mechanical Engineering, Tianjin Polytechnic University, Tianjin, China

III. APPROXIMATION METHOD OF A NURBS CURVE

As shown in Fig.1, if the tool path curve  $r(u)$  is to be approximated, the approximation point should be interpolated onto the curve according to certain rules. After the above process, the distance between chord line formed by the adjacent two approximation points and original curve is the approximation error. In this article, we use the following time-division rule to generate the approximation points. And now, the approximation process of the method is described as follows.

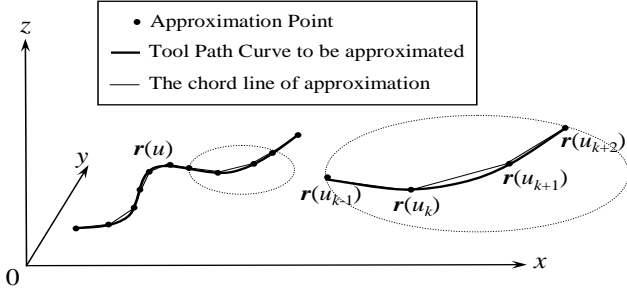


Fig.1 Approximation principle of tool path curve

The key to approximate a parametric tool path curve represented by NURBS in real time (namely command generation) is that the segmentation should be based on equal increments of sampling period  $T_s$ , rather than equal increments of the parameter  $u$ . Accordingly, a method is needed to determine successive values of parameter  $u$  (at each sampling period) such that the ratio between  $\Delta s$  (increment of curve length during each sampling period) and  $T$  is equal to a desired profile  $V(t)$  in a federate control problem. The procedures for determining successive values of  $u$  are summarized as follows. Considering the federate  $V(t)$  along the curve,  $r(u)$  in Eq.(2),  $V(t)$  can be expressed as

$$V(t) = \left\| \frac{dr(u)}{dt} \right\| = \left\| \frac{dr(u)}{du} \right\| \left( \frac{du}{dt} \right) \quad (4)$$

hence

$$\frac{du}{dt} = \frac{V(t)}{\left\| \frac{dr(u)}{du} \right\|} \quad (5)$$

Using 1st order Taylor expansion method, Eq.(5) can be processed as the following equation

$$u_{k+1} = u_k + \frac{V(t_k) \cdot T_s}{\left\| \frac{dr(u)}{du} \right\|_{u=u_k}} \quad (6)$$

where  $u_k$  and  $u_{k+1}$  denote the value of  $u$  at  $t=t_k=kT_s$  and  $t=t_{k+1}=(k+1)T_s$ , respectively.

And also, the 2nd order approximate algorithm is

$$u_{k+1} = u_k + \frac{V(t_k) \cdot T_s}{\left\| \frac{dr(u)}{du} \right\|_{u=u_k}} - \frac{\left\| V^2(t_k) \cdot T_s^2 \left( \frac{dr(u)}{du} \cdot \frac{d^2r(u)}{du^2} \right) \right\|_{u=u_k}}{2 \cdot \left\| \frac{dr(u)}{du} \right\|_{u=u_k}^4} \quad (7)$$

The 1st and 2nd derivative of  $r(u)$  with  $u$  can be respectively obtained as

$$\frac{dr(u)}{du} = \frac{\sum_{i=0}^n w_i B'_{i,p}(u) C_i}{\sum_{i=0}^n w_i B_{i,p}(u)} - \frac{\sum_{i=0}^n w_i B'_{i,p}(u) \sum_{i=0}^n w_i B_{i,p}(u) C_i}{\left( \sum_{i=0}^n w_i B_{i,p}(u) \right)^2} \quad (8)$$

$$\frac{d^2r(u)}{du^2} = \frac{\sum_{i=1}^n w_i B_{i,p}(u) \sum_{i=0}^n w_i B''_{i,p}(u) C_i}{\left( \sum_{i=0}^n w_i B_{i,p}(u) \right)^2} - \frac{\sum_{i=0}^n w_i B''_{i,p}(u) \sum_{i=0}^n w_i B_{i,p}(u) C_i}{\left( \sum_{i=0}^n w_i B_{i,p}(u) \right)^2}$$

