International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)
ISSN: 2305-0543 (Online), ISSN: 2411-6173 (Print)
Vol. 12 (45), Jul. 2022, pp. 5218-5224

Fast Precise and Smooth Approach to Calculate Parameter of NURBS by Arc-Length

ZHENG Shuai[1], WANG Xingbo[1], YAN Jihong[2] and ZHU Kongfeng[2]
[1]Department of Mechanical Engineering, Foshan University, Foshan, China;
[2]Guangzhou CNC equipment co, LTD, Guangzhou, China
*Corresponding Author's Email: 153668@qq.com

***Abstract*—NURBS is the standard model of CNC interpolation calculation at present. It is the basis of CNC system position and speed calculation. Especially, the algorithm to inverse calculate parameter of arc length determines the advantages and disadvantages of CNC speed planning. Due to the lack of analytical calculation relationship between the arc length of NURBS curve and its parameters, it is impossible to accurately calculate parameters through the arc length, which makes it difficult to accurately predict the interpolation points in speed planning. To solve the above problems, a high-precision inverse calculation method of NURBS curve parameters from arc length is proposed. Based on the rapid calculation of arc length by adaptive Simpson integral method, the corresponding relationship between arc length and parameters is established by weighting, and then the high-precision inverse calculation of arc length to parameters is realized by proportional optimization results. The example of curve speed planning shows that the algorithm not only has higher accuracy than other methods, but also can quickly calculate the parameters corresponding to any arc length, and can meet the requirements of high-speed and high-precision interpolation for high-performance CNC.**

Keywords—NURBS, CNC interpolation, Parameter back calculation, Speed planning

I. INTRODUCTION

Interpolation is one of the core calculations of CNC system and it directly affects the accuracy and speed of the CNC system [1]. Early interpolation methods based on small line segment and arc are difficult to meet the needs of modern CNC system of high-performance. NURBS has been widely used as a basic model for real-time interpolating calculation in the world [2]. The calculation of NURBS involves control points, weights, knot vector and parameter. The purpose of calculating the parameter is to calculate the Cartesian spatial points on NURBS, so as to carry out later calculations of position and velocity. In the definitions of NURBS, the parameter is determined by a sequence of real numbers called knot vector mathematically. As is known, arc length and chord length parameter are widely used in description of parameter curves. Since the parameter of NURBS has no geometric significance, it is difficult to directly find the corresponding relationship between the parameter and the arc length or chord length of two interpolation points[3]. Therefore, algorithms to calculate parameter from the arc length have been concerned and adopted by many scholars in interpolation calculation.

The early algorithms for NURBS parameter back calculation mainly include the first-order Taylor's expansion method and the second-order Taylor's expansion method. The algorithm for calculating NURBS curve parameter based on chord length has large amount of calculation and low precision, which is difficult to meet the requirements of high-speed and high-precision interpolation of CNC. Recent studies on the inverse calculation of NURBS curve parameter mainly include: Shuai Ji[4] proposed a method of inverse calculation of parameters by equal chord length, which can obtain small velocity fluctuation under low calculation; Kang min[5] proposed a B-spline arc length parameterization method with G2 continuity, but it was not applied to NURBS curves; Xue Chengxi[6] proposed an extended unit incremental interpolation method for generalized NURBS curves, but the incremental interpolation method will cause cumulative error and lead to insufficient accuracy; Mo Chen[7] proposed an augmented Taylor interpolation method of B-spline based on the traditional Taylor interpolation method, but it will have a large amount of calculation in the application of NURBS; Zezhong C[8] generates NURBS tool path with arc length as parameter, which can obtain smooth surface in three-axis machining, and its accuracy is 10^{-4}, which is difficult to meet the requirements of high-performance CNC interpolation; Jia Chunyang[9] used curve discretization to obtain NURBS curve interpolation with equal arc length, and the error range is only 10^{-2}; W Zhao[10] proposed a generalized arc length increment method, which obtains the interpolation through the arc length increment in each interpolation cycle, but it will still produce cumulative error; Jichun Wu[11] fitted the curve by double arc method and obtained the corresponding relationship between arc length and parameters, so as to realize the inverse calculation of arc length to parameters. The calculation speed is fast, but the accuracy is not enough, resulting in large speed fluctuation; Zezhong C[12] realized the high-speed and high-precision machining of B-spline through the method of piecewise path, but it was not applied to NURBS; Michele Heng[13] proposed a numerically effective interpolation algorithm by applying the concept of feed correction polynomial to NURBS tool path, but polynomial fitting takes more time; Fang Kui[14] realized the parameterization of curve arc length by numerical method, which has high feasibility, but the experiment shows that the

Footnotes: 8-point Times New Roman font;

Manuscript received July 1, 2012; revised August 1, 2012; accepted September 1, 2012.

