Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

A Study of NURBS Interpolation Algorithm with Limited Chord Error for EDM*

Huang Ruining and Hua Xinze
School of Mechanical Engineering and Automation
Harbin Institute of Technology ，Shenzhen
Shenzhen 518055, China
hrn@hit.edu.cn

***Abstract* - The traditional EDM interpolation for NURBS curve is to divide the curve into several straight line segments and arc segments for linear and arc interpolation. It takes a long time to generate a large number of G codes, and difficult to control the machining error. In this paper, an interpolation algorithm based on NURBS curve for EDM is proposed which mainly includes two parts: Firstly, using the NURBS interpolation with the chord error control model as a real-time preprocessing to generate a series of small interpolation line segments and store in the loop buffer. Secondly, interpolate the line segment in the loop buffer with the fine interpolation algorithm. Experiment results validate that the EDM interpolation algorithm based on NURBS curve can control the machining error more effectively and improve the machining efficiency greatly.**

***Index Terms* – EDM, interpolation, NURBS, loop buffer.**

I. INTRODUCTION

Electrical discharge machining (EDM) is one of the most extensively used non-traditional machining processes for manufacturing of molds and dies, machining of materials with high hardness and high strength. In EDM, material removal is performed by a series of electrical discharges occurring within the electrical insulated gap between an electrode and a workpiece. During the discharges, a hightemperature plasma channel is created by the applied high voltage in the gap, causing melting and vaporizing of the workpiece material[1][2]. As the core of the CNC system, interpolation determines the efficiency and accuracy of tool path in the process of machining, and directly affects the quality of the workpiece. Different from traditional machining, the feedrate of EDM is very slow, and the short circuit often occurs between electrodes and workpieces, so traditional interpolation algorithm is seldom used directly in EDM.

At present, the research on EDM interpolation is mainly focused on pulse incremental interpolation, and the direct interpolation of parametric curve is very few. References[3][4] are all based on simple arc and straight line interpolation algorithm. The curve is divided into many arcs and straight lines and interpolated in the form of G00\G01. Reference[5][6] proposed a interpolation algorithm for WEDM which is based on unit arc length increment method. In each interpolation period, the interpolation reference point moves 1 unit arc length along the curve and the corresponding increments of each axis are accumulated. All the above algorithms can only interpolate straight lines and arcs, and it is not competent for direct interpolation of general parametric curves. In actual production, complex parts' outlines are always expressed by NURBS curves and surfaces in CAD/CAM system[7]. The traditional interpolation method needs divide the NURBS curve into a large number of small linear or arc segments, and use the traditional linear/circle interpolator to interpolate those dense segments. Two disadvantages have been found in the traditional machining approaches. First, a complex toolpath planning in CAM is normally needed, and the resulting large NC files may occupy a large amount of computer numerical control (CNC) memory[8], Secondly, it can not realize the reverse interpolation. Short circuit often occur between the electrodes and the workpiece. In this case, the electrodes need to be turn back, but this kind of turn back is not simply to make the electrodes retreat, but needs to be turned back according to the original feeding path to ensure that there is no interference damage to the processed part, and after recovery can continue machining[9][10].

In this paper, an EDM interpolation algorithm with chord error control model based on NURBS curve is proposed to overcome the above disadvantages. Two stage is used with the details as follows. In the first stage, NURBS interpolation with chord error control model called in real-time execution to generate a series of small segments. Then these segments are stored as interpolation blocks in the established loop buffer. In the second stage, multiple linear interpolations are executed along the small segments fetched from the loop buffer, it can be called fine interpolation. While the first stage periodically runs, the second stage maintains a stable EDM process. Moreover in the loop buffer, besides the small segments that need to be fine interpolated, there are also segments that have been completed. These segments are used for back interpolation (turn back the electrodes in case of short circuit). Finally, the effectiveness of the algorithm is verified by simulation experiment on MATLAB.

The rest of this paper is arranged as follows: Section 2 discusses the the principle of NURBS curve and the traditional NURBS interpolation. Section 3 details the algorithm that

*This work was partially supported by the National Natural Science Foundation of China (No.51475107), and Shenzhen Basic Research Program (No. JCYJ20170811160440239).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

proposed in this paper. The experimental results and analysis are provided in Section 4. Section 5 gives the conclusions.