$$- \frac{2 \sum_{i=0}^n w_i B'_{i,p}(u) \sum_{i=0}^n w_i B_{i,p}(u) \sum_{i=0}^n w_i B'_{i,p}(u) C_i}{\left( \sum_{i=0}^n w_i B_{i,p}(u) \right)^3} - \frac{2 \sum_{i=0}^n w_i B'_{i,p}(u) \sum_{i=0}^n w_i B_{i,p}(u) C_i \sum_{i=0}^n w_i B'_{i,p}(u)}{\left( \sum_{i=0}^n w_i B_{i,p}(u) \right)^3} \quad (9)$$

where the general algorithm for  $m^{\text{th}}$  order derivative of  $B_{i,p}(u)$  is

$$B_{i,p}^{(m)}(u) = p \left[ \frac{B_{i,p-1}^{m-1}(u)}{u_{i+p-1} - u_i} - \frac{B_{i+1,p-1}^{m-1}(u)}{u_{i+p} - u_{i+1}} \right] \quad (10)$$

and

$$\frac{dr(u)}{du} = \sqrt{(x')^2 + (y')^2 + (z')^2} \quad (11)$$

where

$$x = x(u), y = y(u), z = z(u),$$

$$x' = \frac{dx(u)}{du}, y' = \frac{dy(u)}{du}, z' = \frac{dz(u)}{du} \quad (12)$$

Using Eqs. (8), (9), (11) and (12), then  $x', y', z', x'', y''$  and  $z''$  can be calculated and using Eqs.(5), (6) and (8),  $u_{k+1}$  can be acquired. This is called the first order approximation algorithm of NURBS curve. Similarly, the second order approximation algorithm also can be presented with Eqs.(5), (7), (9) and (11). Thus parameter  $u$  is updated in each time period to cope with federate  $V(u_k)$ .  $V(u_k)$  can be obtained with the following acceleration and deceleration (Acc/Dec) method during the course of arbitrary one time-division point.

Assumed that the acceleration and deceleration is  $a$  and destination federate is  $v$  as shown in Fig.2.

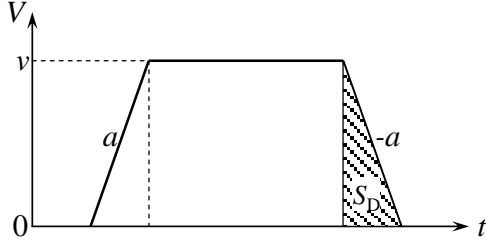


Fig.2 Mathematical model of linear acceleration and deceleration

During the process of accelerating, the following method can be used

$$V(u_k) = at_k$$

While the decelerating course, the key problem to be solved is evaluation of deceleration point. In Fig.2,  $S_D$ , distance of deceleration, can be calculated as follows

$$S_D = \frac{v^2}{2a} \quad (13)$$

The distance of deceleration, which is also the length between deceleration point and destination point along the spline curve, can be calculated as follows

$$\int_{u_D}^1 \left\| \frac{dr(u)}{du} \right\| du = S_D \quad (14)$$

where  $u_D$  is corresponding parameter of the deceleration point. The following Newton-Rapson method can be used to calculate the value of  $u_D$ .

If

$$f(u) = \int_u^1 \left\| \frac{dr(u)}{du} \right\| du - S_D \quad (15)$$

we have

$$f(u_D) = 0$$

Assumed that initial value of  $u_D$  is  $u_0$  and the initial value is 1.0, that is,  $f(u_0) = -S_D$ , then we have the following equation

$$u_{i+1} = u_i - f(u_i) / f'(u_i) \quad i \geq 0 \quad (16)$$

where  $i$  is set as an integer. Eq.(15) can be equivalent to the following formula

$$u_{i+1} = u_i + \frac{f(u_i)}{\left\| \frac{dr(u_i)}{du} \right\|} \quad i \geq 0 \quad (17)$$

$f(u_i)$  in Eqs.(15) and (16) can be obtained using Simpson integral method.

If we assume that we had the following equation

$$u(t_k) = \frac{V(t_k)}{\left\| \frac{dr(u)}{du} \right\|_{u=u_k}} \quad (18)$$

then Eq.(7) can be rewritten as follows according to Eq.(18)

$$u_{k+1} = u_k + T_s u'(t_k) + T_s^2 u''(t_k) / 2 \quad (19)$$

The following back ward difference approximation method can be employed here to compute  $u'(t_k)$  and  $u''(t_k)$ , which is

$$u'(t_k) = (u_k - u_{k-1}) / T_s \quad (20)$$

and

$$u''(t_k) = (u_k - 2u_{k-1} + u_{k-2}) / T_s^2 \quad (21)$$

Substituting Eqs.(20) and (21) into Eq.(19) will yield an initial value of  $u_{k+1}$ .