Article History:
Received Date: Dec. 01, 2021, Accepted Date: Feb. 14, 2022, Available Online: Jul. 01, 2022

process of this method is unstable and the speed fluctuates greatly.

It is not difficult to see that the current research on NURBS parameter inverse calculation is mainly divided into two directions: chord length and arc length. The inverse calculation of parameters from chord length is convenient for the research on speed planning of CNC interpolation, but the chord length is variable, and the uniqueness of the result is greatly reduced. Only when the interpolation is subdivided into small enough, can we get higher precision results. Because the arc length is unique and the interpolation obtained by the inverse arc length parameters can obtain a smooth surface, it has natural advantages compared with the chord length method. Based on this, a method for inverse calculation of arc length parameters of NURBS curves is proposed in this paper. Experimental results show that this method can quickly calculate the parameters corresponding to the arc length of any NURBS curve, and has high accuracy.

II. Related Work

A. NURBS curve definition

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

A NURBS curve of degree p can be expressed as:

$$C(u)=\frac{\sum_{i=0}^{n}\omega_i d_i N_{i,p}(u)}{\sum_{i=0}^{n}\omega_i N_{i,p}(u)},u\in[0,1] \tag{1}$$

which:

$$\begin{cases} N_{i,0}=\begin{cases}1, u_i\le u\le u_{i+1}\\ 0, otherwise\end{cases} \\ N_{i,p}(u)=\dfrac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\dfrac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u)\end{cases}$$

0/0=0 is defined here.

In the above formula, u is the parameter of the curve, p is the number of NURBS curves, which also represents order p+1, and $d_i(i=0,1,\cdots,n)$ represents the control point of the curve, $\omega_i(i=0,1,\cdots,n)$ expressed as the ith weight factor. The knot vector of p-degree NURBS curve controlled by N+1 control points are:

$$U=[u_0,u_1,\cdots,u_{n+p+1}]$$

The first-order derivation of NURBS curves can be expressed as:

$$P'(u)=\sum_{i=0}^{n}d_i N'_{i,p}(u)$$
$$=\frac{\sum_{i=0}^{n}\omega_i d_i N'_{i,p}(u)}{\sum_{i=0}^{n}\omega_i N_{i,p}(u)}-\frac{\sum_{i=0}^{n}\omega_i N'_{i,p}(u)\sum_{i=0}^{n}\omega_i d_i N_{i,p}(u)}{\left[\sum_{i=0}^{n}\omega_i N_{i,p}(u)\right]^2} \tag{2}$$

B. Calculation of NURBS Arc length

Let the curve: $P=P(u),u\in[a,b]$, then the arc length of the curve can be expressed as:

$$s=\int_a^b\|P'(u)\|du$$

The first-order derivation of NURBS curve is obtained by equation (2). In this paper, Simpson integral method is selected to realize the integral operation through multiple iterations. The more iterations, the higher the accuracy. Simpson formula is as follows:

$$\int_a^b f(x)dx=\sum_{k=1}^{n}\int_{x_{k-1}}^{x_k}f(x)dx$$
$$=\frac{h}{6}\sum_{k=1}^{n}\left[f(x_{k-1})+4f(x_{k-\frac{1}{2}})+f(x_k)\right]$$

Because the Simpson integral operation is difficult to control the error, based on it, the adaptive Simpson[15] algorithm is adopted in this paper.

Adaptive Simpson divides the interval $[a,b]$ into two sections each time, and calculates the left and right intervals respectively by the three-point formula. The three-point Simpson formula is as follows:

$$S(a,b)=\int_a^b f(x)dx\approx\frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]$$

According to the input accuracy EPS, determine whether it meets the accuracy requirements by equation above:

$$|S(a,c)+S(c,b)-S(a,b)|<15EPS$$

If equation above holds, it is judged that the set accuracy is met in the corresponding interval; otherwise, the interval that does not meet the accuracy is divided into two sections again, and the accuracy is reduced by half, and then it is judged whether it meets the conditions. This continues until all intervals meet the judgment conditions.

Using the adaptive Simpson algorithm to calculate the arc length of NURBS curve can not only ensure the accuracy, but also reduce repeated calculation, improve the calculation speed, and quickly calculate the arc length of parameter u between any sampling points.

