IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 30, 2019, accepted July 29, 2019, date of publication August 13, 2019, date of current version August 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934977

A Real-Time Feedrate Planning Method and Efficient Interpolator With Minimal Feedrate Fluctuation for Parametric Toolpath

JIALI JIANG[1], FUYAN LIN[1], YONG ZHANG[2], HUI ZHANG[2], AND PEIQING YE[2]
[1]Department of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing 100083, China
[2]Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China

Corresponding authors: Fuyan Lin (linfy@cumtb.edu.cn), Yong Zhang (zhangy10@126.com), and Hui Zhang (wwjj@tsinghua.edu.cn)

This work was supported by the National Science and Technology Major Project under Grant 2017ZX04010001.

ABSTRACT Different from linear and circular interpolation, feedrate fluctuation is inevitable for parametric toolpath, which is caused by the nonlinear relation between spline parameters and arc-length. In order to improve the accuracy and efficiency of real-time interpolation algorithm with minimal feedrate fluctuation, this thesis presents an iterative feedback interpolator to restrain the feedrate fluctuation. Estimator and corrector stage are two essential steps of our proposed method. The result of the first-order Taylor expansion is used for the iterative initial value in the estimator stage. Thereafter, based on Steffensen iterative accelerator method, the feedback of current feedrate and command feedrate will control to update parameters value repeatedly until feedrate fluctuation rate reaches the allowable value in the corrector stage. By means of the numerical method, it is proved that this iteration method can satisfy the high-order convergence and fast convergence theoretically. In the simulation, the third-degree B-spline curve is utilized to test the proposed method which is equipped with obvious effects on restraining the feedrate fluctuation and improving computation efficiency compared with four existing methods.

INDEX TERMS Parametric toolpath, feedrate fluctuation, feedrate planning, CNC machining, iterative feedback interpolator.

I. INTRODUCTION

With the development of CAD/CAM technology, traditional linear and circular interpolation cannot meet the requirements of modern CNC machining in grinding apparatus, aviation and other fields [1]–[4]. In order to process complex surfaces, parametric toolpath has played an important role in altering the traditional interpolation methods of small straight-line segment approximating parametric curves. Straightway interpolation parametric toolpath can prominently increase machining efficiency and machining quality [5]. In addition, parametric interpolator can prolong the service life of the motors and reduce the burden of data storage and computation for the hardware.

The spline curve is a presentation of the parametric toolpath, which is usually determined by a group of control points. B-spline curves and NURBS curves are widely used to describe complicated toolpath in the numerical control system [6]. The reason why spline toolpath interpolation has always been a research hotspot is that NURBS curves can provide a universal mathematical presentation form for free curves and surfaces and the NURBS curve is the only method to exchange data between CAD/CAM and CNC system [7].

There are several methods proposed for spline interpolation. However, an inaccurate mapping between arc length of spline toolpath and curve parameters during interpolation inevitably leads to truncation error, which will definitely result in the discrepancy between the desired feedrate and the actual feedrate command and stirs up the feedrate fluctuation in current interpolation period. Moreover, the feedrate fluctuation is the main factor to affect the process quality of the surfaces. So it is the key to improve the accuracy and efficiency of real-time interpolation algorithm with minimal feedrate fluctuation.

In the past research, scholars have developed many methods to improve the accuracy of spline curve parameter calculation. These interpolation parameters can be roughly divided into three types with Taylor Expansion (TE), Polynomial

The associate editor coordinating the review of this article and approving it for publication was Shuaihu Li.

Fitting (PF) and Predictor-Corrector (PCI). Bedi [8] proposed the uniform parameters interpolation method at the earliest. This method utilizes constant micro parameters to calculate the interpolation parameters in the next interpolation period, which causes the unequal chord during each interpolation period. This could rise the feedrate fluctuation obviously. Huang [9] and Koren [10] made use of the first and second order time-based Taylor expansion (TTE) to confirm the parametric value of the approximate curve under constant feedrate, but the truncation error still exists. It is the simple form of this method that satisfies normal requirements, so TTE method could be frequently chosen. However, as the expansion order increases, feedrate fluctuation will decline, which brings large amount of calculation. In order to find a solution to the above problems, Farouki [11] chose first order time-based Taylor expansion and amended parameter, which could make the fluctuation cut down a little. Afterwards, Yeh and Hsu [12] also came up with an interpolation method to control the feedrate by adding the compensation to reduce the fluctuation. In the same year, Tsai and Cheng [13] introduced Runge-Kutta interpolation method and compared different interpolation algorithms on the feedrate fluctuation and computing time. While Runge-Kutta interpolation method depends on Taylor series to avoid higher-order derivative calculus, its interpolation accuracy is higher than the accuracy of the second-Taylor formula expansion. All methods mentioned are based on Taylor formula expansion, which are utilized to diminish feedrate fluctuation by means of compensating error of calculation.

The core of the polynomial fitting method is to calculate the corresponding relation between the parameter and arc length and form the boundary at some points. Afterwards, piecewise polynomials are chosen to fit the relation on the basis of boundary condition. Erkorkmaz and Altintas [14] adopted quintic polynomial to interpolate with feed correction. Liu [15] solved the quartic equation to eliminate the feedrate fluctuation of secondary non-uniform rational B-spline curves and deduced to higher order spline. Heng and Erkorkmaz [16] applied the seventh order polynomial to replace the quantic spline for polynomial wiggle. In the paper [17]–[19], the polynomial mapping method was used to improve the accuracy of arc length estimation and curve fitting with minimal feedrate fluctuation. However, the approximation error will accumulate along the parametric toolpath, especially for the curve with large curvature variation or uneven parametrization. Furthermore, in order to compensate the error in the form of closed-loop, some other researchers tried to combine iterative evaluation theory into interpolation parameters calculation. It is such a feedback interpolation method [20]–[22] in which the deviation of the desired feedrate and the scheduled feedrate could become the restricted condition of iterative update parameters. Lo [23] came up with using feedback to calculate interpolation parameters for the first time. Lately, Cheng [24], [25] improved this method and proposed a real-time algorithm based on predictor-corrector interpolator (PCI). In the predictor stage, a simple algorithm is used to estimate the servo command at the next time. In the corrector stage, the feedback of current feed and feed command controls to update parameters value. On this basis, Lee [26] used Newton's method to update parameters repeatedly until feedrate fluctuation rate reached the allowable value. Zhao [27] thought feedrate fluctuation is caused by interpolator and servo controller. In the interpolation, parameters approximation and trajectory deviation caused fluctuation, after that he presented chord-tracing algorithm (CTA) to eliminate feedrate fluctuation further. In CTA interpolation process, an initial value of parameter is evaluated by second-order Taylor expansion and Newton Iterative method is used in feedback correction scheme. In contrast to PCI, CTA has higher convergence rate and steadier computation. Therefore, it could meet the requirements of real-time interpolation, especially at sharp corners.

In this research paper, an iterative feedback interpolation algorithm was proposed to eliminate the feedrate fluctuation caused by the nonlinear relationship between curve parameters and arc length in the interpolation process. By means of the numerical method, the corresponding constraints could ensure the machining accuracy. Furthermore, the theoretical proof method is also presented, which proves that the interpolation method can satisfy the high-order convergence and fast convergence. The feedback iterative method is adopted in simulation experiments of parametric interpolator.

The remainder of this paper is organized as follows. Spline curves matrix expression and calculation are presented in Section 2. A kind of feedrate planning method of parameter toolpath is recommended in Section 3. The smooth transfer along a continuous segment of parameter toolpath is introduced in Section 4. In Section 5, the iterative feedback interpolation algorithm with the minimal feedrate fluctuation is presented and its convergence has been proved. In Section 6, the iterative feedback interpolation algorithm mentioned is compared with Taylor expansion method, Newton iterative method and simple Newton iterative method in feedrate fluctuation and computational efficiency in simulation experiments. The conclusions are given in Section 7.