II. NURBS Interpolation

NURBS interpolation is a real-time interpolation, which is often used in high-speed milling. Its principle is to calculate the parameters of the next interpolation point based on mathematical recursion formula according to the coordinate data, curve parameters and feedrate of the current point[11]. In this section, the definition of NURBS curve will be introduced firstly, then explain the principle of NURBS interpolation in detail.

A. *Definition of NURBS Curve*

A k-order NURBS curve can be expressed as a rational polynomial vector function:

$$\mathrm{C}(u)=\frac{\sum_{i=0}^{n} w_i c_i N_{i,k}(u)}{\sum_{i=0}^{n} w_i N_{i,k}(u)}, u\in[0,1]. \tag{1}$$

u is the parameter of NURBS curve equation, k is the order of NURBS curve, c_i (i=1,2,... n) is the control point to form a control polygon, w_i (i = 1,2,... n) is the weight factor of the corresponding control point, and $N_{i,k}(u)$ is the k-order B-spline basis function defined on the node vector U. The node vector is:

$$U=[\hat{u}_0,\hat{u}_1,\ ...,\hat{u}_k,\hat{u}_{k+1},...,\hat{u}_n,\hat{u}_{n+1},\ ...,\hat{u}_{n+k+1}]. \tag{2}$$

Usually $\hat{u}_0 = ... = \hat{u}_k = 0$, $\hat{u}_{n+1} = ... = \hat{u}_{n+k+1} = 1$, the other values ranged from 0 to 1 and increased monotonously. The recurrence formula of the k-th basis function $N_{i,k}(u)$ is as follows:

$$\begin{cases} N_{i,0}(u)=\begin{cases}1, & \hat{u}_i<u<\hat{u}_{i+1}\\ 0, & other\end{cases} \\ N_{i,k}(u)=\dfrac{u-\hat{u}_i}{\hat{u}_{i+1}-\hat{u}_i}N_{i,k-1}(u)+\dfrac{\hat{u}_{i+k+1}-u}{\hat{u}_{i+k+1}-\hat{u}_{i+1}}N_{i+1,k-1}(u)\end{cases}. \tag{3}$$

B. *Interpolation parameter calculation*

The parameter u corresponds to the points on the curve one by one. NURBS interpolation process is to segment the curve by calculating the interpolation parameters, and use chord to approximate arc.

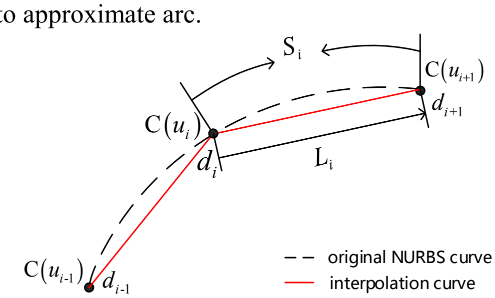

Fig. 1 Interpolation parameter calculation

The parameter of the current point d_i is known to be u_i and its coordinates are C(u_i). The feedrate given in the current interpolation period is V(u_i). If the parameter increment is Δu, then we can calculate the parameter u_{i+1} with the formula(4):

$$u_{i+1}=u_i+\Delta u. \tag{4}$$

Figure 1 shows that the real arc S_i is approximated by chord L_i between interpolation point d_i and d_{i+1}. The chord length $\|L_i\|$ is determined by the given feedrate V(u_i) in the interpolation period, we call it the interpolation step. If the interpolation period is T, there are:

$$\|L_i\|=\mathrm{V}(u_i)\cdot T=\|\mathrm{C}(u_{i+1})-\mathrm{C}(u_i)\|. \tag{5}$$