III. NURBS Arc Length Parameterization Method

Set a parameter curve:

$$P=P(u),a\le u\le b$$

IEEE Office - SEET Complex, Federal University of Technology, Owerri, Nigeria, https://aeuso.org
EISSN: 2305-0543, PISSN: 2411-6173

The interval $[a,b]$ where the parameter u is located is equally divided into N parts, and the sampling points of N+1 Parameter u are obtained. The step between adjacent sampling points is taken as Δu. Available:

$$a = u_0 < u_1 < \cdots < u_n = b$$

$$\Delta u = \frac{b-a}{n}$$

Calculate the arc length between each two sampling points ds_i:

$$ds_i = \int_{u_i}^{u_{i+1}} |P'(u)| du, i = 0,1,\cdots,n-1$$

From ds_i calculate Arc length s_i between each sampling point $u_i (i = 0,1,\cdots,n)$ and u_0.

$$s_i = \begin{cases} 0, i = 0 \\ \sum_{j=0}^{i} ds_j, (i = 1,\cdots,n) \end{cases}$$

For any parameter $s \in [s_{i-1}, s_i]$, there are:

$$un = u_{i-1} + \frac{s - s_{i-1}}{ds_{i-1}} \Delta u$$

The above algorithm can preliminarily calculate the parameter UN corresponding to the arc length S. in this paper, the above algorithm is referred to as the decentralization method (the decentralization method is similarly introduced in literature [14]). Obviously, the parameter accuracy obtained by the decentralization method is difficult to meet the requirements of CNC interpolation and needs to be optimized. Figure 1 below is an example to analyze its shortcomings and give the optimization idea.

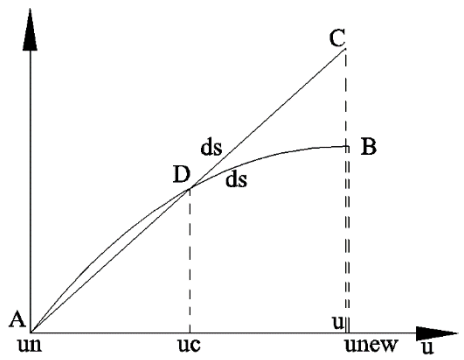


Fig. 1. Error analysis

In Figure 1, *un* is obtained by the above decentralization method. It is assumed that *unew* is the accurate value of the parameter corresponding to the arc length *s*, and the arc length corresponding to arc AB is the arc length between the parameters *u=un* and *u=unew*, so that the length of this arc length is *ds*.

Firstly, the arc length *sun* corresponding to *u=un* is calculated by the adaptive Simpson integral method,

$$sun = s_{i-1} + \int_{u_{i-1}}^{un} |P'(u)| du$$

ds is obtained from the following equation: (*ds* can be negative):

$$ds = s - sun$$

Considering that it is difficult to calculate the distance between the parameters *un* and *unew* on the arc AB in general, and when it degenerates into a straight line, the distance between *un* and *unew* can be accurately calculated through the proportional relationship between similar triangles. A parameter *uc* between *un* and *unew* can be found out according to the "differential triangle" relationship in equation above, and the parameter *uc*:

$$uc = un + \frac{ds}{2ds_i} \cdot \Delta u$$

Let the parameter *uc* correspond to point D on the curve, as shown in Fig. 1. Firstly, the coordinates of the points corresponding to *un* and *uc*: $\left[P(un_x), P(un_y), P(un_z)\right]$, $\left[P(uc_x), P(uc_y), P(uc_z)\right]$ and then the length of ad can be calculated as follows:

$$|AD| = \sqrt{\left[P(uc_x) - P(un_x)\right]^2 + \left[P(uc_y) - P(un_y)\right]^2 + \left[P(uc_z) - P(un_z)\right]^2}$$

Make a straight line AC through point a and point d so that the length of AC is ds, that is:

$$|AC| = ds$$

Then the parameter u corresponding to point C can accurately approximate the exact value unew, and there is the following proportional relationship:

$$\frac{|AD|}{|AC|} = \frac{uc - un}{ur - un}$$

By simplification:

$$ur = \frac{uc - un}{AD} \cdot AC + un = un + \frac{uc - un}{suc} \cdot ds \quad (3)$$

The flow chart of realizing the above algorithm is shown in Fig. 2.