II. MATRIX PRESENTATION OF THE PARAMETRIC TOOLPATH

Parametric toolpath becomes more and more popular in CNC machining and it mainly includes two types: polynomial curves and spline curves. However, in practice, when toolpath is described by polynomial, the corresponding expression is much complex and the toolpath cannot be adjusted partly [28]. To overcome these defects, B-spline and NURBS curves are used to describe complicated toolpath in CNC systems. Since NURBS curves can construct CAD modelling, its performance is better than B-spline curves [29]. In the following, the matrix expression of NURBS curves could be obtained from the recursive definition of B-spline curves and the basic theory of NURBS curves.

A third degree NURBS curve can be defined as follows:

$$p(u) = \frac{\sum_{i=0}^{n} \omega_i d_i N_{i,3}}{\sum_{i=0}^{n} \omega_i N_{i,3}} \tag{1}$$

where $\omega_i (i = 0, 1, \ldots, n)$ is the weight factor, which is corresponding to control points $d_i (i = 0, 1, \ldots, n)$. The first and the last weight factor ω_0,ω_n meet ω_0,$\omega_n > 0$ respectively, and others meet $\omega_i \geq 0$. $\{d_i = (x_i, y_i, z_i) | i = 0, 1, \ldots, n\}$ is the control points range. For convenience, $N_{i,k}$ is B-spline basis function in this section, and they are recursively defined as follows [30]:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & u_i \leq u \leq u_{i+1} \\ 0 & others \end{cases} \\ N_{i,k}(u) = \dfrac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) \\ \qquad + \dfrac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \\ define : \frac{0}{0} = 0 \end{cases} \tag{2}$$

One subscript i of $N_{i,k}(\mu)$ stands for the sequence number, and the other parameter k stands for the degree of the toolpath. According to B-spline basis function, nonzero value $N_{i-3,3}$, $N_{i-2,3}$, $N_{i-1,3}$, $N_{i,3}$ would be obtained in $\mu \in [\mu_i, \mu_{i+1}]$ as follows:

$$\begin{aligned} N_{i,3} &= \frac{(u - u_i)^3}{(u_{i+3} - u_i)(u_{i+2} - u_i)(u_{i+1} - u_i)} \\ N_{i-1,3} &= \frac{(u - u_{i-1})^2 (u_{i+1} - u)}{(u_{i+2} - u_{i-1})(u_{i+1} - u_{i-1})(u_{i+1} - u_i)} \\ &\quad + \frac{(u - u_{i-1})(u_{i+2} - u)(u - u_i)}{(u_{i+2} - u_{i-1})(u_{i+2} - u_i)(u_{i+1} - u_i)} \\ &\quad + \frac{(u_{i+3} - u)(u - u_i)^2}{(u_{i+3} - u_i)(u_{i+2} - u_i)(u_{i+1} - u_i)} \\ N_{i-2,3} &= \frac{(u - u_{i-2})(u_{i+1} - u)^2}{(u_{i+1} - u_{i-2})(u_{i+1} - u_{i-1})(u_{i+1} - u_i)} \\ &\quad + \frac{(u - u_{i-1})(u_{i+2} - u)(u_{i+1} - u)}{(u_{i+2} - u_{i-1})(u_{i+1} - u_{i-1})(u_{i+1} - u_i)} \\ &\quad + \frac{(u_{i+2} - u)^2 (u - u_i)}{(u_{i+2} - u_{i-1})(u_{i+2} - u_i)(u_{i+1} - u_i)} \\ N_{i-3,3} &= \frac{(u_{i+1} - u)^3}{(u_{i+1} - u_{i-2})(u_{i+1} - u_{i-1})(u_{i+1} - u_i)} \end{aligned} \tag{3}$$

where $\nabla_i = u_{i+1} - u_i$, $\nabla_i^2 = u_{i+2} - u_i$, $t = u - u_i / (u_{i+1} - u) \in [0, 1]$, $m_{11} = (\nabla_i)^2 / \nabla_{i-2}^3 \nabla_{i-1}^2$, $m_{212} = \nabla_{i-1} \nabla_i^2 / \nabla_{i-1}^3 \nabla_{i-1}^2$, $m_{222} = 1/\nabla_{i-1}^3$, $m_{444} = (\nabla_i)^2 / \nabla_i^3 \nabla_i^2$, $m_{423} = (\nabla_i)^2 / \nabla_{i-1}^3 \nabla_i^2$, $\Delta = 1/\nabla_i$, the matrix of third-degree NURBS curve is obtained as follows:

$$p_i(t) = \frac{[1, t, t^2, t^3] M_i \begin{bmatrix} \omega_{i-3} d_{i-3} \\ \omega_{i-2} d_{i-2} \\ \omega_{i-1} d_{i-1} \\ \omega_i d_i \end{bmatrix}}{[1, t, t^2, t^3] M_i \begin{bmatrix} \omega_{i-3} \\ \omega_{i-2} \\ \omega_{i-1} \\ \omega_i \end{bmatrix}} \tag{4}$$

Coefficient matrix M_i can be obtained by formula (5), as shown at the bottom of this page. The expression of parameters is assumed as formula (6).

$$\begin{cases} a = m_{11}\omega_{i-3}d_{i-3} + m_{12}\omega_{i-2}d_{i-2} + m_{13}\omega_{i-1}d_{i-1} \\ \quad + m_{14}\omega_i d_i \\ b = m_{21}\omega_{i-3}d_{i-3} + m_{22}\omega_{i-2}d_{i-2} + m_{23}\omega_{i-1}d_{i-1} \\ \quad + m_{24}\omega_i d_i \\ c = m_{31}\omega_{i-3}d_{i-3} + m_{32}\omega_{i-2}d_{i-2} + m_{33}\omega_{i-1}d_{i-1} \\ \quad + m_{34}\omega_i d_i \\ e = m_{41}\omega_{i-3}d_{i-3} + m_{42}\omega_{i-2}d_{i-2} + m_{43}\omega_{i-1}d_{i-1} \\ \quad + m_{44}\omega_i d_i \end{cases}$$
$$\begin{cases} a_1 = m_{11}\omega_{i-3} + m_{12}\omega_{i-2} + m_{13}\omega_{i-1} + m_{14}\omega_i \\ b_1 = m_{21}\omega_{i-3} + m_{22}\omega_{i-2} + m_{23}\omega_{i-1} + m_{24}\omega_i \\ c_1 = m_{31}\omega_{i-3} + m_{32}\omega_{i-2} + m_{33}\omega_{i-1} + m_{34}\omega_i \\ e_1 = m_{41}\omega_{i-3} + m_{42}\omega_{i-2} + m_{43}\omega_{i-1} + m_{44}\omega_i \end{cases} \tag{6}$$

