IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 February 2023, accepted 20 February 2023, date of publication 23 February 2023, date of current version 17 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3248081

RESEARCH ARTICLE

Jerk-Continuous Feedrate Optimization Method for NURBS Interpolation

MINGXING NIE, LIWEI ZOU, AND TAO ZHU, (Member, IEEE)
School of Computer Science, University of South China, Hengyang 421001, China

Corresponding author: Tao Zhu (tzhu@usc.edu.cn)

This work was supported in part by the Natural Science Foundation of Hunan Province under Grant 2021JJ30574, in part by the Research Foundation of Education Bureau of Hunan Province under Grant 21B0424, and in part by the Hengyang Science and Technology Major Project under Grant 202250015428.

ABSTRACT Feedrate scheduling is one of the most critical technologies in CNC machining, requiring a reasonable balance between efficiency and quality. This paper proposes a jerk-continuous feedrate smoothing (JCFS) method to generate a low-vibration and smooth feedrate profile for non-uniform rational B-spline (NURBS) interpolation. Firstly, the segmentation concept is introduced to subdivide the entire trajectory into segments to accommodate curvature changes of the NURBS curve, accelerating the acceleration/deceleration process. Secondly, a length threshold-based curve segment classification method is proposed to overcome the complexity of the traditional acceleration and deceleration algorithms. The curve segments are divided into long, medium, and short types, and the length threshold calculation model is derived. Next, to avoid computational complexity for engineering applications, a model is established for the first time to calculate the actual maximum feedrate for different types of segments. Finally, the horizontal-8-shaped and butterfly-shaped NURBS curves are simulated and analyzed. The simulation results indicate that the machining quality is steadily improved while several key indicators remain within the given tolerances. Compared with the traditional method, the proposed method reduces the computational and interpolation time by 17.2% and 22.8%, respectively, demonstrating the feasibility and effectiveness of the method.

INDEX TERMS Feedrate scheduling, NURBS interpolation, jerk-continuous, segmentation.

I. INTRODUCTION

Due to the advantages of NURBS interpolation in terms of machining accuracy and code size, many scholars have been interested in applying NURBS direct interpolation in CNC to meet the requirements of surface quality and machining time [1], [2], [3], [4], [5]. NURBS direct interpolation has gradually become the main technology in multi-axis CNC machining [6], [7], [8], [9], [10].

After decades of development, several NURBS direct interpolation methods have been proposed. Shpitalni et al. first proposed the concept of NURBS direct interpolation [6]. Then, Yang and Kong presented Taylor's expansion-based NURBS interpolation [11]. Erkorkmaz et al. developed a quintic spline interpolation method [12]. Lei et al. proposed the inverse length functions (ILF) method [13]. However, these methods inevitably suffer from high chord errors, acceleration, and jerk exceeding given tolerances due to curvature.

The feedrate should be changed to keep the chord error within the tolerance. The NURBS interpolator feedrate scheduling method is applied to improve machining accuracy and motion smoothness. Yeh and Hsu put forward an adaptive feedrate interpolator to satisfy the given tolerance of chord error [14]. However, this method suffers from the feedrate discontinuity problem. Sun et al. proposed a new algorithm based on proportional adjustment for the sensitive regions to obtain the adaptive feedrate profile [15], which is too computationally intensive for real-time imputation. Wang et al. developed an offline-online two-stage NURBS interpolator, which applies a constant feedrate scheduling method to the NURBS segment [16]. Jia et al. also proposed a systematic NURBS interpolator that maintains a constant feedrate in the federate-sensitive region, which reduces the

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.

TABLE 1. The comparison of jerk-continuous ACC/DEC methods.