IEEE Office - SEET Complex, Federal University of Technology, Owerri, Nigeria, https://aeuso.org
EISSN: 2305-0543, PISSN: 2411-6173

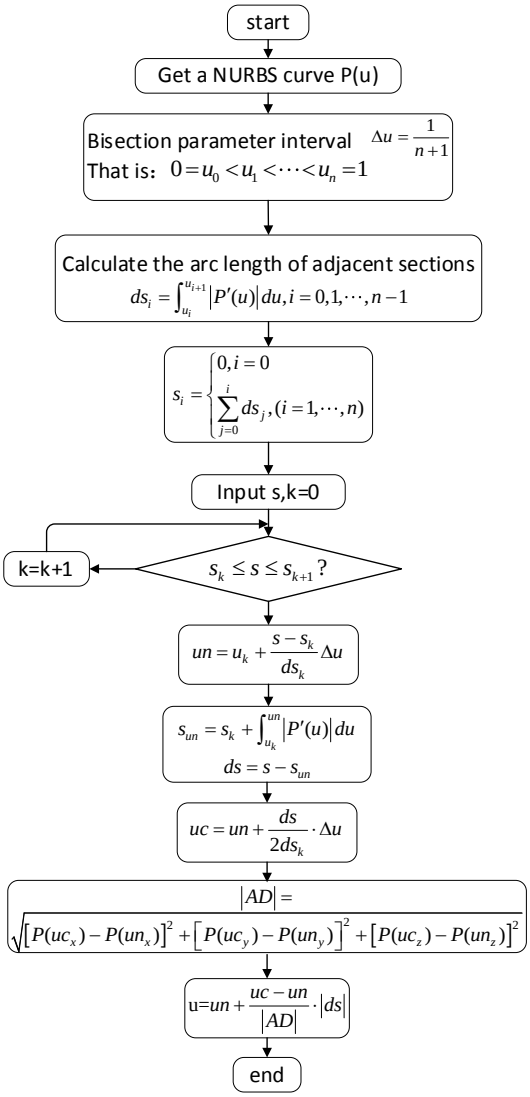


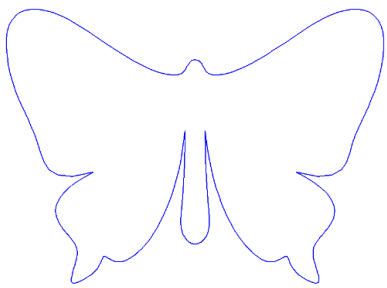
Fig. 2. algorithm flow chart

IV. EXPERIMENT AND COMPARISON

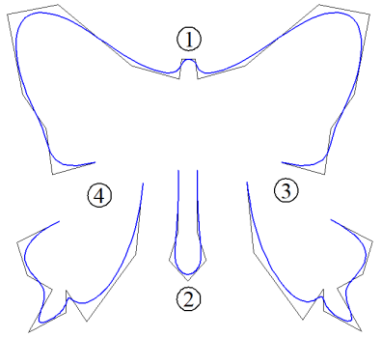
A. *subject: Butterfly curve*

This section takes the commonly used cubic NURBS curve as an example to realize the arc length parameterization algorithm by programming in ARX, verify the feasibility of the algorithm described in Section 2, and analyze the error of the results.

The experimental spatial NURBS curve, its control polygon and control vertex are shown in Fig. 3:



(a) NURBS Curves



(b) decomposition of NURBS curve

Fig. 3. NURBS curve and control polygon

Because the butterfly NURBS curve drawn in Fig. 3(a) consists of four NURBS non closed curves, as shown in Fig. 3(b). Here, take the long first paragraph as an example for verification and speed planning.

There are 18 control points of the first NURBS curve. Using Hartley Judd [16] algorithm to calculate the knot vector, the knot vector of the cubic NURBS curve of these 18 control points is obtained as follows:

$$U = [0,0,0,0,0.05635873,0.12824176,0.20256601,$$
$$0.29967945,0.38319131,0.45319495,0.48849730,$$
$$0.51150269,0.54680504,0.61680869,0.70032054,$$
$$0.79743398,0.87175823,0.94364126,1,1,1,1]$$

There are 22 knot vectors in total, and the weight factor is 1. The first NURBS curve as shown in Fig. 3 (b) can be drawn, and the total arc length of the curve is 37.8707857920mm.