Substituting formula (6) into (3), the NURBS toolpath is simplified as:

$$p_i(t) = \frac{a + bt + ct^2 + et^3}{a_1 + b_1 t + c_1 t^2 + e_1 t^3} \tag{7}$$

$$\begin{aligned} M_i &= \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \\ &= \begin{bmatrix} m_{11} & \nabla_{i-2}^2 \Delta m_{11} + m_{212} & \frac{\nabla_{i-1}}{\nabla_i^2} m_{212} & 0 \\ -3m_{11} & \left(1 - 2\nabla_{i-2}^2 \Delta\right) m_{11} + \left(\frac{\nabla_i}{\nabla_{i-1}} - \frac{\nabla_i}{\nabla_i^2} - 1\right) m_{212} + \nabla_i^2 m_{222} & \nabla_{i-1} m_{222} + \left(\frac{2\nabla_i - \nabla_{i-1}}{\nabla_i^2}\right) m_{212} & 0 \\ 3m_{11} & \left(\nabla_{i-2}^2 \Delta - 2\right) m_{11} + \left(\frac{\nabla_i}{\nabla_i^2} - \frac{\nabla_i}{\nabla_{i-1}} - \frac{1}{\nabla_{i-1} \nabla_i^2 \Delta^2}\right) m_{212} - 2\nabla_i m_{222} & \left(\frac{(\nabla_i)^2}{\nabla_{i-1} \nabla_i^2} - \frac{2\nabla_i}{\nabla_i^2}\right) m_{212} + \left(\nabla_i - \frac{\nabla_{i-1} \nabla_i}{\nabla_i^2}\right) m_{222} + \Delta \nabla_i^3 m_{444} & 0 \\ -m_{11} & m_{11} + \frac{(\nabla_i)^2}{\nabla_{i-1} \nabla_i^2} m_{212} + m_{423} & -\frac{(\nabla_i)^2}{\nabla_{i-1} \nabla_i^2} m_{212} - m_{423} - m_{444} & m_{444} \end{bmatrix} \end{aligned} \tag{5}$$

TABLE 1. Common ways to determine the length of units.

Method	Algorithm	Comments
Feedrate command dynamic method	$L_u = FT \cdot n$	L_u : the length of units T : interpolation period F : feedrate command n : security coefficient
Fixed unit length method	$L_u = m$	m : fixed value
Real-time planning unit method	$\begin{cases} F \geq \dfrac{L_{u\_min} \times 10^5}{1.0061 k_s} \\ L_u = 1.0061 \times 10^{-5} F \cdot k_s \end{cases}$ $else: L_u = L_{u\_min}$	L_{u_min} : the minimum unit length. $k_s > 1$: the security coefficient. $a_{max} = 5000mm/s^2$ $T = 4ms$
Curvature-based planning unit method	$L_u = [u_s, u_i]$	$[u_s, u_e]$: points vector interval u_i : correspond with $V_i - V_{i-1} \leq \Delta V$

where a, b, c, e, a_1, b_1, c_1, e_1 is determined by NURBS structure parameters. Since derivation is used to calculate length and curvature of NURBS curves, it could be easier by adopting matrix to express the first and second derivatives, which increases the calculation efficiency in the real-time interpolation.

III. A REAL-TIME FEEDRATE SCHEDULING WITH JERK LIMITED PROFILE

There are some factors to be avoided in CNC machining, such as vibration and excess error. Therefore, machine tool kinematics constraints, machining path geometric constraints and machining error constraints need to be overall considered [31]. In addition, corresponding motion algorithms should be designed to control acceleration and deceleration, which ensures machining accuracy and efficiency. Taking NURBS curves as an example, the feedrate planning of parameter toolpath could be shown in the following. Since the curvature is complex and unpredictable along the NURBS toolpath, the feedrate vector is variable and the acceleration is much difficult to control in machining [32], [33]. Considering the different curvature characteristics on the different trajectories, the feedrate planning algorithm needs to confirm the suitable feedrate on each segment as follows:

In the proposed feedrate planning method, a whole NURBS curve is divided into several planning units. Theoretically, the length of units could be determined in random, however, the number and length of units would affect the machining efficiency. Therefore, the following table provides the way to determine the length of units specifically.

The four ways have their advantages and disadvantages. The feedrate command dynamic method lies in avoiding cross-unit interpolation and simplifying the interpolation algorithm. The fixed unit length method needs more attention. When the feedrate command is high, cross-unit interpolation could be emergent. The real-time planning unit method is used in certain conditions according to the experimental fitting algorithm of meeting the real-time requirements, but the process of planning is complex. The curvature-based planning unit method can improve the machining efficiency and ensure the machining quality when the curvature changes a little.

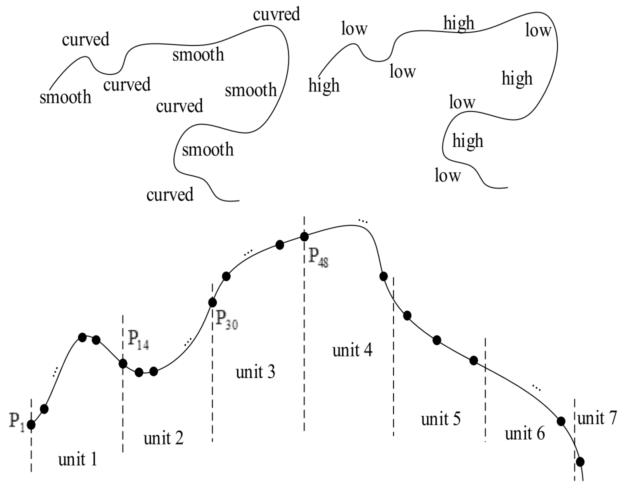

FIGURE 1. Trajectory curvature and feedrate characteristics.

After dividing the NURBS curve into planning units by above methods, the feedrate limitation needs to be planned, which is determined by the accuracy and acceleration [34]. At present, chord is widely used to describe the error of machining accuracy, and we used parametric toolpath segment of arc-length approximation as an example, which is shown in Figure 2.

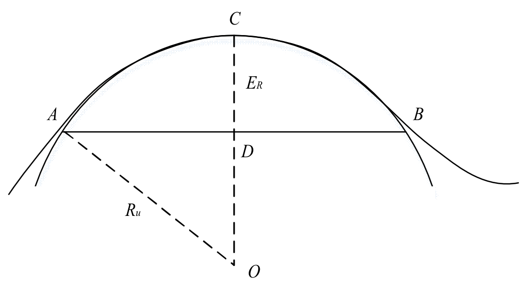

FIGURE 2. Approximate arc interpolation schematic diagram.

In Figure 2, AB is an interpolation step-length of a segment of NURBS curves. The maximum distance between each point on the curve to AB is the chord error. O shows the circle center. κ_u and R_u means curvature and curvature radius respectively. E_R is the maximum permissible error in the machining process.

$$R_u^2 = (\frac{V_R T}{2})^2 + (R_u - E_R)^2 \tag{8}$$

Considering special circumstances, the feedrate limitation related to the accuracy can be obtained as follows:

$$\begin{cases} V_R = \dfrac{2R_u}{T}\sqrt{1-(1-\frac{E_R}{R_u})^2} & R_u > E_R \\ V_R = \dfrac{2R_u}{T} & E_R \geq R_u \\ V_R = +\infty & \kappa = 0 \end{cases} \tag{9}$$

When feedrate vector has changed, the feedrate will be limited by acceleration in formula (10).

$$\begin{cases} V_A = \sqrt{a_{\max} R_u} & \kappa \neq 0 \\ V_A = +\infty & \kappa = 0 \end{cases} \tag{10}$$

F is the feedrate command. Under the circumstance of the restraint by the acceleration and the chord error, the minimal feedrate V_c could be obtained in a planning unit as follows:

$$V_c = \min\{V_R, V_A, F\} \tag{11}$$

At present, there is one NURBS curve divided into several planning units, whose limited feedrate has been determined. And these planning units are connected sequentially from the first to the last. V_{cn} stands for the limited feedrate of the nth unit. The endpoints of each planning unit are called transfer points. If the transfer point B is connected smoothly and the direction of the feedrate vector has not changed, the feedrate of point B is expressed according to formula (12).