Reference	ACC/DEC method	Jerk Profile Continuity	Machining efficiency	Computational load
[24]	5th order polynomial functions	second-order continuous	Low	High
[25]	piecewise trigonometric function	Continuous	High	High
[26]	piecewise quartic spline curve	first-order continuous	Low	High
[27]	trigonometric functions	Continuous	Low	Low
[28]	polynomial functions	discontinuous	Low	Low
[29]	trigonometric methods	Continuous	High	High
[30]	polynomial functions	Continuous	High	High

processing complexity at the expense of time efficiency [17]. Ni et al. presented a bidirectional feedrate s. Guo et al. developed a feedrate planning method, including three combined curves: a critical deceleration curve, a straight line, and a critical acceleration curve to reduce computing load and improve efficiency [19]. However, sudden changes in curvature will lead to discontinuous feedrates, causing acceleration beyond the machine's capacity and vibration.

Several new interpolations were constructed in literature to address these issues [20], [21], [22], [23]. Javad et al. first proposed the acc-jerk-limited feedrate scheduling by the improved quintic feedrate profile [20]. Sun et al. [21] developed a relaxation mathematical process feedrate scheduling method with a little jerk for five-axis CNC machining. Erwinski et al. [22] presented a PSO-based feedrate optimization method to select appropriate axial velocity, acceleration, and jerk for NURBS interpolation. Although Krystian Adam Erwinski et al. [23] presented a new NURBS feedrate optimization method with axial acceleration and jerk constraints, the presented method still suffers from the jerk discontinuity problem.

Various jerk-continuous methods have been presented in the literature to solve the jerk-discontinuity problem, as shown in Table 1 [24], [25], [26], [27], [28], [29], [30]. P. Boscariol et al. presented novel trajectory-planning algorithms for an industrial robot using fifth-order polynomial functions [24]. Fang et al. presented a modified sine jerk motion profile based on piecewise trigonometric functions. However, these methods are only suitable for low-speed robot joint motion [25]. Zhao et al. proposed a new ACC/DEC approach with jerk continuity using the piecewise quartic spline curve [26]. Liu et al. adopted the trigonometric function method to establish the jerk continuous profile for NURBS interpolation. However, these methods cannot maintain the maximum jerk and acceleration, significantly reducing the machining efficiency [27]. Zhang et al. designed a linear jerk-continuous profile for feedrate scheduling [28], which cannot maintain the maximum acceleration. Ni et al. proposed a jerk profile construction method by combining trigonometric and polynomial functions [29]. However, this method has 16 stages and too many types of processing, leading to a high time overhead. Zhang et al. developed a new jerk-smooth feedrate scheduling method [30]. However, this method is time-consuming due to the use of high-order polynomials. Aiming at the smoothing of the velocity and acceleration at the end of the trajectory of adjacent micro-straight lines, Sun and Yang et al. focused on jerk-continuous tool path smoothing and feedrate planning for 6-axis robot machining with more complex kinematics [31], [32]. However, these methods mainly construct a B-spline curve in the remaining part of the adjacent micro-straight lines trajectory, somewhat different from the proposed NURBS direct interpolation.

Inspired by these works, this paper aims to design an efficient trigonometric-based jerk-continuous profile for NURBS interpolation to reduce feedrate fluctuations and improve interpolation efficiency. Compared to traditional works, the proposed approach has the following salient features. Firstly, NURBS curves are subdivided into segments by confined curvature to accommodate curvature variations. Secondly, a length threshold-based segment classification method is proposed to overcome the complexity of conventional acceleration and deceleration algorithms. Finally, a model is derived for the first time to calculate the actual maximum feedrate for different segments.

The main contributions of this paper can be summarized as follows:

(1) We focus on the critical role of curvature in the NURBS interpolation, present the concept of confined curvature, and implement segmentation of NURBS to adapt to the curvature variations.

(2) A trigonometric-function-based jerk-continuous feedrate smoothing (JCFS) method is proposed to overcome the feedrate fluctuation.

(3) The proposed JCFS is employed to derive the analytic expression for calculating the maximum feedrate of different length curve types for the first time, which can significantly simplify the ACC/DEC implementation process.

