

## **A METHOD OF REAL-TIME NURBS INTERPOLATION WITH CONFINED CHORD ERROR FOR CNC SYSTEMS**

**Nguyen Huu Quang<sup>\*</sup>, Banh Tien Long**

*Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Ha Noi, Viet Nam*

<sup>\*</sup>Email: [quang.nguyenhuu@hust.edu.vn](mailto:quang.nguyenhuu@hust.edu.vn)

Received: 8 February 2017; Accepted for publication: 10 April 2017

### **ABSTRACT**

This paper presents a method of real-time CNC interpolation for complex curved profiles which can be described with NURBS equations. The real-time interpolation algorithm is based on second order Taylor's expansion with the principle part being a formula for updating parametric value  $u$  after each sampling period. With the updated value of  $u$ , a new interpolated point is calculated based on the DeBoor's algorithm. Chord error caused by the interpolation algorithm depends on two main factors which are machining feedrate and radius of curvature of the NURBS curve. In this paper, an efficient method of limiting chord error is also presented with the basic idea of reducing machining feedrate at positions with a radius of curvature smaller than a critical value. Simulation results on Matlab software verify effectiveness of the proposed method.

*Keywords:* CNC interpolation; CAD/CAM; NURBS; DeBoor's algorithm.

### **1. INTRODUCTION**

Nowadays, parametric curves such as Bezier, B-spline, NURBS, etc. are being widely applied in well-developed CAD systems to design parts with complex curved profiles. On the other hand, conventional CNC machines typically only support motion along straight lines (G01) and circular arcs (G02,G03). To bridge the gap between CAD systems and CNC machines, there are two main approaches: In the first one, original parametric curves is approximated by a sequence of short linear segments with the use of CAM softwares; And in another one, a new parametric interpolator is suggested to be developed for CNC systems, so that it has the ability of calculating interpolated positions in real-time based on the parametric equations of the curve [1 – 5].

In the linear approximation method (the currently popular method), contour error depends on the number of linear segments. In order to achieve the desired tolerance (typically being 10  $\mu m$ ), the number of segments is usually very huge and the length of each segment is very short, leading to the large size of G-code files, the deviation of machining feedrate, as well as the reduction of machining productivity and parts surface quality, etc. In the real-time method, the interpolation algorithm takes account of the commanded feedrate in calculations of the tool

position for next cycle from the current position. Therefore, the machining feedrate will be more stable in the real-time method. Additionally, the size of G-code files will be much smaller, because the geometric parameters of the curves (e.g., control points coordinates, weights, etc.) will be passed to CNC systems instead of end points coordinates of linear segments [1 – 5].

This paper presents a real-time interpolator for NURBS curves. NURBS (Non-Uniform Rational B-Spline) are generalizations of non-rational and rational Bezier, as well as non-rational B-splines curves. They offer one common mathematical form for both standard analytical shapes (e.g., circular arcs, ellipses, etc.) and free-form ones, with the great flexibility to design a large variety of geometries through adapting control points, weights and knot values. Also, they can be evaluated reasonably fast by numerically stable and accurate algorithms. With many advantages listed above and more, NURBS has been widely used in CAD systems today. In that trend, real-time NURBS interpolators will promisingly have expansion applications, and also promote the integration of CAD, CAM and CNC technologies to improve productivity and quality in mechanical production. In this paper, the problem of chord error appeared in the real-time interpolation process is also considered, and a simple but efficient formula to adapt machining feedrate is proposed for the purpose of limiting this kind of error.