$$\begin{cases} V_B = \min\left\{V_{cn}, V_{c(n+1)}\right\} & \text{others} \\ V_B = 0 & \text{B is the endpoint} \\ & \text{of NURBS curve} \end{cases} \tag{12}$$

The feedrate of endpoint has been determined, the acceleration or deceleration will not exceed the limited feedrate in a unit. If certain unit appears $|V_k^2 - V_{k-1}^2|/2a_{\max} > L_{uk}$, the transfer points have to be adjusted, where L_{uk} is the length of the kth unit, $a_{\max}$ shows the maximum acceleration, V_{k-1} and V_k represent the feedrate of two neighbor transfer points respectively. It illustrates that feedrate could not reach the transfer feedrate in the allowable acceleration range of machine tool. Then the adjustment is formula (13).

$$\begin{cases} \begin{cases} V_k = \sqrt{2a_{\max}L_{uk} + V_{k-1}^2} \\ V_{k-1} = V_{k-1} \end{cases} & V_k > V_{k-1} \\ \begin{cases} V_{k-1} = \sqrt{2a_{\max}L_{uk} + V_k^2} \\ V_k = V_k \end{cases} & V_k < V_{k-1} \end{cases} \tag{13}$$

The second adjustment of the transfer point feedrate starts from the starting point of the NURBS curve $V_0 = 0$ and the adjustment proceeds sequentially. If the starting point feedrate of a unit is changed, the transfer point feedrate needs to be recalculated in reverse. The calculation will carry on until the transfer point feedrate of the whole curve has been determined. Afterwards, output parameters of each feedrate planning unit include control points, weight factors, node vector interval and the feedrate of the transfer points, which would prepare for interpolation.

TABLE 2. Type value points information.

x	0	10	20	30	40
y	0	10	0	10	0

IV. SMOOTH CONNECTION BETWEEN PARAMETRIC TOOLPATH

The matrix expression of NURBS curves and the look-ahead feedrate planning of the trajectory based on planning units are introduced in previous section. In the expression of complex curves, B-spline curves could not accurately represent circular arc and other quadric curves. But B-spline is easier to be described and calculated for most of free-form curves [35].

Taking third-degree B-spline curves as an example, its motion control algorithm is also divided into two parts: the feedrate planning of the toolpath based on planning units and parametric interpolation. Contrary to NURBS curves, B-spline curves need to read the information of type value points rather than control points in feedrate scheduling module, so we need to do inverse calculation firstly and construct the spline curves by control points.

In order to achieve real-time interpolation, the feedrate planning module of third-degree B-spline needs to read the type value points by groups. However, the two curves which are generated from two groups of type value points in sequence are G^0 continuous. Therefore, direct interpolation will not only generate errors, but also lead to direction jump of feedrate at connection points. Furthermore, the acceleration is difficult to control. For instance, the following table is the sequence of type value points in the x-y plane.

At first, the third-degree B-spline curve 1 is interpolated by five type value points. Then, the curve 2 is interpolated by the first three type value points. The curve 3 is interpolated by the last three type value points as shown.

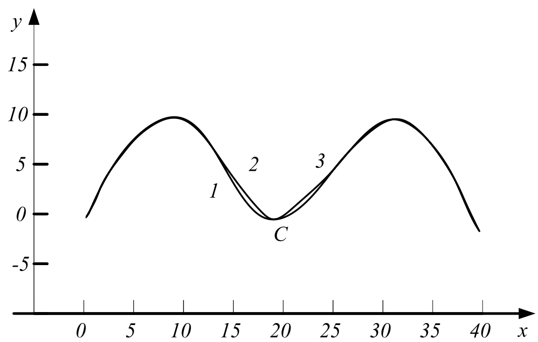

FIGURE 3. Normal connection between two curves.

As shown in Figure 3, it is obvious that curve 1 is smoother and the curve made up of 2 and 3 is continuous but not smooth at the point C. Besides, there is considerable error between curve 1 and the curve 2 and 3. The specific analysis is as following:

Assuming read the sequence of type value points $\{P_1P_2\ldots P_n\}$, the sequence of control points $\{C_0\ C_1\ \cdots\ C_{n+1}\}$ could be calculated by the inverse formula. If $C_n = C_{n+1}$ is boundary condition, the derivative of the end point on the first curve is calculated from $\overrightarrow{C_{n-1}C_n}$. Reading the second sequence of type value points $\left\{P_n\ P_{n+1}\ \cdots\ P_{n+m}\right\}$, the sequence of control points $\left\{C_{n-1}\ C_n\ \cdots\ C_{n+m+1}\right\}$ could be calculated. If $C_{n-1} = C_n$ is boundary condition, the derivative of the end point on the second curve is calculated from $\overrightarrow{C_n C_{n+1}}$. Therefore, the smooth connection of the two curves cannot be guaranteed.

Overlapping connection method is adopted in this section. Two B-spline curves are calculated successively and connected smoothly in interpolation. The main procedure is that the control points at connection are calculated repeatedly, and the reasonable part is selected to construct the curve to ensure the smooth transition at the connection point.

For the sequence of type value points $\{P_1\ P_2\ \ldots\ P_{N-1}\}$, $\{P_1\ P_2\ \ldots\ P_n\}$ is read firstly and the sequence of control points $\left\{C_0\ C_1\ \cdots\ C_{n+1}\right\}$ is calculated by the inverse formula. After reading $\{P_{n-m}\ P_{n-m+1}\ \ldots\ P_{2n-m-1}\}$, the sequence of control points $\{C_{n-m-1}\ C_{n-m}\ \ldots\ C_{2n-m}\}$ could be calculated, where $0 < m < n-1$, m is an odd number. Then, there are overlapping control points $\{C_{n-m}C_{n-m+1}\ \ldots C_n\}$. After sieving these control points, $\{C_{n-m}\ C_{n-m+1}\ \ldots\ C_{n-(m+1)/2}\}$ is retained and $\{C_{n-(m-1)/2}\ C_{n-(m-3)/2}\ \ldots\ C_{n+1}\}$ is abandoned in the first sequence. $\{C_{n-(m-1)/2}\ C_{n-(m-3)/2}\ \ldots\ C_n\}$ is retained and $\{C_{n-m-1}\ C_{n-m+1}\ \ldots\ C_{n-(m+1)/2}\}$ is abandoned in the second sequence. Finally, the new sequence of control points $\{C_0\ C_1\ \ldots\ C_{n-(m+1)/2}\}$ and $\{C_{n-(m+5)/2}\ C_{n-(m+3)/2}\cdots C_{2n-m}\}$ are constructed. The end point position of the first curve and the starting point position of the second curve are determined by $\{C_{n-(m+5)/2}\ C_{n-(m+3)/2}\ C_{n-(m+1)/2}\}$. Therefore, it is obvious that two B-spline curves determined by the new control point sequence are smoothly connected at the connection. Taking $m = 5$ as an example, Figure 4 is as following:

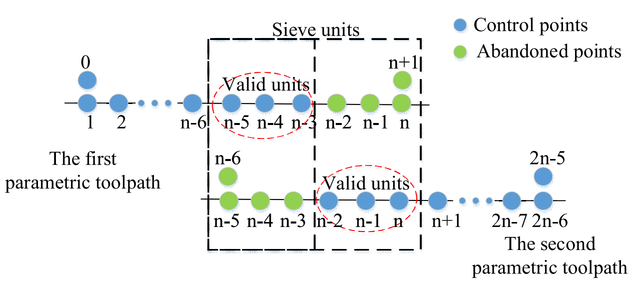

FIGURE 4. Schematic diagram of overlapping connection method.