The rest of this paper is organized as follows. Section II describes the architecture of the proposed jerk-continuous feedrate optimization interpolator. Section III introduces the preliminaries, including NURBS interpolation, confined curvature, and curve segmentation. The jerk-continuous NURBS interpolation is designed in Section IV. Section V provides the simulation results of the proposed method applied to horizontal-8-shaped and butterfly-shaped NURBS curves. Finally, conclusions and future aspects are presented in Section VI.

II. JERK-CONTINUOUS FEEDRATE OPTIMIZATION INTERPOLATOR ARCHITECTURE

The interpolation feedrate will be continuously influenced by geometric and dynamic characteristics [35]. The feedrate should be dynamically adjusted to satisfy the tolerance of

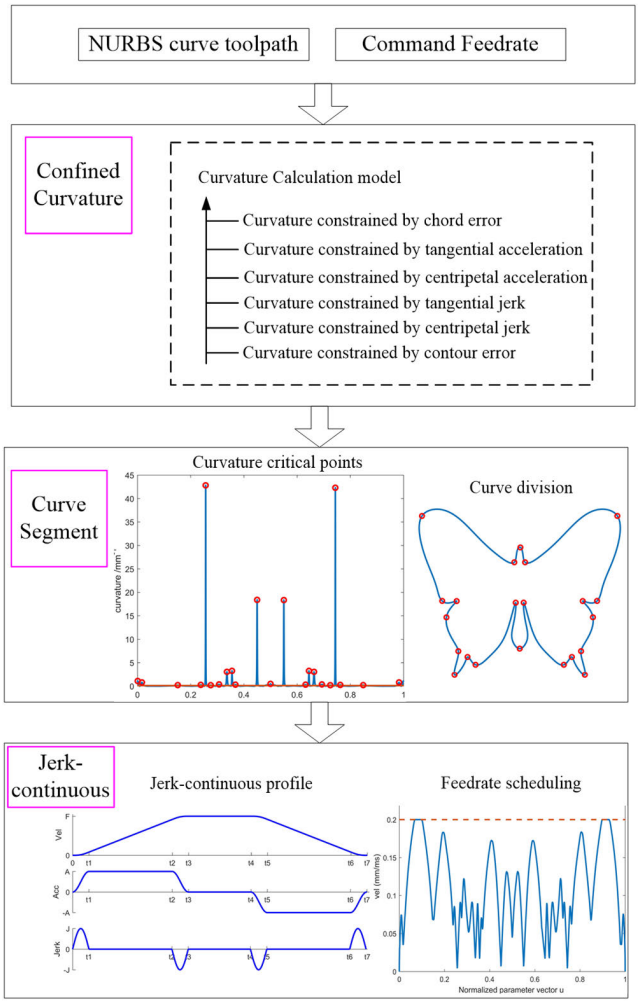

FIGURE 1. The architecture of the jerk-continuous NURBS interpolator.

specific constraints. However, since the geometric properties vary continuously along with the NURBS, using the same feedrate limits for the entire curve is inefficient. Different processing strategies can be employed depending on the region of curve curvature. Here, the method proposed in reference [3] for partitioning NURBS curves according to the confined curvature is utilized [3].

The curve segmentation is based on dividing a NURBS curve into segments by critical curvature points. A critical curvature point is obtained from the local maximum curvature point in the curvature-sensitive region obtained by the confined curvature.

As shown in Fig. 1, the architecture of the proposed method proposed is specified as follows.

Firstly, according to the NURBS curve defined in Eq. 1, the interpolator reads the NURBS curve tool path and kinematic parameters in turn, including control points and node vectors, as well as other information such as commanded feed rate, acceleration, and Jerk. Based on these inputs, the constrained curvature is calculated, and the NURBS curve can be divided into several parts with constrained curvature containing curvature-sensitive areas.

Then, the critical point is found at the curvature-sensitive region, which is the local curvature maximum point. The NURBS curve is segmented into several curve parts by the critical points so that the subsequent acceleration and deceleration can be processed independently in the curve segment unit.