$$\bar{u}_{k+1} = 2.5u_k - 2u_{k-1} + 0.5u_{k-2} \quad (22)$$

However, the federate  $\bar{V}_k$  resulting from  $\bar{u}_{k+1}$  can be calculated as follows

$$\bar{V}_k = \left\| r(\bar{u}_{k+1}) - r(u_k) \right\| / T_s$$

In fact,  $\bar{V}_k$  may not be accurate. Let  $V_k^*$  be the desired federate, so the error of federate should be equal to  $|V_k^* - \bar{V}_k|$ .

Assumed that

$$\tilde{u}_{k+1} = b_1 \bar{u}_{k+1} + (1-b_1)u_k \quad (23)$$

where

$$b_1 = V_k^* / [b_2 \bar{V}_k + (1-b_2)V_k^*], b_2 \in (0, 1]$$

If federate error  $|V_k^* - \bar{V}_k|$  is enough small, such that the error is smaller than the given tolerance  $\varepsilon$ , then the iterative calculation described in Eq.(23) will be stopped and let

$$u_{k+1} = \tilde{u}_{k+1} \quad (24)$$

#### IV. RELATION ALGORITHMS FOR THE APPROXIMATION THEORY

The above approximation calculation method has some important algorithms such as arbitrary order derivative of NURBS curve  $r(u)$  and total length of  $r(u)$  between arbitrary two parameters. In the rest of this article, we have concentrated our efforts for getting the calculation results.

##### A. Arbitrary order derivative of $r(u)$

According to the Eq.(1), the following first order derivative can be obtained

According to the Eq.(1), the following first order derivative can be obtained

$$r'(u) = \left( \frac{A(u)}{w(u)} \right)' = \frac{A'(u)w(u) - A(u)w'(u)}{[w(u)]^2} = \frac{A'(u) - r(u)w'(u)}{w(u)} \quad (25)$$

where

$$A(u) = \sum_{i=0}^n w_i B_{i,p}(u) C_i$$

and

$$w(u) = \sum_{i=0}^n w_i B_{i,p}(u)$$

For the  $r^{\text{th}}$  order ( $r \in [1, p]$ ) derivative of  $r(u)$ , the following result can be deduced according to the Leibnitz method

$$\begin{aligned} A^{(r)}(u) &= (w(u)r(u))^{(r)} = \sum_{i=0}^r \binom{r}{i} w^{(i)}(u) r^{(r-i)}(u) \\ &= w(u)r^{(r)}(u) + \sum_{i=1}^r \binom{r}{i} w^{(i)}(u) r^{(r-i)}(u) \end{aligned} \quad (26)$$

where the operator  $\binom{r}{i}$  can be expressed as

$$\binom{r}{i} = \frac{r!}{i!(r-i)!}$$

The  $r^{\text{th}}$  order derivative of  $r(u)$  can be acquired from Eq.(8) as

$$r^{(r)}(u) = \frac{A^{(r)}(u) - \sum_{i=1}^r \binom{r}{i} w^{(i)}(u) r^{(r-i)}(u)}{w(u)} \quad (27)$$

where

$$A^{(r)}(u) = \sum_{i=0}^n w_i B_{i,p}^{(r)}(u) C_i$$

and

$$w^{(r)}(u) = \sum_{i=0}^n w_i B_{i,p}^{(r)}(u)$$

According to Eq.(9), we can find that the calculation of  $r^{\text{th}}$  order derivative of  $B_{i,p}(u)$ .

The 1st derivative of B spline basic function with p degree is

$$B_{i,p}^{(1)}(u) = p \left( \frac{B_{i,p-1}^{(0)}(u)}{u_{i+p} - u_i} - \frac{B_{i+1,p-1}^{(0)}(u)}{u_{i+p+1} - u_{i+1}} \right) \quad (28)$$

where  $B_{i,p}^{(0)}(u)$  is representing itself of basic function  $B_{i,p}(u)$ . So we have the following iterative equation of  $r^{\text{th}}$  order derivative of  $B_{i,p}(u)$

$$B_{i,p}^{(r)}(u) = p \left( \frac{B_{i,p-1}^{(r-1)}(u)}{u_{i+p} - u_i} - \frac{B_{i+1,p-1}^{(r-1)}(u)}{u_{i+p+1} - u_{i+1}} \right) \quad (29)$$