## 2. PARAMETRIC NURBS CURVES

A  $p$ th-degree NURBS curve is defined as follows [6]:

$$\mathbf{C}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \frac{\sum_{i=0}^n N_{i,p}(u)w_i\mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(u)w_i}, \quad a \leq u \leq b \quad (1)$$

where,  $\{\mathbf{P}_i = [X_i, Y_i, Z_i]^T\}$  ( $0 \leq i \leq n$ ) are *control points*,  $\{w_i\}$  are the corresponding *weights* of  $\{\mathbf{P}_i\}$ .  $\{N_{i,p}(u)\}$  are  $p$ th-degree B-spline basis functions defined on the *knot vector*  $U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\}$ . The  $p$ th-degree B-spline basis functions are recursively defined as follows:

$$N_{i,0}(u) = \begin{cases} 1, & u \in [u_i, u_{i+1}) \\ 0, & u \notin [u_i, u_{i+1}) \end{cases} \quad (2)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (3)$$

One of the most important algorithms for NURBS is the DeBoor's algorithm [6]. This algorithm is a generalization of the deCasteljau's algorithm for Bezier curves. It provides a fast and numerically stable way to find a point on a NURBS curve, given a  $u$  value in the domain.

Some important geometric properties of NURBS curves (e.g., tangent, curvature, etc.) concern high order derivatives of them. The  $k$ th derivative of a NURBS curve  $\mathbf{C}(u)$ , denoted by  $\mathbf{C}^{(k)}(u)$ , is defined as following:

$$\mathbf{C}^{(1)}(u) = \frac{d\mathbf{C}(u)}{du} = [x'(u), y'(u), z'(u)]^T \quad (4)$$

$$\mathbf{C}^{(k)}(u) = \frac{d\mathbf{C}^{(k-1)}(u)}{du}, \quad k \geq 2 \quad (5)$$

According to [6], the derivative of a  $p$ th degree NURBS curve is a  $(p-1)$ th degree NURBS curve. Therefore, the DeBoor's algorithm can also be applied to calculate high order derivatives of NURBS curves.

### 3. DESIGN OF A REAL-TIME NURBS INTERPOLATOR

Based on input data including: (a) parametric equations of the NURBS path; (b) machining feedrate  $V_s$ ; (c) interpolation period  $T$ ; and (d) the current position of the tool, the real-time NURBS interpolator designed in this section will calculate the target position of the tool for next interpolation cycle in consideration of the chord error tolerance.

#### 3.1. A NURBS interpolation algorithm based on the second order Taylor's expansion

According to differential geometry, the feedrate  $V_s$  along the NURBS path can be calculated as following:

$$V_s = \left\| \frac{d\mathbf{C}(u)}{dt} \right\| = \left\| \frac{d\mathbf{C}(u)}{du} \cdot \frac{du}{dt} \right\| = \|\mathbf{C}^{(1)}(u)\| \cdot \frac{du}{dt} \quad (6)$$

From (6) we can deduce:

$$\frac{du}{dt} = \frac{V_s}{\|\mathbf{C}^{(1)}(u)\|} \quad (1)$$

and,

$$\frac{d^2u}{dt^2} = -\frac{V_s^2 \cdot \langle \mathbf{C}^{(1)}(u), \mathbf{C}^{(2)}(u) \rangle}{\|\mathbf{C}^{(1)}(u)\|^4} \quad (8)$$

where,  $\langle \rangle$  denotes scalar product of two vectors, and  $\| \|$  denotes magnitude of a vector.

Define  $u_k = u(kT)$ ,  $V_{s,k} = V_s(kT)$ . By using Taylor's expansion, we have the approximation up to the second derivative as following:

$$u_{k+1} = u_k + T \cdot \left. \frac{du}{dt} \right|_{t=kT} + \frac{T^2}{2} \cdot \left. \frac{d^2u}{dt^2} \right|_{t=kT} + \varepsilon \quad (9)$$

In Equation (9),  $\varepsilon$  is truncation error which is determined in requirement.