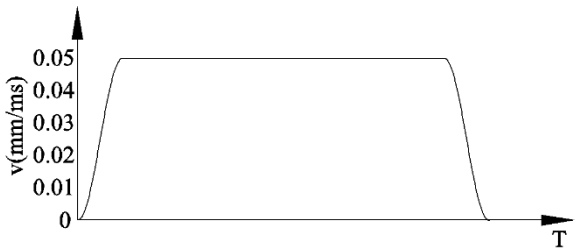
B. *experimental content: S-curve comparison*

The feed speed of CNC has the process of acceleration, uniform speed and deceleration. In this experiment, the uniform speed is the maximum feed speed, which is about v = 0.05mm/ms. The interpolation period is taken as t = 1ms, so the arc length of each cycle is $ds = v \cdot T$. In the acceleration stage, the curve is divided into several increasing arc length segments, several equal arc length segments in the uniform speed stage, and several decreasing arc length segments in the deceleration stage, so as to realize the process of acceleration, uniform speed and deceleration under the condition of equal cycle.

Select the sine function sin to realize the S-shaped increase or decrease of speed, that is,

$$v = 0.05(\sin(\theta - \frac{\pi}{2}) + 1) / 2 (0 < \theta < \pi)$$

the ideal speed curve is shown in Fig. 4:



IEEE Office - SEET Complex, Federal University of Technology, Owerri, Nigeria, https://aeuso.org
EISSN: 2305-0543, PISSN: 2411-6173

Fig. 4. ideal speed curve

According to the algorithm described in Section 2, the start parameters and end parameters corresponding to each arc are calculated, and the error is tested by calculating the arc length between two adjacent parameters.

C. *experimental results*

Experiment 1.

The adaptive Simpson algorithm is used to calculate the arc length, the accuracy is 10^{-10}mm, and the curve is equally divided into 100 segments, that is, the parameter step size is taken $\Delta u = 0.01$.

After 0.689s operation, 852 points are calculated. Limited to space, only a few comparison results are listed in Table 1.

TABLE I. THE DATE OF $\Delta u = 0.01$

parameter	Theoretical value (mm)	Actual value (mm)
u0=0	—	—
u1=0.0000001157	0.0000136686	0.0000136686
u2=0.0000005784	0.0000546593	0.0000546593
u3=0.0000016189	0.0001229274	0.000a1229273
…	…	…
u94=0.0352946975	0.0499863314	0.0499863329
u95=0.0364409374	0.0500000000	0.0499999999
…	…	…
u757=0.9642286815	0.0500000000	0.0499999999
u758=0.9653750559	0.0499920137	0.0499920142
u759=0.9665200503	0.0499623812	0.0499623845
…	…	…
u852=1.0000000000	0.0000056822	0.0000056813

In Table 1, the u0-u95 arc length increases gradually, which is the acceleration stage; The arc length of u95-u730 is equal, which is a constant velocity stage; u757-u852 arc length decreases gradually, which is the deceleration stage. After comparing the calculated theoretical value with the actual value point by point, the error curve is drawn by subtracting the actual value from the theoretical value, as shown in Figure 5:

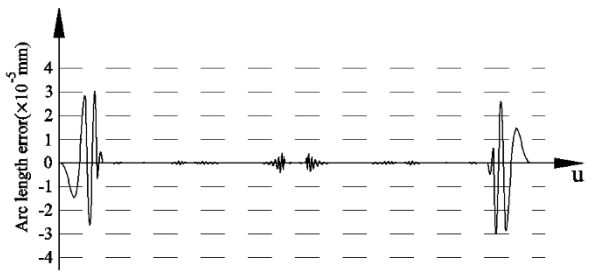


Fig. 5. $\Delta u = 0.01$ error diagram

It can be seen that the error curve in Figure 5 is in a central symmetrical shape as a whole, the arc length error in the acceleration and deceleration stages at both ends is large, and the constant speed stage is relatively stable with a little fluctuation. But the overall error can be maintained at $3\times10^{-5}mm$.

Experiment 2.

$\Delta u = 0.001$ calculate the interpolation point.

The running time of the program is 0.294s, and a total of 852 points are calculated (some results are listed in Table 2).