The interpolation parameters correspond to the time series one by one, so the parameter u can be regarded as a function of time t. Here $u(t_i) = u_i$, $u(t_{i+1}) = u_{i+1}$. At present, the commonly parameter calculation method is Taylor expansion method[12]. The function of parameter u to time t expanded into Taylor expression at t_i time:

$$u(t)=u(t_i)+u'(t_i)(t-t_i)+...+\frac{u^{(n)}(t_i)}{n!}(t-t_i)^n. \tag{6}$$

Note $T = t_{i+1} - t_i$, then at t_{i+1}:

$$u(t_{i+1})=u(t_i)+u'(t_i)T+...+\frac{u^{(n)}(t_i)}{n!}T^n. \tag{7}$$

Usually, the interpolation period T is at the level of milliseconds, and the higher-order terms of formula (7) can be ignored when calculating parameter u_{i+1}. Therefore, the calculating formulas of parameter u_{i+1} based on second-order Taylor expansion is:

$$u(t_{i+1})=u(t_i)+u'(t_i)T+\frac{u^{(2)}(t_i)}{2!}T^2. \tag{8}$$

For formula (8), the problem of calculating interpolation parameters is transformed into the problem of calculating the derivative of parameter u to time t. As shown in Figure 1, if the tool moves along the NRUBS curve during interpolation, the feedrate V(u) will be as follows:

$$\mathrm{V}(u)=\left\|\frac{d\,\mathrm{C}(u)}{dt}\right\|=\left\|\frac{d\,\mathrm{C}(u)}{du}\right\|\frac{du}{dt}. \tag{9}$$

The derivative calculation of NURBS curve for parameter u has been given in Reference [13].

$$\frac{du}{dt}=\frac{\mathrm{V}(u)}{\|\mathrm{C}'(u)\|}. \tag{10}$$

The second derivative of parameter u for time t can be calculated by the first derivative and the second derivative of NURBS curve at parameter u. The specific formulas are as follows:

$$\frac{d^2u}{dt^2}=\frac{d}{dt}\frac{\mathrm{V}(u)}{\|\mathrm{C}'(u)\|}=\frac{\mathrm{V}^2(u)\|\mathrm{C}''(u)\|}{\|\mathrm{C}'(u)\|^3}. \tag{11}$$

If the parameter of the current point u_i, the interpolation period T and the feedrate V(u_i) are known, the parameter of the next interpolation point u_{i+1} can be calculated with the formula(12).

$$u_{i+1}=u_i+\frac{\mathrm{V}(u)T}{\left\|\mathrm{C}^{'}(u)\right\|_{u=u_i}}+\frac{-(\mathrm{V}(u)T)^2\left\|\mathrm{C}^{"}(u)\right\|_{u=u_i}}{\left\|\mathrm{C}^{'}(u)\right\|^3_{u=u_i}}. \quad (12)$$

III. NURBS Interpolation With Limited Chord Error

EDM is different from traditional milling. Its feedrate very slowly and uncertainty, the short circuit often occurs between electrode and workpiece. So, it does not directly use NURBS interpolation for real-time interpolate. In this paper, we use the NURBS interpolation based on chord error control model to preprocess NURBS curve in real−time to generate a series of small line segments and store in the loop buffer. And then, fine interpolate the small line segments which extract from the loop buffer one by one.

A. Chord Error Control Model

The feed step is usually very small. In one interpolation period, the small curve segment corresponding to the chord L_i is usually approximated as an arc as shown in Figure 2.

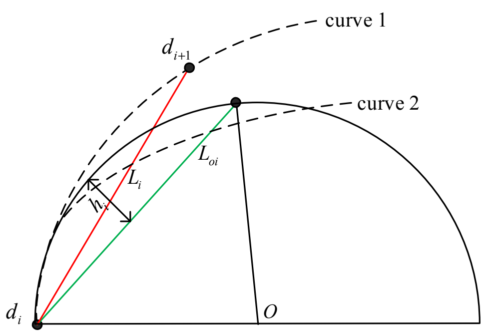

Fig. 2 Chord error estimate of the interpolated curve.