Till now, we have the following expression for the derivative of basic function

$$B_{i,p}^{(r)}(u) = \begin{cases} B_{i,0}^{(0)}(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}) \\ 0 & \text{other} \end{cases} & r = 0 \\ \frac{u - u_i}{u_{i+p} - u_i} B_{i,p-1}^{(0)}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}^{(0)}(u) & 1 \leq r \leq p \\ p \left( \frac{B_{i,p-1}^{(r-1)}(u)}{u_{i+p} - u_i} - \frac{B_{i+1,p-1}^{(r-1)}(u)}{u_{i+p+1} - u_{i+1}} \right) & 1 \leq r \leq p \end{cases} \quad (30)$$

For the solution of Eq.(30), the follow iterative method is proposed and applied in this paper.

Assumed that matrix of  $\overset{(r)}{M}(u, i, p)$  and

$\overset{(r)}{N}(u, i, p)$  is

$$\overset{(r)}{M}(u, i, p) = \begin{cases} \begin{bmatrix} \frac{u - u_i}{u_{i+p} - u_i} & \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} \end{bmatrix} & r = 0 \\ \begin{bmatrix} p & -p \end{bmatrix} & r \neq 0 \end{cases} \quad (31)$$

and

$$\overset{(r)}{N}(u, i, p) = \begin{bmatrix} B_{i,p}^{(r)}(u) \\ B_{i+1,p}^{(r)}(u) \end{bmatrix} \quad (32)$$

then we have the following equation according to Eq.(30)

$$\overset{(r)}{B}_{i,p}^{(r)}(u) = \begin{cases} \overset{(r)}{M}(u, i, p) \overset{(r)}{N}(u, i, p-1) & r = 0 \\ \overset{(r)}{M}(u, i, p) \overset{(r-1)}{N}(u, i, p-1) & r \neq 0 \end{cases} \quad (33)$$

According to Eq.(33), the following block matrix method can be deduced and the method also can be validated with mathematical induction.

$$\overset{(r)}{B}_{i,p}^{(r)}(u) = \overset{(r)}{M}(u, i, p) \times \begin{bmatrix} \overset{(r-1)}{M}(u, i, p-1) & \mathbf{0} \\ \mathbf{0} & \overset{(r-1)}{M}(u, i+1, p-1) \end{bmatrix} \times \begin{bmatrix} \overset{(r-2)}{M}(u, i, p-2) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \overset{(r-2)}{M}(u, i+1, p-2) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \overset{(r-2)}{M}(u, i+1, p-2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{(r-2)}{M}(u, i+2, p-2) \end{bmatrix} \times \dots \times \begin{bmatrix} \overset{(0)}{M}(u, i, p-r) & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \overset{(0)}{M}(u, i+1, p-r) & \ddots & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \ddots & \ddots & \overset{(0)}{M}(u, i+r-1, p-r) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \overset{(0)}{M}(u, i+r, p-r) \end{bmatrix} \times \dots \times \begin{bmatrix} \overset{(0)}{M}(u, i, 1) & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \overset{(0)}{M}(u, i+1, 1) & & & \\ \vdots & \vdots & \overset{(0)}{M}(u, i+1, 1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \overset{(0)}{M}(u, i+p-2, 1) & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \overset{(0)}{M}(u, i+p-2, 1) \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \overset{(0)}{M}(u, i+p-1, 1) \end{bmatrix} \times \begin{bmatrix} \overset{(0)}{N}(u, i, 0) \\ \overset{(0)}{N}(u, i+1, 0) \\ \overset{(0)}{N}(u, i+1, 0) \\ \vdots \\ \overset{(0)}{N}(u, i+p-2, 0) \\ \overset{(0)}{N}(u, i+p-2, 0) \\ \overset{(0)}{N}(u, i+p-1, 0) \end{bmatrix} \quad (34)$$

where the all matrix about  $\overset{(r)}{M}$  in Eq.(34) is block square matrix.

### B. Calculation of approximated tool path curve

During the course of acceleration and deceleration controlling method mentioned above, there are two types of computational problems: 1) total length of  $r(u)$  between arbitrary two parameters(i.e.  $a$  and  $b$ ); 2) one of parameter

when curve length is given. Here we use the following numerical analysis method to solve the above difficulties.