TABLE II. THE DATE OF $\Delta u = 0.001$

parameter	Theoretical value (mm)	Actual value (mm)
u0=0	—	—
u1=0.0000001157	0.0000136686	0.0000136686
u2=0.0000005784	0.0000546593	0.0000546593
u3=0.0000016189	0.0001229274	0.0001229274
…	…	…
u94=0.0352946975	0.0499863314	0.0499863314
u95=0.0364409374	0.0500000000	0.0500000000
…	…	…
u757=0.9642286815	0.0500000000	0.0500000000
u758=0.9653750559	0.0499920137	0.0499920137
u759=0.9665200502	0.0499623812	0.0499623812
…	…	…
u852=1.0000000000	0.0000056822	0.0000056813

Error curve, as shown in Fig. 6:

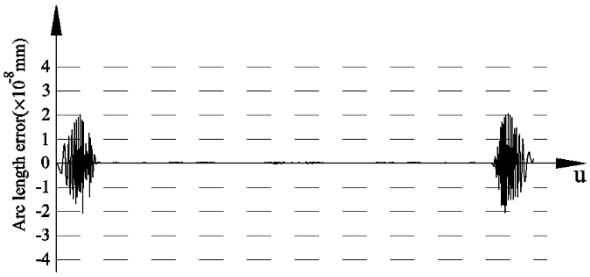


Fig. 6. $\Delta u = 0.001$ error diagram

The error curve shown in Fig. 5 still fluctuates greatly at both ends of acceleration and deceleration, but the maximum error has been controlled at $2\times10^{-8}mm$, and the error in the constant speed part is less than $1\times10^{-9}mm$, which can fully meet the requirements of high-performance interpolation.

D. *experimental features: high precision and stable speed*

The algorithm studied in this paper mainly focuses on the calculation of arc length. The longer the arc length, the higher the accuracy and the longer the calculation time. From the experimental results, When the parameter step Δu is taken as 0.01, the arc length of each section is greater than Δu is taken as 0.001 long, and the calculation time of each arc length increases accordingly. With the same precision setting, Δu =0.001 ratio Δu =0.01 although the calculation times are increased, the time is shorter, which shows that the accuracy is increased and the time is shorter.

IEEE Office - SEET Complex, Federal University of Technology, Owerri, Nigeria, https://aeuso.org
EISSN: 2305-0543, PISSN: 2411-6173

In order to investigate the stationarity of speed with increased accuracy. The chord length between each two interpolation points is divided by the period to obtain the speed of each period, and the actual speed curve is drawn, as shown in Fig. 7.

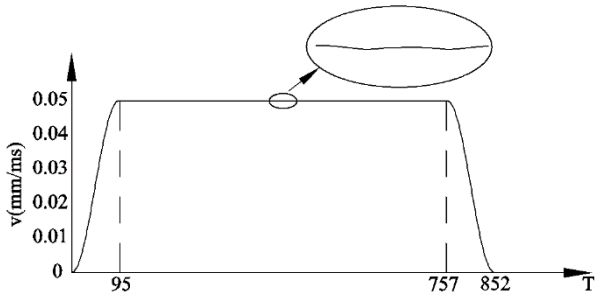


Fig. 7. actual speed curve

As a whole, the speed in Figure 7 shows three stages: acceleration, uniform speed and deceleration. In the acceleration and deceleration stages, there is no abnormality in the state of increasing and decreasing speed, but there are small speed fluctuations in the constant speed stage, as shown in the part framed by the ellipse in the Figure. Therefore, the velocity fluctuation rate in the uniform stage is calculated according to equation (4),

$$\delta_i = \frac{v_{i+1} - v_i}{v_i} = (1 - \frac{|P(u_{i+1}) - P(u_i)|}{v_i \cdot T}) \times 100\% \qquad (4)$$

And draw the velocity fluctuation curve, as shown in Figure 8:

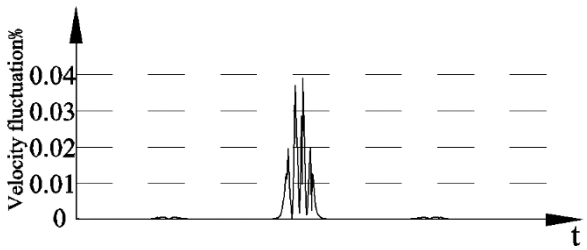


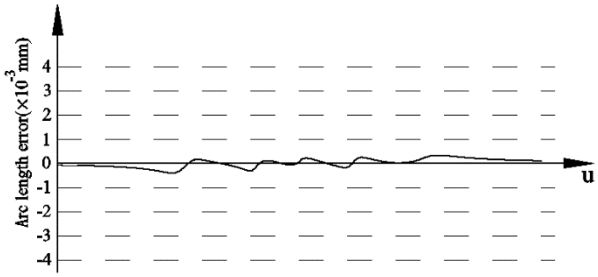
Fig. 8. velocity fluctuation

After analysis, the reason for the above fluctuations is that the arc length is used to calculate the parameters and the approximate chord length is used to calculate the speed: in the uniform speed stage, the equal arc length is used to calculate the parameters, and the velocity is obtained by dividing the chord length by the period. Therefore, there will be large velocity fluctuations in the place with large curvature. In Figure. 8, the velocity fluctuation rate fluctuates greatly in the middle, corresponding to the middle part of paragraph 1 in Figure. 3 (b), which is also a section with large curvature, while the velocity is stable in other places with small curvature. The maximum volatility remained within 0.04%, still relatively stable.