Curve1 is a part of a NURBS curve whose curvature ρ_i at d_i equals one part of the circle O's radius. d_{i+1} is the next interpolation point to be calculated. h_i is considered as chord error of arc segment in circle O. The chord error of L_i is approximately equal to h_i . According to geometric knowledge, the chord error of an arc can be expressed by its radius and chord length through formula (12).

$$h_i=\rho_i-\sqrt{\rho_i^{\,2}-\left(\frac{|L_i|}{2}\right)^2}. \quad (13)$$

The curvature is calculated by the first derivative and the second derivative of the curve at the interpolation point d_i combined with formula (12):

$$\rho_i=\frac{\left\|\mathrm{C}^{'}(u_i)\right\|^3}{\left\|\mathrm{C}^{'}(u_i)\times\mathrm{C}^{"}(u_i)\right\|}. \quad (14)$$

Given the h_{max}, the length of the interpolation line segment satisfying the maximum chord error can be calculated as the formula (14).

$$|L_i|_{\max}=2\sqrt{\rho_i^{\,2}-(\rho_i-h_{\max})^2}. \quad (15)$$

Through formula (5), (12) and (15), the calculation of the interpolation parameters can be summarized as follows:

$$u_{i+1}=u_i+\frac{|L_i|_{\max}}{\left\|\mathrm{C}^{'}(u)\right\|_{u=u_i}}+\frac{-|L_i|^2_{\max}\left\|\mathrm{C}^{"}(u)\right\|_{u=u_i}}{\left\|\mathrm{C}^{'}(u)\right\|^3_{u=u_i}}. \quad (16)$$

The distance between the midpoint of chord and the midpoint of arc line is used as the estimated value in calculating the actual chord error of curves. For curve 1, the chord error can be effectively controlled within the range of h_{max} by the above model. However, for curve2, the actual chord error is larger than that obtained from approximate arc. It is necessary to further modify the interpolation parameters in order to make the error meet the requirements. Therefore, the model uses "dichotomy" to iterate the parameters.

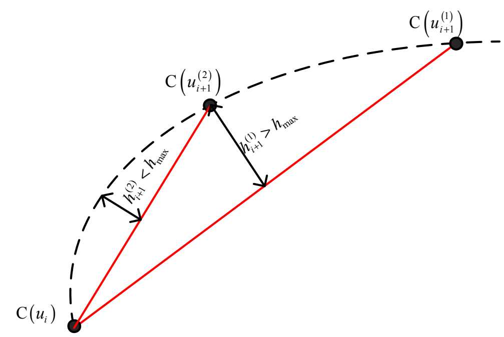

Fig. 3 Amendment of Interpolation Parameters by Dichotomy.

As shown in Figure 3, after each new interpolation parameter is obtained, it is necessary to calculate whether the line segments generated by the parameter meet the requirements of chord error. If it is less than h_{max}, the next interpolation parameter calculation is continued. Otherwise, the median values of u_i and u_{i+1} are taken as new values of u_{i+1} and iterated continuously until the error is satisfied.

B. Loop Buffer

A series of small line segments are obtained by NURBS interpolation. The next step is to fine interpolate these small line segments, but due to the EDM feedrate are very slow and always uncertain, so it is necessary to store these interpolated segments for the continuity of machinin. A loop buffer, as shown in Figure 4, is established, which divided a CNC memory into three parts: reserved area, unprocessed area and unused area. In the reserved area, there are blocks that have been fine interpolated. the unprocessed area is used to store the new interpolation blocks calculated by NURBS interpolation. Tail records the location of the next block to be fine interpolated. Head records the location of the next interpolation block to be added. Each time only one block is extracted from the buffer for fine interpolation. When the fine interpolation completed, the block will not be discarded directly. Instead, the whole reserved area will be moved forward and the earliest one will be discarded. The size of the reserved area is fixed. It can be set by CNC system. Its size determines the scope of backtracking interpolation when a short circuit occurs.

In addition, in order to improve the calculation efficiency of algorithm, multithreading is used in the process of software implementation. One thread performs NURBS interpolation and the other thread performs fine interpolation. When there is no extra space in the buffer, thread 1 goes to sleep. After thread 2 finishes fine interpolation of a block, it generates blank space, and then wakes up thread 1 to continue NURBS interpolation.