Substituting (7) and (8) into (9) we have the following second order Taylor's approximation:

$$u_{k+1} = u_k + \frac{V_{s,k} \cdot T}{\|\mathbf{C}^{(1)}(u_k)\|} - \frac{V_{s,k}^2 \cdot T^2 \cdot \langle \mathbf{C}^{(1)}(u_k), \mathbf{C}^{(2)}(u_k) \rangle}{2 \cdot \|\mathbf{C}^{(1)}(u_k)\|^4} \quad (10)$$

The Equation (10) has a recursive form and can be used to update the parametric value  $u$  after each interpolation period. With the new value of  $u$ , the next target tool position can be obtained by applying the parametric equations of the NURBS curve and the DeBoor's algorithm:

$$(x_{k+1}, y_{k+1}, z_{k+1}) = \mathbf{C}(u_{k+1}) \quad (11)$$

Equations (10) and (11) form the basis of a NURBS interpolation algorithm. They can be implemented in real-time by digital signal processing systems.

### 3.2. Chord error calculation

In an interpolation period, the tool moves linearly on the chord connecting  $\mathbf{C}(u_k)$  and  $\mathbf{C}(u_{k+1})$  and causes the chord error which is the maximum deviation of the chord from the corresponding curve segment. To evaluate this kind of error, we consider a circular arc which is tangent with and has the same radius of curvature as the NURBS curve at  $\mathbf{C}(u_k)$ , as demonstrated in Figure 1. If the tool didn't move along the NURBS curve but the circular arc in one interpolation period with feedrate  $V_s$ , then it would reach a position  $\mathbf{P}(u_{k+1})$  instead of  $\mathbf{C}(u_{k+1})$ . Let  $\rho_k$  denote the radius of curvature of the NURBS curve at  $\mathbf{C}(u_k)$ , and define  $\ell_k = \|\mathbf{P}(u_{k+1}) - \mathbf{P}(u_k)\|$ . Because the value of  $u_k$  and  $u_{k+1}$  are very close together, the chord error in movement along the NURBS path from  $\mathbf{C}(u_k)$  to  $\mathbf{C}(u_{k+1})$  can be estimated by the chord error,  $e_k$ , in movement along the circular path from  $\mathbf{P}(u_k)$  to  $\mathbf{P}(u_{k+1})$  (with  $\mathbf{P}(u_k) \equiv \mathbf{C}(u_k)$ ). From geometric relationships we can infer:

$$e_k = \rho_k - \sqrt{\rho_k^2 - \left(\frac{\ell_k}{2}\right)^2} \quad (12)$$

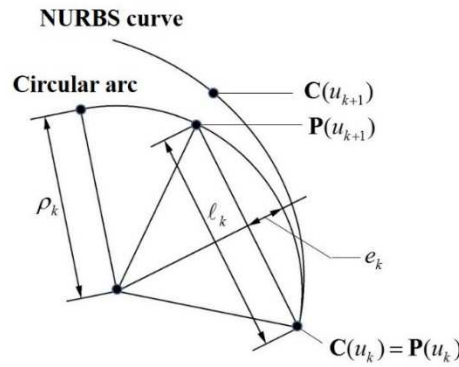


Figure 1. Estimation of chord error.

Additionally,  $\ell_k$  approximates to the path length which the tool moves in one period  $T$ , so:

$$\ell_k = V_{s,k} \cdot T \quad (13)$$

From and we deduce:

$$e_k = \rho_k - \sqrt{\rho_k^2 - \frac{V_{s,k}^2 \cdot T^2}{4}} = \frac{V_{s,k}^2 \cdot T^2}{4 \left( \rho_k + \sqrt{\rho_k^2 - \frac{V_{s,k}^2 \cdot T^2}{4}} \right)} \approx \frac{T^2}{8} \cdot \frac{V_{s,k}^2}{\rho_k} \quad (14)$$

Equation (14) is an estimation of the chord error. It shows that the chord error is dependent on two major variable which are the feedrate,  $V_{s,k}$ , and the radius of curvature of the curve,  $\rho_k$ .