As shown in Figure 4, when B-spline curves are planned, control points are retained in the ellipse and the others near the endpoints in the box are abandoned. The new control point sequence is $\{C_0\ C_1\ \ldots C_{n-3}\}$, which ensures that spline interpolation is real-time and smooth.

After constructing the third-degree uniform B-spline by overlapping joint method, we could complete the feedrate planning as the same as following flow of NURBS curves.

V. THE FEEDRATE FEEDBACK INTERPOLATION ALGORITHM OF STEFFENSEN ITERATIVE ACCELERATOR

The feedrate planning of parametric toolpath is presented above. The output units of feedrate planning are about to be interpolated in sequence. Different from linear interpolation, parametric toolpath interpolation leads to feedrate fluctuation due to the non-linear relationship between arc length of free curves and curve parameters [36]. In order to eliminate feedrate fluctuation during the interpolation, a feedrate feedback interpolation strategy based on Steffensen iterative accelerator is introduced in this section. In the following, the algorithm will be demonstrated theoretically, so as to show that the algorithm satisfies the high-order convergence and fast convergence, which will be fully proved in the numerical simulation.

Traditionally, Taylor expansion method is the most widely applied in interpolation, which could obtain the next interpolation parameter u_{i+1} and then calculate the actual interpolation points. However, because of the non-linear correspondence between the curve arc length and the parameters, there is deviation during parameters calculation by means of Taylor formula according to the chain derivation [30] as follows:

$$\dot{s} = \frac{ds}{dt} = ||\mathrm{r}'(u)|| \cdot \frac{d\hat{u}}{ds_d} \cdot \frac{ds_d}{dt} = ||\mathrm{r}'(u)|| \cdot \frac{d\hat{u}}{ds_d} \cdot \dot{s}_d \qquad (14)$$

where s is arc length of the curve, $\dot{s}$ is the actual feedrate value at the moment, and $\hat{u}$ is the curve parameter estimated from the numerical method. Because of the existing nonlinear mapping, $||\mathrm{r}'(u)|| \cdot (d\hat{u}/ds_d)$ is not equal to 1. Besides, in the formula (14), s_d is the arc length determined by expected feedrate at the moment, which means $s_d = v_k^* T_s$. Inevitably, when the curve is interpolated, there exists the problem of replacing arc length with chord length. As shown in the Figure 5 [27].

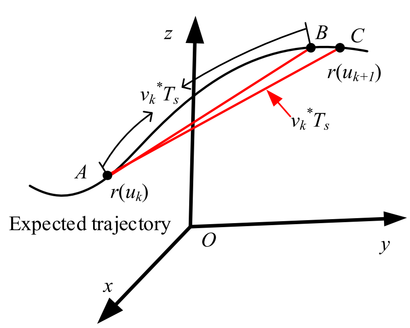

FIGURE 5. The problem of replacing arc length with chord length.

Due to the reasons above, it is inevitable to find the deviation between expected feedrate v_k^* and actual feedrate v_k, which could bring the feedrate fluctuation. The result is shown as formula (15). The actual feedrate could be calculated by the arc length and the interpolation period $v_k = |\mathrm{r}(u_{k+1}) - \mathrm{r}(u_k)| / T_s$.

$$\varepsilon = \frac{\left|v_k^* - v_k\right|}{v_k^*} \times 100\% \tag{15}$$

For minimal feedrate fluctuation, it only needs to find u_{k+1} to establish $f(u) = 0$ in formula (16). However, the theoretical solution of the formula is much complex.

$$f(u) = \|\mathrm{r}(u) - \mathrm{r}(u_k)\| / v_k^* - T_s = 0 \tag{16}$$

The Newton iterative method (CTA) is adopted in most numerical methods. The iteration process is given as formula (17). And the initial value of iteration can be obtained from the traditional Taylor formula.

$$\begin{aligned}
&do: \\
&t_i = t_{i-1} - \frac{f(t_{i-1})}{f'(t_{i-1})}, \quad i = 1, 2, 3 \cdots \\
&v_i = \|\mathrm{r}(t_i) - \mathrm{r}(u_k)\| / T_s \\
&until: \left|\frac{v_k^* - v_i}{v_k^*}\right| \leq \varepsilon \\
&then: u_{k+1} = t_i
\end{aligned} \tag{17}$$

where t_i is the parameter obtained at the ith iteration for calculating the curve parameter u_{k+1}, v_i is the estimated feedrate at the ith iteration. ε is the upper bound of the given feedrate fluctuation. If using a constant M to replace $f'(t_{i-1})$, simple Newton iterative method (SCTA) avoids the derivative calculation, but its convergence property is related to the selection of constant M. In addition, it could only ensure linear convergence and not improve calculating efficiency.

In order to get higher efficiency, the convergence order needs to be increased on the basis of ensuring the accuracy. Therefore, the Aitken accelerated iteration algorithm is introduced. If x_k is the predicted value of x^*, x_k is corrected by iterative formula for the first time as follows:

$$x_{k+1} = f(x_k) \tag{18}$$

Formula (19) could be obtained by the mean value theorem, and ξ is between x_k and x^*.

$$x_{k+1} - x^* = f(x_k) - f(x^*) = f'(\xi)(x_k - x^*) \tag{19}$$

If $f'(\xi)$ varies little in the interval, the estimated value is L. Formula (20) could be defined.

$$x_{k+1} - x^* = f'(\xi)(x_k - x^*) \approx L(x_k - x^*) \tag{20}$$

x_k is corrected again as follows:

$$\begin{aligned}
x_{k+2} &= f(x_{k+1}) \\
x_{k+2} - x^* &\approx L(x_{k+1} - x^*)
\end{aligned} \tag{21}$$

By integrating formula (20) and formula (21), we can obtain a formula as follows:

$$\frac{x_{k+1} - x^*}{x_{k+2} - x^*} \approx \frac{x_k - x^*}{x_{k+1} - x^*} \tag{22}$$

Then the formula above can be simplified as:

$$x^* \approx \frac{x_k x_{k+1} - (x_{k+2})^2}{x_{k+2} - 2x_{k+1} + x_k} = x_k - \frac{(x_{k+1} - x_k)^2}{x_{k+2} - 2x_{k+1} + x_k} \tag{23}$$

Generally, the right side of the formula is regarded as the new approximate value of x^*, which could be written as $\bar{x}_{k+1}$, so we can derive Aitken accelerated iteration expression in formula (24).

$$\bar{x}_{k+1} = x_k - \frac{(x_{k+1} - x_k)^2}{x_{k+2} - 2x_{k+1} + x_k} \tag{24}$$

Prove that the convergent rate of Aitken acceleration sequence $\{\bar{x}_k\}$ is faster than equivalent of original sequence $\{x_k\}$ in mathematical analysis as follows:

First, $e_k = x_k - x^*$ then,

$$\lim_{k \to \infty} \frac{e_{k+1}}{e_k} = c \tag{25}$$

Further,

$$\lim_{k \to \infty} \frac{e_{k+2}}{e_k} = \lim_{k \to \infty} \frac{e_{k+2}}{e_{k+1}} \cdot \frac{e_{k+1}}{e_k} = c^2 \tag{26}$$

According to the transformation of formula (24), we could obtain formula (27):

$$\begin{aligned}
\bar{x}_{k+1} - x^* &= x_k - x^* - \frac{(x_{k+1} - x_k)^2}{x_{k+2} - 2x_{k+1} + x_k} \\
&= x_k - x^* - \frac{(e_{k+1} - e_k)^2}{e_{k+2} - 2e_{k+1} + e_k}
\end{aligned} \tag{27}$$