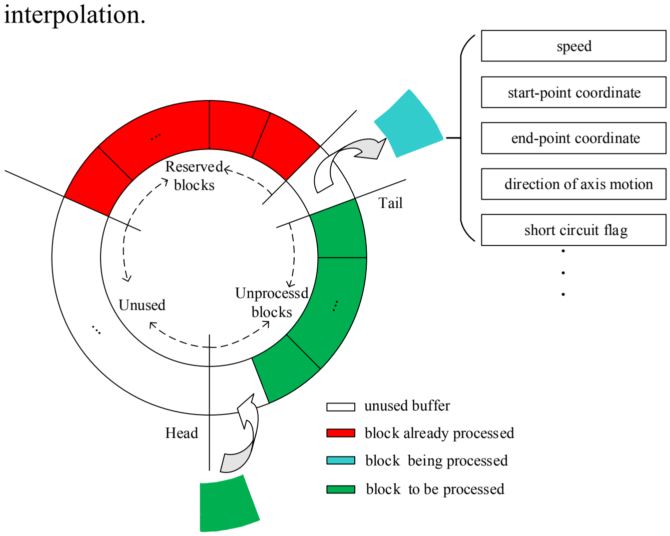

Fig. 4 Loop buffer.

C. Fine Interpolation

The interpolation block cached in the loop buffer by the method proposed in sections A and B, is actually a data structure, which stores the required parameter information for fine interpolation, including feedrate, start-point coordinate, end-point coordinate, orientation of each axis and short-circuit flag, etc. as shown in Figure 4. This step is to fine interpolate the block one by one.

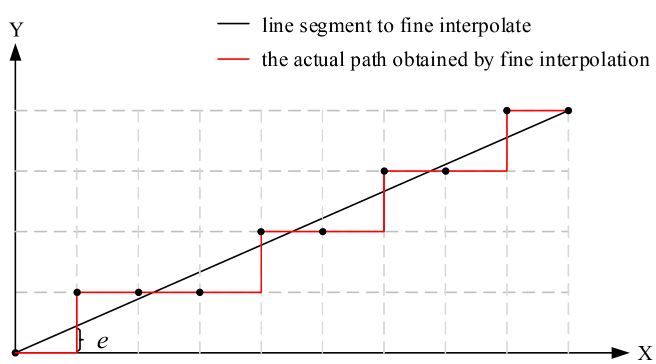

Fig. 5 Bresenham linear interpolation algorithm.

Fine interpolation is usually accomplished by CNC system hardware. By controlling the feed of each axis, the tool can move according to the predetermined trajectory. In this paper, the classical impulse incremental interpolation algorithm—Bresenham linear interpolation algorithm is used as fine interpolation algorithm[14].

Mapping the motion range of the machine tool into a mesh region as shown in Figure 5. The length of a grid is a pulse equivalent, and the tool can only move to the position of the grid node. Pulse equivalent is the basic unit of axis motion, and the movement of tool along each axis can only be an integer multiple of the pulse equivalent. Using the given information in the interpolation block to calculate the motion of each axis (the number of feed pulse equivalents), and choose the axis with large feed pulse equivalents as the long-axis. When a pulse occurs, the short-axis refers to the long-axis to decide whether to feed or not. The specific feeding strategy is as follows:

Step1: Let the error term e = 0 and calculate the slope k, so that $e = e + k$;

Step2: If e is less than 0.5, then X axis feeds one step, Y axis does not feeds, otherwise X feeds one step, Y feeds one step, and $e = e - 1$; when $x > x_1$, turn to Step4;

Step3: $e = e + k$, turn to Step2;

Step4: End.

D. Short Circuit Turn Back

EDM corrodes the workpiece by discharging. So short circuit will inevitably occur between the electrodes and the workpiece in the process of machining. In the case of short circuit, the electrodes need to turn back quickly until the discharge is recovered. The reserved area in the loop buffer is designed for this purpose, which effectively avoids repetitive interpolation calculation.