Finally, the jerk-continuous method based on trigonometric functions is proposed. The curves are divided into long, medium, and short categories. All curve segments are recalculated to the maximum achievable velocity. Besides, the jerk-continuous feedrate optimization is performed according to the proposed acceleration/deceleration method to realize continuous interpolation of velocity/acceleration/jerk.

III. PRELIMINARIES

A. PRINCIPLE OF THE NURBS INTERPOLATION

The definition of the NURBS curve can be expressed as follows [33]:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\omega_i p_i}{\sum_{i=0}^{n} N_{i,p}(u)\omega_i}, u_1 \le u \le u_{n+p+1} \tag{1}$$

where i=0, 1, …, n; $\{P_i\}$ is the set of control points; $\{\omega_i\}$ is the set of weights; $\{N_{i,p}(u)\}$ represents the B-spline basic functions; u is the normalized parameter vector.

The commonly used Taylor expansion method is adopted to estimate the parameter for point $C(u)$ [34].

$$u_{i+1} = u_i + \left.\frac{du}{dt}\right|_{t=t_i} + \frac{1}{2}\left.\frac{d^2u}{dt^2}\right|_{t=t_i} T^2 + \varepsilon \tag{2}$$

where T is the cycle period. ε is a high-order term that can be ignored.

The feedrate can be obtained by differentiating from Eq. (1) versus t

$$v(u_i) = \left\|\frac{dC(u)}{dt}\right\|_{u=u_i} = \left\|\frac{dC(u)}{du}\right\|_{u=u_i} \left.\frac{du}{dt}\right|_{t=t_i} \tag{3}$$

Then, the first-order derivative of u with respect to time t is given by

$$\left.\frac{du}{dt}\right|_{t=t_i} = \frac{v(u_i)}{\left\|\frac{dC(u)}{du}\right\|_{u=u_i}} \tag{4}$$

Similarly, the second-order derivative of u is given by

$$\left.\frac{d^2u}{d^2t}\right|_{t=t_i} - \frac{v^2(u_i)\left(\frac{dC(u)}{du}\cdot\frac{dC^2(u)}{du^2}\right)\Big|_{u=u_i}}{\left\|\frac{dC(u)}{du}\right\|^4_{u=u_i}} \tag{5}$$

The curvature of the NURBS curve can be expressed as

$$k = \frac{\left\|C^{(1)}(u) \times C^{(2)}(u)\right\|}{\left\|C^{(1)}(u)^3\right\|} \tag{6}$$

The radius of curvature can be obtained as

$$\rho = \frac{1}{k} \tag{7}$$

Thus, substituting equations 4 and 5 into Eq.2 gives

$$u_{i+1} = u_i + \frac{v(u_i)T}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} - \frac{v^2(u_i)\left(\frac{dC(u)}{du} \cdot \frac{dC^2(u)}{du^2}\right)\Big|_{u=u_i}}{\left\| \frac{dC(u)}{du} \right\|^4_{u=u_i}} \quad (8)$$

According to Eq.8, the parameter u_{i+1} of $C(u_{i+1})$ can be estimated based on the current interpolation parameter u_i, the current velocity $v(u_i)$, and the interpolation period T.

B. CONFINED CURVATURE AND CRITICAL POINT

Geometric and kinematic characteristics, drive, and contour error constraints constantly influence Feedrate. The multiple constraints are introduced here [35] to obtain the constrained curvature radius model as

$$\begin{cases} \rho_{i,1} = \frac{v^2T^2+4\delta^2}{8\delta} \\ \rho_{i,2} = \frac{v^2}{a_n} \\ \rho_{i,3} = \sqrt{\frac{v^3}{j_n}} \\ \rho_{i,4} = \frac{v^2T^2}{2\varepsilon} \end{cases} \quad (9)$$

where v, T are the velocity and cycle time, δ, ε are the tolerance of chord and contour errors, respectively, a_n, j_n are the maximum normal acceleration and normal jerk.