$L(a,b)$ , the total length from a point  $a$  to another point  $b$  along a curve  $r(u)$ , can be expressed as follows

$$\begin{aligned} L(a,b) &= \int_a^b \|r'(u)\| du \\ &= \int_a^b \sqrt{[x'(u)]^2 + [y'(u)]^2 + [z'(u)]^2} du \\ &= \int_a^b f(u) du \end{aligned} \quad (35)$$

If  $h = \frac{b-a}{l}$ ,  $u_k = a + kh, k = 0, \dots, l; l = 2m$ , then we

have

$$\begin{aligned} L(a,b) &= S_l(f(u)) \\ &= \frac{h}{3} \left( f(a) + 4 \sum_{k=0}^{m-1} f(u_{2k+1}) + 2 \sum_{k=1}^{m-1} f(u_{2k}) + f(b) \right) \end{aligned} \quad (36)$$

Assumed that the total length ( $L$ ) between the parameter  $u_d$  and the parameter 1 is given, the destination is to calculate the value of  $u_d$ . that is

$$\int_{u_d}^1 \|r'(u)\| du = \int_{u_d}^1 f(u) du = L \quad (37)$$

Let

$$g(u) = \int_u^1 \|r'(v)\| dv - L = \int_u^1 f(v) dv - L \quad (38)$$

and the initial value of  $u_d$  is  $u_0 (u_0=1)$ , then we have

$$g(u_0) = -L$$

Meanwhile, there is following iterative expression as Eq.(39)

$$u_{i+1} = u_i - g(u_i) / g'(u_i) \quad i \geq 0 \quad (39)$$

Eq.(39) is equal to Eq.(40)

$$u_{i+1} = u_i + \frac{g(u_i)}{f(u_i)} \quad i \geq 0 \quad (40)$$

where  $g(u_i)$  can be easily solved according to the method of Eq.(36). The condition of terminating the iterative calculation is

$$\frac{|u_{i+1} - u_i|}{u_i} = \frac{|g(u_i)|}{u_i f(u_i)} < \eta \quad u_i \neq 0 \quad (41)$$

where  $\eta$  is the allowable error of iterative calculation.

Fig.3 shows the tool path curve to be approximated, which is a quarter of circle with radius of 10 mm represented in 3<sup>rd</sup> order NURBS form, and its control points are  $P_0(10,0,0)$ ,  $P_1(10,10,0)$  and  $P_2(0,10,0)$  with weights 1, 0.707 and 1 respectively.

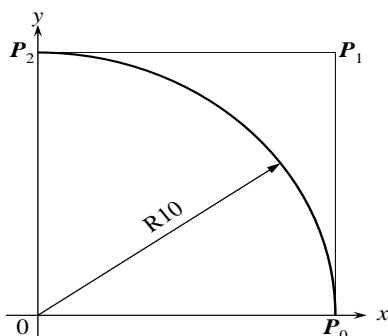


Fig.3 The example NURBS curve with knot vector [0,0,0,1,1,1]

Table 1 has listed the contrast of curve length between computational result using proposed method and theory value when different conditions demonstrated in Eq.(36) happened. From the Table 1 we can conclude that the more divisions, the higher approximation accuracy.

Table 1. Tool path curve length between two points with contrast to the theory values (mm)

	$a=0, b=0.5$	$a=0, b=1$
$l=2$	7.854921	15.75974
$l=4$	7.854038	15.70984
$l=8$	7.853985	15.70808
$l=16$	7.853982	15.70797
theory values	7.8539816	15.707963

Table 2 has described that the approximation accuracy when different iterative allowable errors are given. From the Table 2, we can find that with the increase of the accuracy requirement, the computational result is by and by approximating the real parameter value of the given point. When the allowable error  $\eta$  is set as 0.01, the number of iterative calculation is only three enough to skip the iterative cycle. It can also be seen that the algorithm presented in the paper has high computational efficiency and convergence speed.

Table 2. Parameter calculation result when given tool path curve length

$L/mm$	7.8539816	15.707963
$\eta=0.01$	0.500008	0.000009
$\eta=0.05$	0.500076	0.000065
$\eta=0.10$	0.500104	0.000130
$\eta=0.50$	0.500513	0.000621

## V. TRIAL CUTTING A NURBS CURVE

The proposed method has been realized in our developing 3-axis CNC machine tool, which is equipped with an open architecture controller. The NURBS curve depicted in Fig.4 is transformed the format of NC code and then input into the CNC system to cut the paraffin blank. The cutting result of the NURBS curve is shown in Fig.5 and the tool path approximation accuracy is under  $0.5\mu m$  when the time interval is set as 1ms.