Short circuit occurs in the fine interpolation stage, so there are two situations: first, short circuit occurs when the current interpolation block is not completed; second, short circuit occurs when entering the next interpolation block. In actual machining, the first case is more common than the second case, and it is more complex to deal with. Because there is only complete interpolation block information in the loop buffer, and has no necessary information for the turn back of incomplete interpolation block. So, we need to build a new interpolation block to record the turn back information of the incomplete blocks, including the number of steps has completed and the current coordinates of the tool .

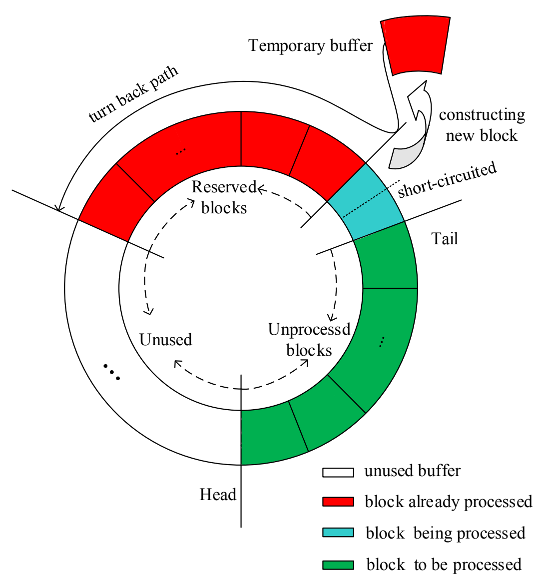

Fig. 6 Turn back interpolation for short circuit.

As shown in the Figure 6, a temporary buffer is set in the loop buffer. When a short circuit occurs and the fine interpolation is not completed, cache the completed part of the current interpolation block into the temporary buffer firstly, and then take the interpolation blocks from the loop buffer to interpolate back according to the turn back path shown in the Figure 6. Turn back interpolation only needs to reverse the orientation of the axis in the block and then perform the fine interpolation. Stopping machining and find out the cause if the discharge state is still in short circuit when all the interpolation blocks in the reserved area are turned back. Otherwise, the electrode will move from the current position to the position where the short circuit occurs after the discharge recovered.

IV. Experiment

In order to verify the efficiency of the algorithm, the traditional pulse incremental interpolation algorithm and the NURBS interpolation algorithm based on chord error control model proposed in this paper are compared on the MATLAB platform. The curve used in the experiment is a third order NURBS curve with the parameters information shown in the table1.

TABLE I
Parameters Of a NURBS Curve

Parameter	Value
Control points	(10,0), (20,20), (12,8), (10,20), (8,8), (0,20), (10,0)
Node vector	0.0, 0.0, 0.0, 0.0, 0.25, 0.5, 0.75, 1.0, 1.0, 1.0, 1.0
Weight vector	1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0

The NURBS curve has seven control points with equal weight, and the node vector is obtained according to the distribution of control points. Figure 7 shows the curve drawn by MATLAB.

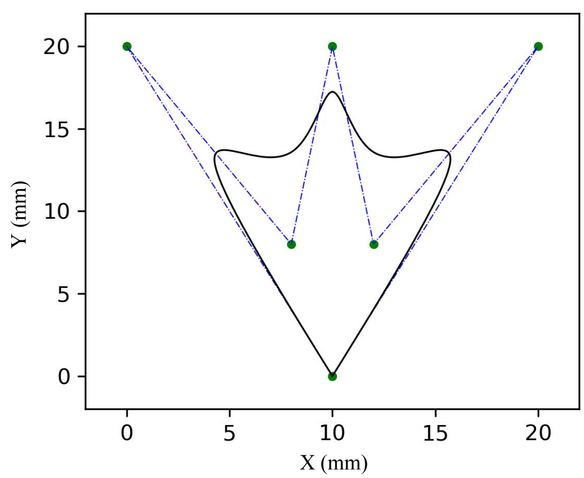

Fig. 7 Experimental NURBS curve.

In order to verify the interpolation algorithm proposed in this paper can effectively reduce the scale of G code and make the chord error smaller than the traditional EDM interpolation algorithm, a set of comparative simulation experiments are designed.