Thus, from Eq. 7 and Eq. 9, the confined curvature k_c can be obtained as follows.

$$k_c = \min(1/\rho_{i,1}, 1/\rho_{i,2}, 1/\rho_{i,3}, 1/\rho_{i,4}) \quad (10)$$

The above analysis indicates that when the curve curvature is less than the confined curvature k_c, the feedrate of the NURBS interpolator can satisfy various constraints and remain at the command velocity. On the contrary, when the curve curvature is greater than the confined curvature k_c, the feedrate should be reduced to satisfy the machining accuracy, and the feedrate should be simultaneously adjusted to meet the kinematic characteristic. Therefore, the region of the curve below the confined curvature is defined as the curvature-sensitive region. Due to the curvature continuity characteristics of the cubic NURBS curve, the local curvature maximum point can be found from the curvature-sensitive region and employed as the critical point. The NURBS curve can be divided into several curve segments by the critical points so that the subsequent acceleration and deceleration can be processed independently in terms of curve segments.

C. CURVE SEGMENT LENGTH

The adaptive Simpson's method is utilized to calculate the curve segment length. For more details, please refer to reference [17]. Finally, the arc displacements of each curve segment between two adjacent critical points can be calculated. The constraint velocity for the curve segment can be obtained by inverse derivation of Eq.9 to obtain the corresponding features vector, where u_s and u_e are the start point and end point parameters of the curve segment, v_s and v_e are the constrained velocity, l_i is the arc length of the curve segment.

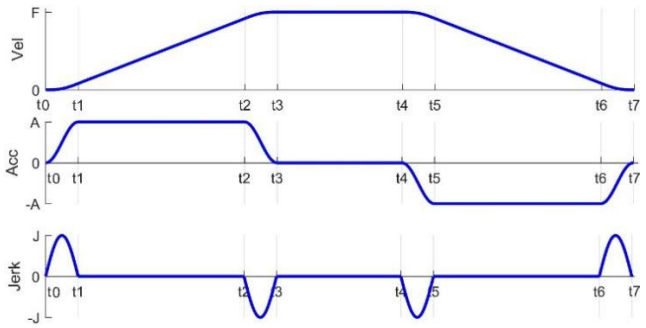

FIGURE 2. The jerk-continuous feedrate profile.

IV. DESIGN OF JERK-CONTINUOUS NURBS INTERPOLATION

The curve segmentation aims to impose acceleration and jerk constraints on the feedrate scheduling in a tangential direction based on the proposed acceleration and deceleration algorithms. However, since it is difficult to determine the actual velocity, the feedrate scheduling is very complicated for different lengths of curve segments, especially medium-long curve segments. Generally speaking, there may be dozens of accelerations and deceleration processing types for different-length curve segments.

This section illustrates the details of the proposed jerk-continuous ACC/DEC feedrate scheduling. Firstly, the jerk-continuous scheduling method is introduced based on the trigonometric function, which can effectively improve jerk continuity at the start and stop times to prevent the vibration caused by the sudden increase or drop of the jerk. Secondly, the calculation model of the actual allowable feedrate of the curve segment is deduced, which is the most effective solution to solve the feedrate scheduling of the curve segment with different lengths. Finally, two constant acceleration stages are added based on the 5-stage jerk-continuous method [36] to improve velocity acceleration and deceleration capabilities.