The radius of curvature of the NURBS curve,  $\rho_k$ , can be calculated as following [6]:

$$\rho_k = \frac{\|C^{(1)}(u_k)\|^3}{\|C^{(1)}(u_k) \otimes C^{(2)}(u_k)\|} \quad (15)$$

where, symbol  $\otimes$  denotes vector product of two vectors.

### 3.3. Confining the chord error

Let  $e_{\max}$  denote the chord error tolerance. We have the following constraint:

$$e_k \leq e_{\max} \Leftrightarrow \frac{V_{s,k}^2}{\rho_k} \leq \frac{8 \cdot e_{\max}}{T^2} = \frac{V_c^2}{\rho_c} \quad (16)$$

In (16),  $V_c$  is the commanded feedrate and  $\rho_c = \frac{V_c^2 \cdot T^2}{8 \cdot e_{\max}}$ . The method of confining the chord error is proposed as following:

- If  $\rho_k \geq \rho_c$  then the machining feedrate is maintained as commanded feedrate:  $V_{s,k} = V_c$ .
- If  $\rho_k < \rho_c$  then the machining feedrate in current interpolation cycle is reduced as below:

$$V_{s,k} = V_c \cdot \sqrt{\frac{\rho_k}{\rho_c}} \quad (17)$$

## 4. SIMULATION AND DISCUSSION

Simulation method is used to evaluate the real-time NURBS interpolation algorithm in two cases: without and with consideration of the chord error tolerance. The test NURBS curve is chosen as Figure 2.

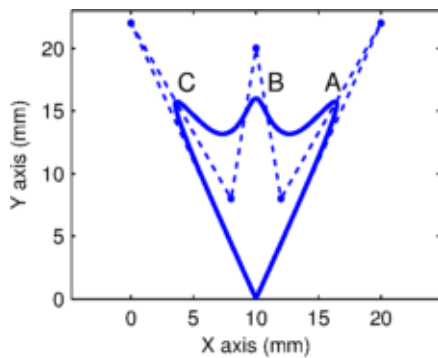


Figure 2. The test NURBS curve (solid line). Dashed line represents its control polygon. A,B,C mark high curvature zones.

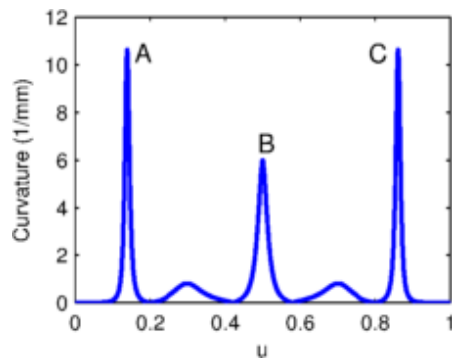


Figure 3. Curvature plot of the test NURBS curve. A,B,C mark corresponding curvatures of A,B,C positions on Figure 2.

As shown in Figure 3, its curvature varies considerably along the curve. In general, this characteristic make the test curve a difficult toolpath for CNC machining [7, 8]. Geometric

parameters of the test NURBS curve are listed in Table 1. The simulation program is developed on the Matlab software with the interpolation period  $T$  chosen as 1 ms.

Table 1. Geometric parameters of the test NURBS curve.

Degree	$p=3$
Knot vector	(0,0,0,0,0.25,0.5,0.75,1,1,1,1)
Control points and weights	$P_0=(10,0,0), w_0=1$
	$P_1=(20,22,0), w_1=1$
	$P_2=(12,8,0), w_2=1$
	$P_3=(10,20,0), w_3=1$
	$P_4=(8,8,0), w_4=1$
	$P_5=(0,22,0), w_5=1$
	$P_6=(10,0,0), w_6=1$

#### 4.1. Evaluation of the NURBS interpolator without confined chord error

The NURBS interpolation algorithm without confined chord error (Section 3.1) is simulated with the test NURBS curve. With different feedrates which are 100 mm/s, 60 mm/s and 30 mm/s, the chord error are shown in Figure 4. It can be seen from Figure 4 the trend of increasing chord error when the feedrate get higher value. With the feedrate of 100 mm/s (6 m/min), the chord error can be larger than 10  $\mu\text{m}$  which is quite large. The chord error decreases to be lower than 5  $\mu\text{m}$  with the feedrate of 60 mm/s, and continues decreasing to about 1  $\mu\text{m}$  with the feedrate of 30 mm/s. These results confirm the idea of reducing chord error by reducing machining feedrate.