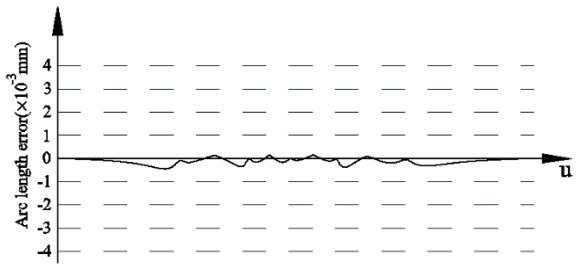
E. comparison

It is compared with the traditional first-order Taylor expansion method, second-order Taylor expansion method and arc length parameterization method in Fang Kui [14].

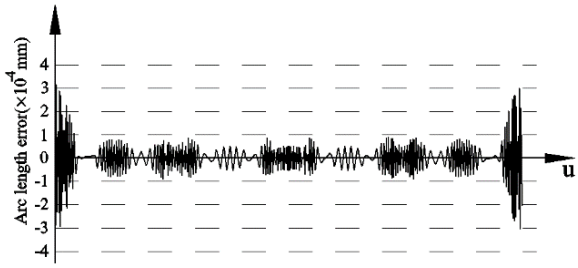
For the same NURBS curves of the above experimental objects, including control points, weight factors and knot vectors, the error curves of the three methods are drawn after calculation, as shown in the following Figure:



(a) First order Taylor error curve



(b) Second order Taylor error curve

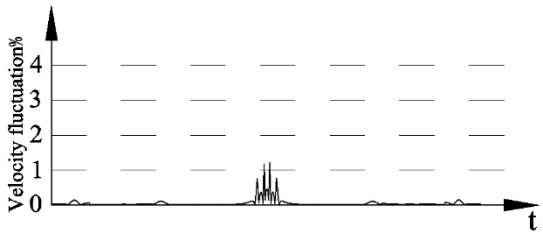


(c) Error curve of fangkui method

Fig. 9. error curves of three methods

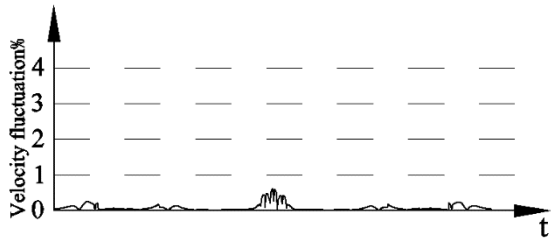
There are three error curves shown in Fig. 9, in which the error of each arc length of the first-order Taylor expansion method is less than $1\times10^{-3}mm$； The error of second-order Taylor expansion development is also less than $1\times10^{-3}mm$ ； The arc length parameterization method in reference [14] can control the arc length error within $2\times10^{-4}mm$, but there are violent fluctuations in the whole process. According to the above method, the arc length error can be controlled within 3 when the parameter step is 0.01 and 0.001, For the case within respectively $3\times10^{-5}mm$ 、 $2\times10^{-8}mm$, the method described in this paper has high accuracy.

In terms of speed fluctuation, the speed fluctuation rates of the three methods are shown in Fig. 10:

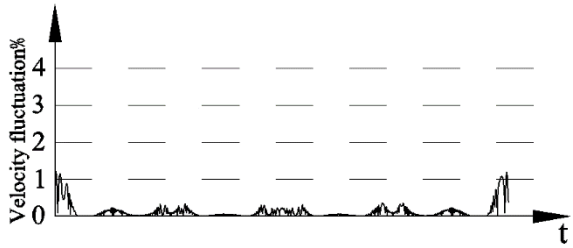


IEEE Office - SEET Complex, Federal University of Technology, Owerri, Nigeria, https://aeuso.org
EISSN: 2305-0543, PISSN: 2411-6173

(a) First order Taylor velocity fluctuation



(b) Second order Taylor velocity fluctuation



(c) Velocity fluctuation rate of fangkui method

Fig. 10. velocity fluctuation rate of three methods

The three velocity volatility curves shown in Fig. 10, of which the maximum velocity volatility of the first-order Taylor method is about 1%; The velocity fluctuation rate of the second-order Taylor method is kept within 1%; The volatility of Fang Kui[14] method is slightly larger, and the maximum volatility is about 2%. As mentioned earlier, the velocity fluctuation rate of the method in this paper can be maintained within 0.04% and has high stability.