A. JERK-CONTINUOUS FEEDRATE PROFILE

As shown in Fig. 2, the trigonometric-function-based jerk-continuous feedrate optimization method is proposed based on [37]. The method includes the acceleration rising stage, constant stage, and falling stage. The acceleration rising stage is subdivided into three phases: acceleration increase/acceleration constant/acceleration decrease. The acceleration falling stage also includes three subdivision phases. The jerk expression at any time t can be described as follows:

$$j(t) = \begin{cases} \frac{\pi}{2T_1}A \sin\frac{\pi}{T_1}t & t_0 \le t < t_1 \\ 0 & t_1 \le t < t_2 \\ -\frac{\pi}{2T_3}A \sin\frac{\pi(t-t_2)}{T_3} & t_2 \le t < t_3 \\ 0 & t_3 \le t < t_4 \\ -\frac{\pi}{2T_5}A \sin\frac{\pi(t-t_4)}{T_5} & t_4 \le t < t_5 \\ 0 & t_5 \le t < t_6 \\ \frac{\pi}{2T_7}A \sin\frac{\pi(t-t_6)}{T_7} & t_6 \le t \le t_7 \end{cases} \quad (11)$$

where A is the maximum acceleration. t_0-t_1 is the phase the jerk changes according to the trigonometric function, which increases from 0 to the maximum jerk, and then decreases from the maximum jerk to 0; t_1-t_2 is the phase where the jerk remains at 0; t_2-t_3 is the inverse process of t_0-t_1. t_3-t_4 is also the phase where the jerk remains at 0; t_4-t_7 is the inverse process of t_0-t_3. From Eq. 15, the jerk is represented by a sinusoidal function, and the jerk variation is relatively continuous.

Integrating Eq.11 yields the following acceleration equation:

$$a(t) = \begin{cases} A(1-\cos\frac{\pi t}{T_1})/2 & t_0 \le t < t_1 \\ A & t_1 \le t < t_2 \\ A(1+\cos\frac{\pi(t-t_2)}{T_3})/2 & t_2 \le t < t_3 \\ 0 & t_3 \le t < t_4 \\ -A(1-\cos\frac{\pi(t-t_4)}{T_5})/2 & t_4 \le t < t_5 \\ -A & t_5 \le t < t_6 \\ -A(1+\cos\frac{\pi(t-t_6)}{T_7})/2 & t_6 \le t \le t_7 \end{cases} \tag{12}$$

where A is the maximum acceleration, t_0-t_1 is the increasing acceleration phase based on the trigonometric function to increase from 0 to the maximum acceleration continuously; t_1-t_2 is the constant acceleration phase, maintaining the maximum acceleration so that the velocity can be increased rapidly with the maximum acceleration; t_2-t_3 is the decreasing acceleration phase, which employs the trigonometric function to decrease from the maximum acceleration to 0 continuously. t_3-t_4 is the phase that keeps the acceleration at 0; t_4-t_7 is the inverse process of t_0-t_3.

By integrating Eq.12, the velocity equation can be obtained as (13), shown at the bottom of the next page, where v_s, V_f and v_e are the start, command, and end velocities. T_1 is the duration time from t_0 to t_1, and T_2-T_7 are similarly the corresponding durations of t_2-t_7. The velocity corresponding to the time points of T_1, T_2, T_3, T_4, T_5, T_6, and T_7 are v_1, v_2, v_3, v_4, v_5, v_6, and v_7. One equation can be obtained from Eq.13.

$$\begin{aligned} v_0 &= v_s \\ v_1 &= v_s + \tfrac{A}{2}T_1 \\ v_2 &= v_s + \tfrac{A}{2}T_1 + AT_2 \\ v_3 &= V_f = v_s + \tfrac{A}{2}T_1 + AT_2 + \tfrac{A}{2}T_3 \\ v_4 &= V_f \\ v_5 &= V_f - \tfrac{AT_5}{2} \\ v_6 &= V_f - \tfrac{AT_5}{2} - AT_6 \\ v_7 &= v_e = V_f - \tfrac{AT_5}{2} - AT_6 - \tfrac{AT_7}{2} \end{aligned} \tag{14}$$

When the maximum Jerk is J, the following equation can be derived from Eq.11

$$J = \frac{\pi}{2T_1}A = \frac{\pi}{2T_3}A = \frac{\pi}{2T_5}A = \frac{\pi}{2T_7}A \tag{15}$$