$e_k = x_k - x^*$ is divided by both sides in formula (27), then:

$$\frac{\bar{x}_{k+1} - x^*}{x_k - x^*} = 1 - \frac{(\frac{e_{k+1}}{e_k} - 1)^2}{\frac{e_{k+2}}{e_k} - 2\frac{e_{k+1}}{e_k} + 1} = 1 - \frac{(\frac{e_{k+1}}{e_k} - 1)^2}{\frac{e_{k+2}}{e_k} - 2\frac{e_{k+1}}{e_k} + 1} \tag{28}$$

When $k \to \infty$, formula (28) is translated into formula (29). If $(\bar{x}_{k+1} - x^*)/(x_k - x^*) \to 0$ could be proved, the convergence feedrate of the Aitkin acceleration sequence $\{\bar{x}_k\}$ is faster than the convergence feedrate of the original sequence $\{x_k\}$.

$$\lim_{k \to \infty} \frac{\bar{x}_{k+1} - x^*}{x_k - x^*} = 1 - \frac{(c-1)^2}{c^2 - 2c + 1} = 0 \tag{29}$$

Steffensen Iterative Accelerator (SIA) could be defined with fixed point iteration on the basis of the Aitken iteration accelerator in formula (30).

$$x_{k+1} = x_k - \frac{f^2(x_k)}{f(x_k + f(x_k)) - f(x_k)} \tag{30}$$

The second-order convergence of SIA method is proved as follows:

$f(x_k)$ is developed through the Taylor formula of Peano remainder:

$$f(x_k) = f'(x^*)e_k + 0.5f''(x^*)e_k^2 + o(e_k^2) \tag{31}$$

$f(x_k + f(x_k))$ is developed through the Taylor formula of Peano remainder:

$$\begin{aligned}
f(x_k + f(x_k)) &= f'(x^*)[e_k + f(x_k)] \\
&\quad + 0.5f''(x^*)[e_k + f(x_k)]^2 + o(e_k^2)
\end{aligned} \tag{32}$$

Then formula (32) minus formula (31) as follows:

$$f(x_k+f(x_k))-f(x_k)=f'^2(x^*)e_k+0.5(3+f'(x^*))f'(x^*)f''(x^*)e_k^2+o(e_k^2) \tag{33}$$

Supposed that the first-order and second-order derivative of $f(x^*)$ are written as f', f'' respectively, formula (33) could transform formula (34).

$$\frac{f(x_k+f(x_k))-f(x_k)}{f(x_k)}=\frac{f'^2+0.5(3+f')f'f''e_k^2}{f'(x^*)+0.5f''e_k}+o(e_k) \tag{34}$$

In order to prove the second-order convergence of $f(x)$, we only need to guarantee that formula (35) has boundary.

$$\lim_{k\to\infty}\frac{x_{k+1}-x^*}{(x_k-x^*)^2}=\lim_{k\to\infty}\frac{e_{k+1}}{e_k^2} \tag{35}$$

After analyzing above, e_{k+1} could be written as follows:

$$\begin{aligned} e_{k+1} &= x_{k+1}-x^* \\ &= x_{k+1}-x_k+x_k-x^* \\ &= e_k-\frac{f(x_k)}{[f(x_k+f(x_k))-f(x_k)]/f(x_k)} \\ &= e_k-\frac{f'(x^*)e_k+0.5f''(x^*)e_k^2+o(e_k^2)}{\frac{f'^2(x^*)+0.5(3+f'(x^*))f''(x^*)e_k^2}{f'(x^*)+0.5f''(x^*)e_k}+o(e_k)} \\ &= \frac{\frac{0.5(1+f'(x^*))f'(x^*)f''(x^*)e_k^2}{f'(x^*)+0.5f''(x^*)e_k}+o(e_k^2)}{\frac{f'^2(x^*)+0.5(3+f'(x^*))f''(x^*)e_k}{f'(x^*)+0.5f''(x^*)e_k}+o(e_k)} \end{aligned} \tag{36}$$

When $k\to\infty$, we substitute formula (36) into formula (35) as follows:

$$\lim_{k\to\infty}\frac{x_{k+1}-x^*}{(x_k-x^*)^2}=\lim_{k\to\infty}\frac{e_{k+1}}{e_k^2}=\frac{(1+f'(x^*))f''(x^*)}{2f'(x^*)} \tag{37}$$

There is an upper boundary in formula (37), so SIA method could ensure the second-order convergence at least.

Combined with curve interpolation process, the SIA method with minimal feedrate fluctuation is presented. As shown in formula (38):

$$\begin{aligned} &do: \\ &y_i=f(t_{i-1}),\ z_i=f(y_i+t_{i-1}) \\ &t_i=t_{i-1}-\frac{y_i^2}{z_i-y_i} \\ &v_i=\frac{\|\mathrm{r}(t_i)-r(u_k)\|}{T} \\ &until: \frac{|v_i-v^*|}{v^*}\le\varepsilon \\ &then: u_{k+1}=t_i \end{aligned} \tag{38}$$

VI. SIMULATION AND EXPERIMENT RESULTS

It is from the previous demonstration that the interpolation on the basis of SIA method is equipped with the second order convergence. Besides, there is no curve derivation calculation during the process, which could converge to the expected value faster. In this section, we make comparison among SIA method, Newton iterative method, simple Newton iterative method, the second-order Taylor formula method and the first-order Taylor formula method.

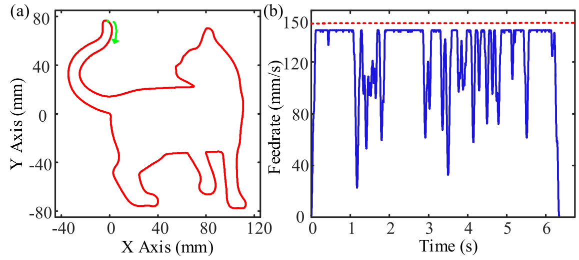

FIGURE 6. Parametric toolpath for simulations and the feedrate profile.

A cat-shaped in Fig. 6 (a) formed from the third-degree B-spline curves is utilized to test the interpolation accuracy and efficiency. Figure 6 (b) is the feedrate profile calculated from interpolation points.

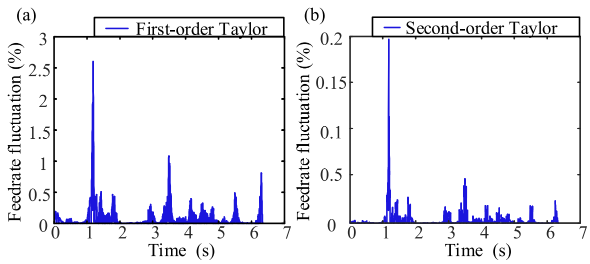

FIGURE 7. Feedrate fluctuation comparison of Taylor interpolation methods.

The feedrate fluctuation in Fig. 7 is calculated from formula (15) by means of the traditional Taylor formula. Figure 7(a) shows that the first-order Taylor interpolation could lead to apparent feedrate fluctuation and worse machining accuracy. Furthermore, when Taylor formula is used to calculate arc length, the more curved the trajectory is, the bigger truncation error is. As a result, the feedrate fluctuation becomes more and more apparent. Compared with the first-order Taylor interpolation, the second-order Taylor interpolation could reduce feedrate fluctuation evidently in the Fig. 7(b). Nevertheless, the feedrate fluctuation of the large curvature is still not satisfied, which causes that desired machining results are difficult to achieve.

In order to test performance of algorithms, the constraint value of the feedrate fluctuation is set as $\varepsilon = 0.0001\%$. The feedrate fluctuation curves are obtained from SIA method, CTA method and SCTA method in Fig. 8. We could discover that the feedrate fluctuation value of all interpolation points meets anticipated restraint.