V. Conclusion

This paper introduces a method to reverse calculate the parameters from the arc length of NURBS curve. Firstly, the results are obtained by dividing and weighting the parameters, and then optimized by using the proportional relationship of similar triangles. In this way, the corresponding relationship between the arc length of NURBS curve and the parameters is established. The experimental results show that When step size Δu is 0.01, 852 interpolation points can be calculated in 0.689s, and the accuracy remains at Within $3\times10^{-5}mm$; When Δu is 0.001, 852 interpolation points can be calculated in 0.294s, and the accuracy can be maintained at Within $2\times10^{-8}mm$ and with relatively stable speed, it can meet the requirements of high-speed and high-precision interpolation for high-performance CNC.

Acknowledgment

The research work is supported by the Ministry of Industry and Information Technology under project on information protection for numerical control systems with No. TC2000H03A and Department of Guangdong Science and Technology under project 2020B1111410002. The authors sincerely present thanks to them all.

References

[1] WU Jichun, XU Huiyu, ZHOU Huicheng. Research on fast NURBS interpolation with arc length as parameter[J]. NC Technology, 2020, 2: 185-190.

[2] XU Rongzhen, XIE Le. Adaptive parametric interpolation scheme with limited acceleration and jerk values for NC machining[J]. Int J Adv Manuf Technol, 2008, 36: 343-354.

[3] ZHANG Liyan, BIAN Yuchao, CHEN Hu. Implementation of a CNC NURBS curve interpolator based on control of speed and precision[J]. International Journal of Production Research, 2009, 47(6): 1505-1519.

[4] Shuai Ji,Tianliang Hu. A NURBS curve interpolator with small federate fluctuation based on arc length prediction and correction[J]. The International Journal of Advanced Manufacturing Technology, 2020, 111: 2095-2104.

[5] Kang Min, Yanyan Sun. An Improved B Spline Fitting Method with Arc Length Parameterization G2 continuous Blending,and Quality Refinement[J]. International Journal of Precision Engineering and Manufacturing,2019.

[6] Xue-Cheng Xi,Hao Chen. Extended unit arc length increment interpolation for generalized NURBS curve in multi-axis EDM[J]. Precision Engineering, 2019, 59: 81-89.

[7] Mo Chen, Wan-Sheng Zhao. Augmented Taylor's expansion method for B-spline curve interpolation for CNC machine tools[J]. International Journal of Machine Tools & Manufacture, 2015, 94: 109-119.

[8] Zezhong C.Chen, Maqsood A.Khan. A new approach to generating arc length parameterized NURBS tool paths for efficient three-axis machining of smooth, accurate sculptured surfaces[J]. Int J Adv Manuf Technol, 2014.

[9] JIA Chunyang, YANG Yue, CHEN Feng. NURBS curve discrete algorithm based on equal arc-length principle. Computer Engineering and Applications, 2014, 50（3）：165-167.

[10] W Zhao,M Chen. A novel trajectory interpolation algorithm for WEDM- Unit Generalized Arc Length Increment Method[J]. The Seventeenth CIRP Conference on Electro Physical and Chemical Machining, 2013,6:256-261.

[11] WU Jichun,ZHOU Huicheng. Fast NURBS interpolation based on the biarc guide curve[J]. Int J Adv Manuf Technol, 2012,58:597-605.

[12] Zezhong C.Chen,Maqsood A.Khan. Piecewise B-Spline Tool Paths With the Arc-Length Parameter and Their Application on High Feed, Accurate CNC Miiiing of Free-Form Profiies[J]. Journal of Manufacturing Science and Engineering, 2012,134:1-13.

[13] Michele Heng,Kaan Erkorkmaz. Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability[J]. International Journal of Machine Tools & Manufacture, 2010,50:281-293.

[14] FANG Kui. Interpolation method for approximate arc length parameterization of parametric curve[J]. Chin. J. CAD & CG, 1996,8(2):115-120.

[15] TANG Jum,YANG Daojie. Research on recursive implementation of adaptive integral[J]. China High-Tech Enterprises, 2016,27:13-15.

[16] David Salomon. Curves and Surfaces for Computer Graphics[M]. Northridge: California State University,2006.