Experiment 1 uses the traditional impulse incremental interpolation algorithm. Firstly, using the CAM software to discretize the curve into a series of small segments, these segments is stored in a text file in the form of G code. Then, parsing the G code file to get the coordinates of interpolation points. The interpolated curve is formed by these points. As shown in Figure 8, the red curve is the interpolated curve and the black curve is the origin NURBS curve.

Experiment 2 uses the interpolation algorithm proposed in this paper. Compared with experiment 1, it does not need to be pre-processed by CAM software. The G code of the NURBS interpolation is customized:

G38 O3 C7 X10 Y0
G38 X20 Y20
G38 X12 Y8
G38 X10 Y20
G38 X8 Y8
G38 X0 Y20
G38 X10 Y0

G38 represents NURBS interpolation, O3 represents third-order curve, C7 represents the number of control points is 7, X10 and Y0 represents the coordinates of control points respectively. Only 7 lines of G code are needed to describe the interpolation information of the curve used in the experiment. The interpolated curve obtained in Experiment 2 is shown in Figure 9.

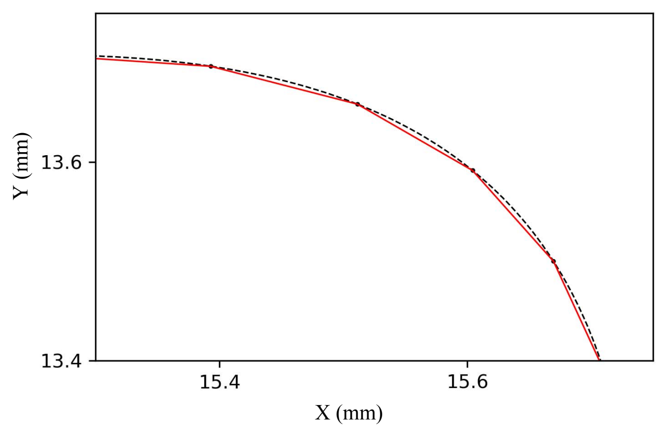

Fig. 8 Local curve obtained by Experiment 1.

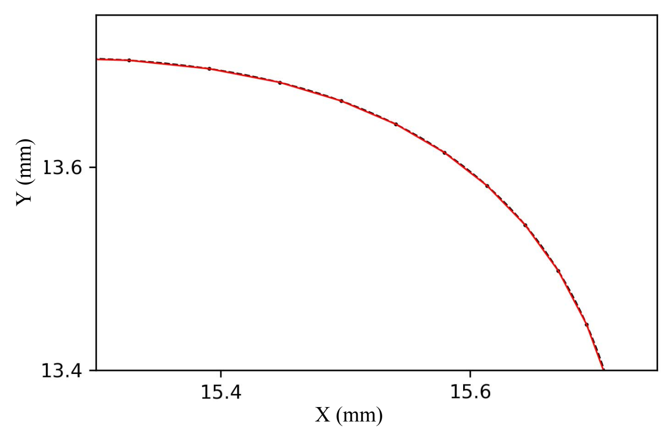

Fig. 9 Local curve obtained by Experiment 2.

By analyzing the experimental results of Figure 8 and Figure 9, it can be concluded that the interpolation points obtained in Experiment 1 are more dense in the part of curve with larger curvature, and more sparse in the part with smaller curvature. In order to analyze the interpolation errors, the chord error curves of the interpolated curve obtained by the above two interpolation algorithms are plotted as shown in Figure 10. The red curve is the error curve obtained in Experiment 1 and the blue curve is the error obtained in Experiment 2. The chord error of Experiment 2 is smaller, but the number of total interpolation points is almost the same. In addition, Experiment 1 needs about 300 lines of G code, while Experiment 2 only uses 7 lines of G code, which greatly reduces the number of G code and improves the interpolation efficiency.