In contrast to the SIA method, we should pay attention to the calculation efficiency of the five methods mentioned in this section. The relative calculation efficiency is defined as formula (39), where t_i^{SIA} is the time spent on the

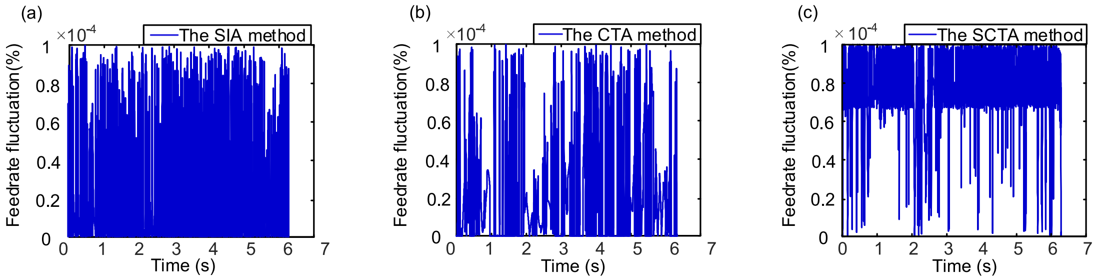

FIGURE 8. Feedrate fluctuation comparison of feedback interpolation.

ith interpolation point calculation.

$$\eta = \frac{t_i - t_i^{SIA}}{t_i^{SIA}} \times 100\% \tag{39}$$

Time consumed by CTA method is related to SIA method at each interpolation point as shown in Fig. 9 (a). We could find that the average time spent by CTA method is 1.48 times consuming of SIA method. Because convergence rate of SIA method is faster than the equivalent of CTA on each interpolation point. In Fig. 9 (b) SCTA method is worse than SIA method in terms of calculation efficiency. The average time of each point is 1.39 times longer than SIA method. When constant M is selected inappropriately, convergence rate will be reduced. In order to show iteration step from one interpolation point to the next point, Fig. 9 (c) and Fig. 9 (d) are compared SIA method with CTA method and SCTA method respectively.

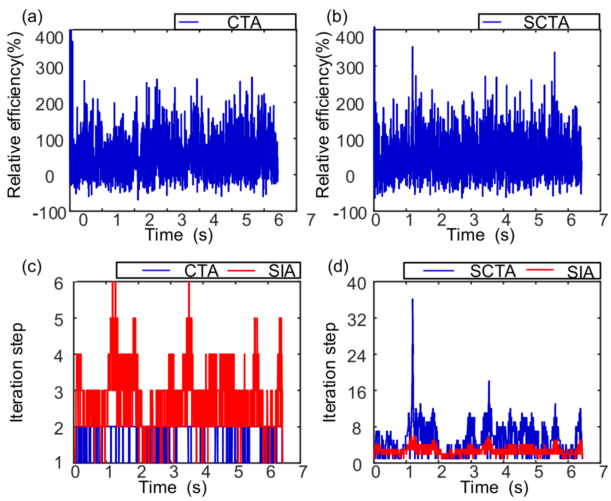

FIGURE 9. Relative efficiency and iteration step of CTA and SCTA method.

Iteration step is one of factors affecting efficiency, and the calculation time of each step is also an important factor. In Fig. 9 (c), the initial value of CTA is closer to the actual value, so the iteration steps are less than SIA steps. But every iteration period needs to calculate derivative in CTA method. In Fig. 9 (d), there is not derivative calculation in SIA method and SCTA method, therefore, step is a main factor, iteration step of SIA method is less than SCTA method's significantly.

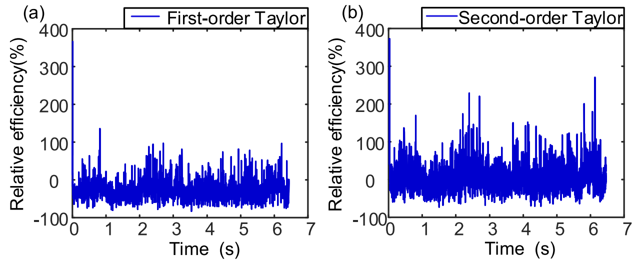

FIGURE 10. Relative calculation efficiency of Taylor interpolation methods.

Similarly, we would compare SIA method with two typical Taylor interpolation methods. On the one hand, as shown in Fig. 10 (a), the first-order Taylor interpolation method only calculates the first-order derivative. However, the apparent feedrate fluctuation is not allowed in the actual machining. On the other hand, as shown in Fig. 10 (b), the second-order Taylor interpolation method needs more complicated calculation with the second-order derivative. Therefore, comparing with SIA method, the average time of each point is shorter 0.99 times. But there is strict limit of feedrate fluctuation by means of SIA method.

VII. CONCLUSION

In this paper, the matrix expression of parametric curves and feedrate planning are introduced. Furthermore, a feedrate feedback interpolation algorithm based on Steffensen iterative acceleration is presented. The parametric toolpath interpolation is optimally designed to guarantee the minimal feedrate fluctuation. Besides, the rapid convergence and the second-order convergence are proved theoretically. Compared with previous interpolation methods in literatures, the proposed method has the following advantages:

(1) The smooth connection of parametric curves is designed to reduce direction jump of feedrate at connection points. (2) The minimal feedrate fluctuation is constrained rigorously in parametric toolpath interpolation process. (3) While improving the convergence speed, the strict convergence is ensured. (4) The algorithm avoids the derivative computation, so computational load is reduced and hard real-time task of interpolation is executed easily in motion control system. Eventually, the proposed method is validated by simulations and compared with Taylor expansion method, Newton iterative method and simple Newton iterative method

in feedrate fluctuation and computational efficiency respectively. In conclusion, this interpolator is suitable for the CNC machine tool with high feedrate and high precision.