According to Eq.15, the following formula can be obtained:

$$T_1 = T_3 = T_5 = T_7 = \frac{\pi}{2}\frac{A}{J} \tag{16}$$

Eq.14 shows that

$$\begin{cases} V_f = v_s + \frac{A}{2}T_1 + AT_2 + \frac{A}{2}T_3 \\ v_e = V_f - \frac{AT_5}{2} - AT_6 - \frac{AT_7}{2} \end{cases} \tag{17}$$

By combining formulas 16 and 17, the following formula can be obtained:

$$\begin{cases} A = \frac{V_f - v_s}{T_1 + T_2} \\ A = \frac{V_f - v_s}{T_5 + T_6} \end{cases} \tag{18}$$

Then

$$\begin{cases} T_2 = \frac{V_f - V_s}{A} - T_1 \\ T_6 = \frac{V_f - V_e}{A} - T_5 \end{cases} \tag{19}$$

For every NURBS curve segment, the time of the seven phases T_1-T_7 in the proposed method can be obtained according to Eqs.17-19. The feedrate at any time can be calculated according to Eq.14.

B. CURVE SEGMENT CLASSIFICATION

The NURBS curve is segmented into several curve segments. However, each curve segment has a different arc length. According to the proposed jerk-continuation method, when the arc length is not enough to accelerate the feedrate to the commanded one, some of the seven stages will not exist. There may be many complex and difficult special cases (i.e., seventeen feedrate profiles in [36]). So far, there is still no simple and effective method to deal with the feedrate scheduling problem for different length curve segments.

The main problem is that the actual maximum velocity cannot be directly obtained for different length curve segments. In this respect, the NURBS curve segments are classified into long, medium, and short segments. For the long segments, since the feedrate can reach the command velocity, the feedrate scheduling process is performed according to the proposed 7-phase jerk-continuous profile; For the medium segments, a mathematical model of the achievable velocity is established to solve the problem that the curve segment is not long enough or not very short and should generate various special feedrate profiles to deal with; For the short segments, considering the complexity and efficiency of processing, the interpolation of the curve segment is completed at a low feedrate in a constant velocity mode. Due to the short arc length, it cannot significantly influence the processing efficiency for the entire NURBS curve, while the provided simplicity and precision are significant.

1) TYPES OF CURVE SEGMENTS

Here, the definitions for three different curve segments are presented based on curve segment length. Suppose L_{med} and

$L_{\min}$ refer to the criterion for the medium and short segments, respectively.

$$\begin{cases} \textit{long} \text{ segment:} & \text{if} L > L_{med} \\ \textit{medium} \text{ segment:} & \text{if } L \geq L_{\min} \text{ and } L \leq L_{med} \\ \textit{short} \text{ segment:} & \text{if} L < L_{\min} \end{cases} \tag{20}$$

where L is the curve segment length. The segment is long if its curve is longer than L_{med}; The segment is short if its curve is shorter than $L_{\min}$; the segment is medium if its curve is between L_{med} and $L_{\min}$.

In [29], the acceleration rising and falling phases are divided into three sub-phases, increasing the velocity processing complexity. In this paper, the three sub-stages are integrated into one stage so that the T_{i1}, T_{i2}, and T_{i3} parts can be combined into one part (T_i) to reduce the processing complexity.

Here, we assume F is the achievable velocity. According to [29], the acceleration and deceleration displacements, denoted by s_{acc} and s_{dec}, can be obtained as:

$$\begin{cases} s_{acc} = \frac{(F+v_s)(T_1+T_2+T_3)}{2} \\ s_{dec} = \frac{(F+v_e)(T_5+T_6+T_7)}{2} \end{cases} \tag{21}$$

where F is the actual velocity, v_s and v_e are the beginning and end velocities, and T_1-T_7 are the time durations of each phase. When the segment is long enough, the actual velocity can equal the command velocity V_f, and there will be a constant velocity phase. As the length of the curve segment reduces, there may be no constant velocity phase, and the velocity cannot reach the command velocity. Let L be the shortest distance that the actual velocity can reach the command velocity, and there is just no constant velocity phase. It can be obtained that:

$$L = s_{acc} + s_{dec} = \frac{(F+v_s)(T_1+T_2+T_3)+(F+v_e)(T_5+T_6+T_7)}{2} \tag{22}$$