For another remark, we can see that the range of machining feedrate below 30 mm/s in which the chord error is about 1  $\mu\text{m}$ , in fact, is suitable with almost normal CNC machining applications. So, the problem of limiting chord error only get more important in high-speed machining.

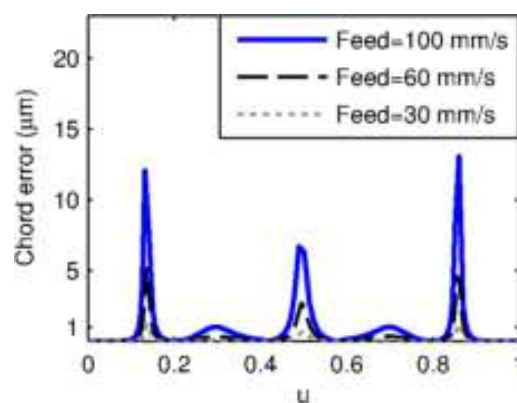


Figure 1. Chord errors caused by NURBS interpolator without confined chord error.

#### 4.2. Evaluation of the NURBS interpolator with confined chord error

The NURBS interpolator with confined chord error is simulated with  $e_{\max} = 1\mu\text{m}$ . When the commanded feedrate is set to be 100 mm/s, Figure 5 (a) show that the chord error is reduced significantly in comparison with the previous simulation results in Figure 4. At several positions, the chord errors are still larger than  $1\mu\text{m}$ , because the feedrate adapting formula is only based on an approximate estimation of the chord error.