REFERENCES

[1] G. W. Vickers and C. Bradley, "Curved surface machining through circular arc interpolation," *Comput. Ind.*, vol. 19, no. 3, pp. 329–337, Dec. 1992.
[2] D. C. H. Yang and and T. Kong, "Parametric interpolator *versus* linear interpolator for precision CNC machining," *Comput. Aided Des.*, vol. 26, no. 3, pp. 225–234, Mar. 1994.
[3] Q. G. Zhang and R. B. Greenway, "Development and implementation of a NURBS curve motion interpolator," *Robot. Comput.-Integr. Manuf.*, vol. 14, no. 1, pp. 27–36, Feb. 1998.
[4] A. Shahzadeh, A. Khosravi, S. Nahavandi, and T. Robinette, "Smooth path planning using biclothoid fillets for high speed CNC machines," *Int. J. Mach. Tools Manuf.*, vol. 132, pp. 36–49, Sep. 2018.
[5] H.-T. Yau and J.-B. Wang, "Fast Bezier interpolator with real-time lookahead function for high-accuracy machining," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 10, pp. 1518–1529, Aug. 2007.
[6] C. Lartigue, F. Thiebaut, and T. Maekawa, "CNC tool path in terms of B-spline curves," *Comput. Aided Des.*, vol. 33, no. 4, pp. 307–319, Apr. 2001.
[7] L. Piegl and W. Tiller, "Curve and surface basics," in *The NURBS Book*, 2nd ed. New York, NY, USA: Springer, 1977, pp. 1–46.
[8] S. Bedi and I. Q. N. Ali, "Advanced interpolation techniques for N.C. machines," *ASME J. Eng. Ind.*, vol. 115, no. 8, pp. 329–336, 1993.
[9] J.-T. Huang and D. C. H. Yang, "Precision command generation for computer controlled machines," in *Proc. ASME Prod. Eng. Division*, New York, NY, USA, 1992, pp. 89–104.
[10] Y. Koren, M. Shpitalni, and C. C. Lo, "CNC interpolators: Algorithms and analysis," in *Proc. ASME Prod. Eng. Division*, New York, NY, USA, 1993, pp. 83–92.
[11] R. T. Farouki and Y. F. Tsai, "Exact Taylor series coefficients for variable-feedrate CNC curve interpolators," *Comput.-Aided Des.*, vol. 33, no. 2, pp. 155–165, Jul. 2001.
[12] S.-S. Yeh and P.-L. Hsu, "Adaptive-feedrate interpolation for parametric curves with a confined chord error," *Comput.-Aided Des.*, vol. 34, no. 3, pp. 229–237, Mar. 2002.
[13] M.-Y. Cheng, M.-C. Tsai, and J.-C. Kuo, "Real-time NURBS command generators for CNC servo controllers," *Int. J. Mach. Tools Manuf.*, vol. 42, pp. 801–813, May 2002.
[14] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation," *Int. J. Mach. Tools Manuf.*, vol. 41, pp. 1323–1345, Jul. 2001.
[15] H. Liu, S. Zhou, C. Li, S. Yuan, and Q. Liu, "Principle and development of NURBS interpolation algorithm with zero-feedrate fluctuation," *Comput. Integr. Manuf. Syst.*, vol. 21, pp. 2650–2667, Oct. 2015.
[16] M. Heng and K. Erkorkmaz, "Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability," *Int. J. Mach. Tools Manuf.*, vol. 50, no. 3, pp. 281–293, Mar. 2010.
[17] S. Y. Jeong, Y. J. Choi, and P. Park, "Parametric interpolation using sampled data," *Comput.-Aided Des.*, vol. 38, no. 1, pp. 39–47, Jan. 2006.
[18] W. T. Lei, M. P. Sung, L. Y. Lin, and J. J. Huang, "Fast real-time NURBS path interpolation for CNC machine tools," *Int. J. Mach. Tools Manuf.*, vol. 47, pp. 1530–1541, Aug. 2007.
[19] H. Liu, Q. Liu, C. Li, S. Yuan, and S. Zhou, "A NURBS interpolation method with minimal feedrate fluctuation for CNC machine tools," *Int. J. Adv. Manuf. Technol.*, vol. 78, pp. 1241–1250, May 2015.
[20] Z. Zhongyi, Y. Chao, Y. Guodong, and J. Fengshui, "High-precision NURBS interpolation algorithm based on the bisection method," in *Proc. IEEE Control Conf.*, Hangzhou, China, Jul. 2015, pp. 8673–8677.
[21] X.-T. Zhang and Z. Song, "An iterative feedrate optimization method for real-time NURBS interpolator," *Int. J. Adv. Manuf. Technol.*, vol. 62, pp. 1273–1280, Oct. 2012.
[22] Z.-Y. Jia, D.-N. Song, G.-Q. Hu, W.-W. Su, and J.-W. Ma, "A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints," *Int. J. Mach. Tools Manuf.*, vol. 116, pp. 1–17, May 2017.
[23] C.-C. Lo, "Feedback interpolators for CNC machine tools," *J. Manuf. Sci. Eng.*, vol. 119, no. 4, pp. 587–592, 1997.
[24] C. W. Cheng, J. Maciejowski, and M. C. Tsai, "Feedrate control for non-uniform rational B-spline motion command generation," *Proc. Inst. Mech. Eng. B, J. Eng. Manuf.*, vol. 220, no. 11, pp. 1855–1861, 2006.
[25] M. C. Tsai and C. W. Cheng, "A real-time predictor-corrector interpolator for CNC machining," *J. Manuf. Sci. Eng.*, vol. 125, no. 3, pp. 449–460, 2003.
[26] A.-C. Lee, M.-T. Lin, Y.-R. Pan, and W.-Y. Lin, "The feedrate scheduling of NURBS interpolator for CNC machine tools," *Comput.-Aided Design*, vol. 43, pp. 612–628, Jun. 2011.
[27] H. Zhao, H. Ding, and L. Zhu, "A parametric interpolator with minimal feed fluctuation for CNC machine tools using arc-length compensation and feedback correction," *Int. J. Mach. Tools Manuf.*, vol. 75, pp. 1–8, Dec. 2013.
[28] M. Liu, Y. Huang, L. Yin, J. Guo, X. Shao, and G. Zhang, "Development and implementation of a NURBS interpolator with smooth feedrate scheduling for CNC machine tools," *Int. J. Mach. Tools Manuf.*, vol. 87, pp. 1–15, Dec. 2014.
[29] W. Zhong, X. Luo, W. Chang, F. Ding, and Y. Cai, "A real-time interpolator for parametric curves," *Int. J. Mach. Tools Manuf.*, vol. 125, pp. 133–145, Feb. 2018.
[30] J. Wu, H. Zhou, J. Chen, and X. Tang, "A NURBS interpolation algorithm with continuous feedrate," *Int. J. Adv. Manuf. Technol.*, vol. 59, pp. 623–632, Mar. 2012.
[31] H. Liu, Q. Liu, and S. Yuan, "Adaptive feedrate planning on parametric tool path with geometric and kinematic constraints for CNC machining," *Int. J. Adv. Manuf. Technol.*, vol. 90, pp. 1889–1896, May 2017.
[32] L. Lu, L. Zhang, Y. Han, J. Zhao, and S. Ji, "An offline predictive feedrate scheduling method for parametric interpolation considering the constraints in trajectory and drive systems," *Int. J. Adv. Manuf. Technol.*, vol. 83, pp. 2143–2157, Apr. 2016.
[33] L. Zhang, Y. Bian, K. Wang, and H. Chen, "Implementation of a CNC NURBS curve interpolator based on control of speed and precision," *Int. J. Adv. Manuf. Technol.*, vol. 47, no. 6, pp. 1505–1519, Jan. 2009.
[34] J. Dong, T. Wang, Y. Ding, and B. Li, "Smooth feedrate planning for continuous short line tool path with contour error constraint," *Int. J. Mach. Tools Manuf.*, vol. 76, pp. 1–12, Jan. 2014.
[35] H. Zhao, L. M. Zhu, and H. Ding, "A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments," *Int. J. Mach. Tools Manuf.*, vol. 65, pp. 89–98, Feb. 2013.
[36] M.-C. Tsai, C.-W. Cheng, and M.-Y. Cheng, "A real-time NURBS surface interpolator for precision three-axis CNC machining," *Int. J. Mach. Tools Manuf.*, vol. 43, no. 12, pp. 1217–1227, Sep. 2003.

JIALI JIANG was born in Qiqihar, Heilongjiang, China, in 1992. She received the B.E. degree in mechanical design, manufacture and automation from the Harbin University of Science and Technology, China, in 2015. She is currently pursuing the Ph.D. degree with the Department of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing.

She has been studying with the Intelligent NC System Technology Laboratory, Tsinghua University, since 2017. She has participated in several major national science and technology projects. She has assisted teammates to publish several papers, e.g., optimal curvature-smooth transition and efficient feedrate optimization method with axis kinematic limitations for linear toolpath and efficient feedrate optimization method for spline toolpath with curvature-base planning and accurate interpolating, accepted by international journals. Her research interests include CNC machining, motion control, tracking characteristic, and parametric interpolation.

FUYAN LIN, photograph and biography not available at the time of publication.

YONG ZHANG, photograph and biography not available at the time of publication.

HUI ZHANG, photograph and biography not available at the time of publication.

PEIQING YE, photograph and biography not available at the time of publication.