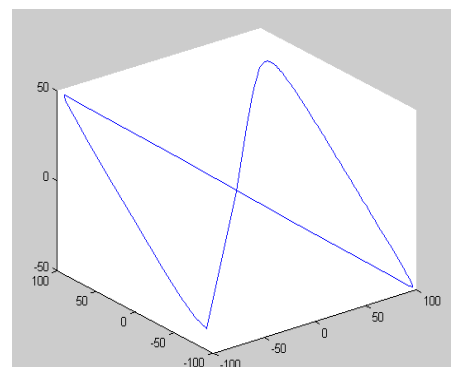
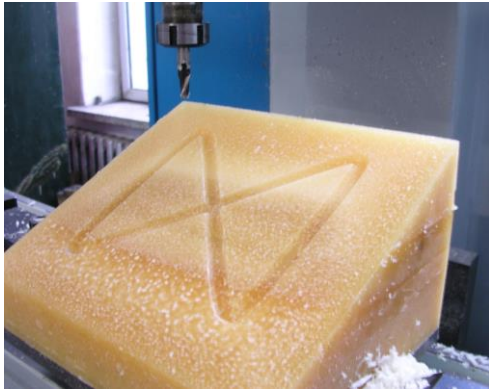


Fig.4 A NURBS curve to be cut





**Fig.5** Result of cutting the NURBS curve

## VI. CONCLUSION

Through our deep research in this paper, we can conclude the following meaningful and valuable conclusions:

(1) NURBS method widely used in the field of CAD can also be directly applied in CNC system.

(2) Through adding approximation algorithm to the controlling software of CNC machines, one can achieve the higher approximation accuracy to tool path curve than the traditional line or circular arc segments approximation method.

(3) The iterative calculation methods presented in the paper have higher computational efficiency and convergence speed. The proposed methods can also provide the theoretic reference for the similar applications of NURBS approximation technology.

## ACKNOWLEDGMENT

This research is wholly supported by the National Natural Science Foundation of China (No.51275485).

## REFERENCES

- [1] Marchenko Tikhon, Tae Jo Ko, "NURBS interpolator for constant material removal rate in open NC machine tools," *International Journal of Machine Tools & Manufacture*, vol.44, 2004, pp. 237-245.
- [2] M. Tsai, C. Cheng, and M. Cheng, "A real-time NURBS surface interpolator for precision three-axis CNC machining," *International Journal of Machine Tools & Manufacture*, vol.43, 2003, pp. 1217-1227.
- [3] B.Bahr, X. Xiao, and K. Krishnan, "A real-time scheme of cubic parametric curve interpolations for CNC systems," *Computers in Industry*, vol. 45, 2001, pp. 309-317.
- [4] S. Nakamura, *Applied Numerical Methods in C*, Prentice-Hall International, Inc, USA, 1993.
- [5] M. Cheng, M. Tsai, and J. Kuo, "Real-time NURBS command generators for CNC servo controllers," *International Journal of Machine Tools & Manufacture*, vol. 42, 2002, pp. 801-813.
- [6] S. Asiaban, M. Moghaddam, and M. Abbaspur, "A real-time scheduling algorithm for soft periodic tasks," *International Journal of Digital Content Technology and its Applications*, vol. 3, 2009, no. 4, pp. 100-111.
- [7] T. Mursalin, F. Eishita, A. Islam, etc., "Real time automated fabric defect detection system using microcontroller," *Journal of Convergence Information Technology*, vol. 3, 2008, no. 1, pp.42-50.
- [8] G. Qiyi, R. Zhang, and B.Greenway, "Development and implementation of a NURBS curve motion interpolator," *Robotics and Computer-Integrated Manufacturing*, vol.14, 1998, pp.27-36.
- [9] X. Zhiming, C. Jincheng, and F. Zhengjin, "Performance evaluation of a real-time interpolation algorithm for NURBS curves," *International Journal of Advanced Manufacturing Technology*, vol.20, 2002, pp. 270-276.
- [10] Liangji Chen, Xianzhang Feng. Calculation of NURBS Machining Paths for Free-form Surfaces. *IEEE ETT* 2010, 4:403-406.
- [11] L. Piegl, "On NURBS: a survey," *IEEE Computer Graphics & Application*, no.1, vol.11, 1991, pp. 55-71.