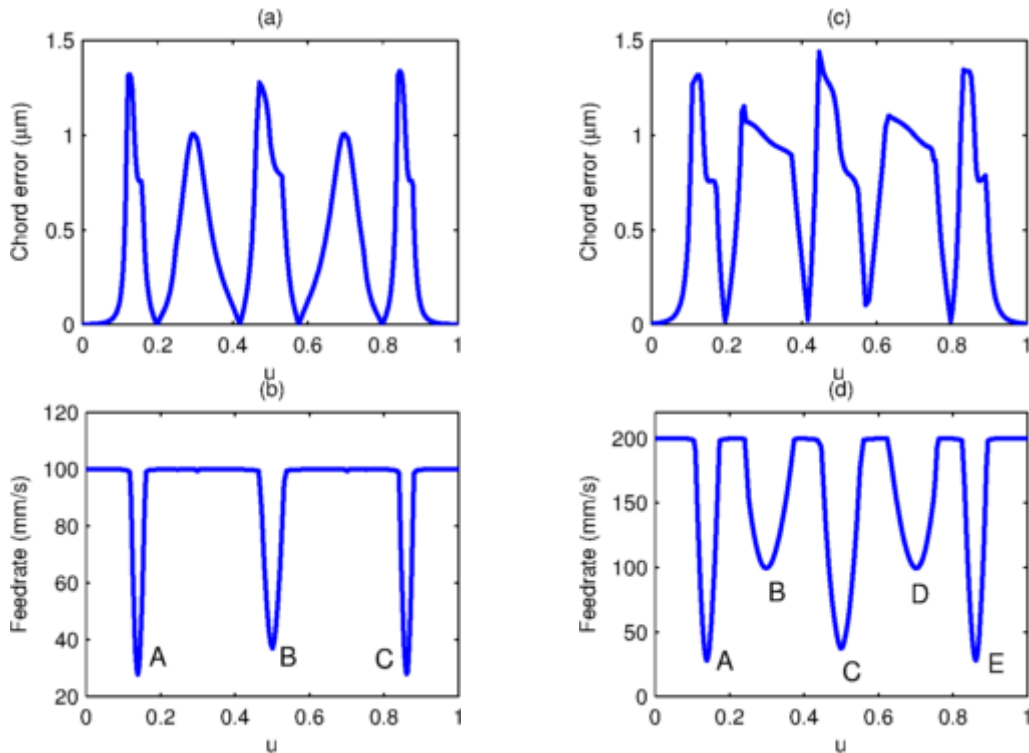


Figure 2. Simulation results of the NURBS interpolator with confined chord error: (a),(b) corresponding to the feedrate of 100 mm/s; (c),(d) corresponding to the feedrate of 200 mm/s.

Figure 5(b) represents the machining feedrate. It can be seen that the feedrate is reduced at some positions (marked as A,B,C), as a result of the error limiting algorithm. At other positions, the machining feedrate is maintained almost the same as commanded feedrate.

When the commanded feedrate is doubled to 200 mm/s, the chord error is hardly greater, as shown in Figure 5(c). This result confirms the effectiveness of the error limiting algorithm. However, the number of positions with feedrate reduction increase, as shown in Figure 5(d). The reason is that the critical radius of curvature,  $\rho_c$ , gets larger when the commanded feedrate get higher, leading to the expansion of the curve segments at which machining feedrate need to adapt. This effect will decrease the average machining feedrate, as well as increase the machining time.

### 5. CONCLUSION

In this paper, a real-time interpolation method is presented for parametric NURBS curves. The interpolation algorithm is based on the second order Taylor's expansion. This method can

maintain a small feedrate fluctuation, but the chord error can be large at positions on the curve which have a high curvature (i.e., has a small radius of curvature). The problem of chord error can be overcome by monitoring the radius of curvature along the NURBS curve during interpolation time, and adapting the machining feedrate such that it get properly smaller values at positions which have a radius of curvature below a critical value,  $\rho_c$ . Simulation results on the Matlab software shows that, although the feedrate adapting formula is quite simple, it is very useful in respecting the chord error tolerance.

**Acknowledgement.** This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2016-PC-061.

## REFERENCES

1. Koren Y., Lo C. C., Shpitalni M. - CNC interpolators: algorithms and analysis, *Asme Prod Eng Div Publ Ped*, Asme, New York, USA **64** (C) (1993) 83–92.
2. Shpitalni M., Koren Y., Lo C. C. - Realtime curve interpolators, *Comput. Des.* **26** (11) (1994) 832–838.
3. Zhiming X., Jincheng C., Zhengjin F. - Performance evaluation of a real-time interpolation algorithm for NURBS curves, *Int. J. Adv. Manuf. Technol.* **20** (4) (2002) 270–276.
4. Hong Tzong Yau, Ming Tzong Lin, Meng Shiun Tsai - Real-time NURBS interpolation using FPGA for high speed motion control, *CAD Comput. Aided Des.* **38** (10) (2006) 1123–1133.
5. Banh Tien Long, Nguyen Huu Quang - A method of real-time NURBS interpolation for CNC systems, *Journal of Science & Technology Technical Universities* **114** (2016) 48–53 (in Vietnamese).
6. Les Piegl, Wayne Tiller, *The NURBS book*, 2nd ed., Springer, New York, 1996.
7. Banh Tien Long, Bui Ngoc Tuyen - *Surface machining theory and application in mechanical engineering*, Vietnam Education Publishing House, Ha Noi, 2012 (in Vietnamese).
8. Banh Tien Long, Tran The Luc, Nguyen Chi Quang - *Theory of cutting tool geometry and machining*, Science and Technics Publishing House, Ha Noi, 2006 (in Vietnamese).