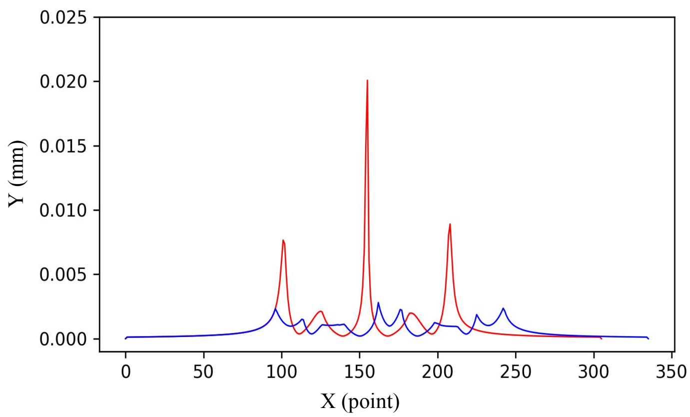

Fig. 10 Comparisons of chord error between two algorithms.

V. Conclusion

In order to provide an easy approach for processing NURBS curves with high machining performance by EDM, a real-time NURBS interpolation algorithm with limited chord error is developed in this paper. It consists of two steps, the first-stage discretes the curve into several small segments and cache in a ring buffer; the second-stage executes the fine interpolations for those segments, and if the short circuit occur, turn back interpolation can be achieved by scheduling the loop buffer. Finally, the simulation experiment is designed. and the experimental results show that the interpolation algorithm proposed in this paper needs to generate less G code and make the chord length error smaller than the traditional algorithm.

References

[1] Kunieda M, Lauwers B, Rajurkar KP, Schumacher BM, "Advancing EDM through fundamental insight into the process". *Annals of the CIRP*, vol. 54, no. 2, pp. 599-622, 2005.

[2] Salah NB, Ghanem F, Atig KB, "Numerical study of thermalaspects of electric discharge machining process". Int J Mach Tool Manufact, vol. 46, no. 7-8, pp. 908-911, 2006.

[3] Masory O, Koren Y, "Reference—word circular interpolators for CNC systems", ASME J Eng Ind, vol. 104, no. 11, pp. 400-405, 1982.

[4] Qin KH, Bin HZ, "Three-point recursion interpolation theory and algorithm of space circular arcs in a CNC system", *Comput Ind* , vol. 15, no. 4, pp. 355-362, 1990.

[5] Chen Mo, Chen Hao, Zhao Wansheng, "Four-axes Drived Unit Arc Length Increment Method for WEDM", *Electromachining & Mould*, no. 4, pp. 13-17, 2014.

[6] Chen Mo, Chen Hao, Xi Xuecheng, "Generalized Unit Arc Length Increment Method for Six-axis Electro-discharge Machining", *Electromachining & Mould*, no. 3, pp. 1-7, 2015.

[7] Lin RS, "Real-time surface interpolator for 3-D parametric surface machining on 3-axis machine tools". Int J Mach Tool Manufact, vol. 40, no. 10, pp. 1513-1526, 2000.

[8] Koren Y, Lo CC, Shpitalni M, "CNC interpolators: algorithms and analysis". ASME Winter Annu Meet, no. 64, pp. 83-92, 1993.

[9] Huang Haipeng, Chi Guanxin, Wang zhenlong,"Reversible Interpolation of Multi-axis EDM-CNC System", Electromachining & Mould, no. 1, pp. 4-8, 2009.

[10] Wang Cheng, Luo Fuyuan, Li Xuan, "Study on Reversible Interpolation Algorithm of Wire Electrical-discharge Machining", Electromachining & Mould, no. 15, pp. 13-16, 2015.

[11] Du DS, Liu YD, Yan CL, Li CX, "An accurate adaptive parametric curve interpolator for NURBS curve interpolation". Int J Adv Manufact Technol, vol. 32, no. 9-10, pp. 999-108, 2007.

[12] YANG DCH, "Parametric interpolator versus linear interpolator for precision CNC machining. Computer-Aided Design, vol. 26, no. 3, pp. 225-234, 1994.

[13] WANG Yunsen, YANG Dongsheng, LIU Yin-zhong "Velocity planning and parameter calculating in NURBS interpolation", Computer Integrated Manufacturing System, vol. 20, no. 8, pp. 1896-1912, 2014.

[14] Haque A U , Rahman M S , Bakht M , "Drawing lines by uniform packing", Computers and Graphics, vol. 30, no. 2, pp. 207-212, 2006.