Then, the criterion length for L_{med} defined in Eq.20 can be expressed as:

$$L_{med} = \frac{(F+v_s)(T_1+T_2+T_3)+(F+v_e)(T_5+T_6+T_7)}{2} \tag{23}$$

The expression of $L_{\min}$ will be given in Section 4.2.3.

2) THE ACTUAL ACHIEVABLE FEEDRATE

This paper considers the following definitions. The maximum achievable feedrate for the long segment is equal to the command feedrate. The maximum achievable feedrate for the short segment is the minimum feedrate at both ends of the segment. The maximum achievable feedrate for the medium segment is generally less than the command velocity, directly related to the curve segment length.

Next, the derivation process of the maximum achievable feedrate for the medium segment will be presented. Suppose L is the length of the medium segment, where $L \geq L_{\min}$ and $L \leq L_{med}$.

Substituting Eq. (16) and Eq. (19) into Eq. (23) gives (24), as shown at the bottom of the next page.

Then, we can get the following quadratic equation:

$$2F^2 + F\frac{\pi A^2}{J} + v_s\frac{\pi A^2}{2J} + v_e\frac{\pi A^2}{2J} - v_s^2 - v_e^2 - 2AL = 0 \tag{25}$$

Let $a = 2$, $b = \frac{\pi A^2}{J}$, $c = v_s\frac{\pi A^2}{2J} + v_s\frac{\pi A^2}{2J} + v_e\frac{\pi A^2}{2J} - v_s^2 - v_e^2 - 2AL$

In this way, the quadratic equation for F can be represented as:

$$aF^2 + bF + c = 0$$

In order to solve (25), the following notation is considered:

$$\begin{aligned} \Delta &= b^2 - 4ac \\ &= 8(v_s - \frac{1}{4}\frac{\pi A^2}{J})^2 + 8(v_e - \frac{1}{4}\frac{\pi A^2}{J})^2 + 16AL \end{aligned} \tag{26}$$

The detailed derivation process is presented in Appendix A. since $A > 0$ and $L > 0$, $\Delta = b^2 - 4ac > 0$

Then, there are two real solutions:

$$F = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{27}$$

Since $F > 0, -b < 0$ and 2a>0, the solution of the equation is

$$F = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{28}$$

Thus, the maximum achievable velocity for the three segments can be obtained as follows (29), as shown at the bottom of the next page.

$$v(t) = \begin{cases} v_s + \frac{A}{2}\left(t - \frac{T_1}{\pi}\sin(\frac{\pi}{T_1}t)\right) & t_0 \leq t < t_1 \\ v_s + \frac{A}{2}T_1 + A(t - t_1) & t_1 \leq t < t_2 \\ v_s + \frac{A}{2}T_1 + AT_2 + \frac{A}{2}\left((t - t_2) + \frac{T_3}{\pi}\sin(\frac{\pi}{T_3}(t - t_2))\right) & t_2 \leq t < t_3 \\ V_f & t_3 \leq t < t_4 \\ V_f - \frac{A}{2}\left((t - t_4) - \frac{T_5}{\pi}\sin(\frac{\pi}{T_5}(t - t_4))\right) & t_4 \leq t < t_5 \\ V_f - \frac{AT_5}{2} - A(t - t_5) & t_5 \leq t < t_6 \\ V_f - \frac{A}{2}T_5 - AT_6 - \frac{A}{2}\left((t - t_6) + \frac{T_7}{\pi}\sin(\frac{\pi}{T_7}(t - t_6))\right) & t_6 \leq t \leq t_7 \end{cases} \tag{13}$$

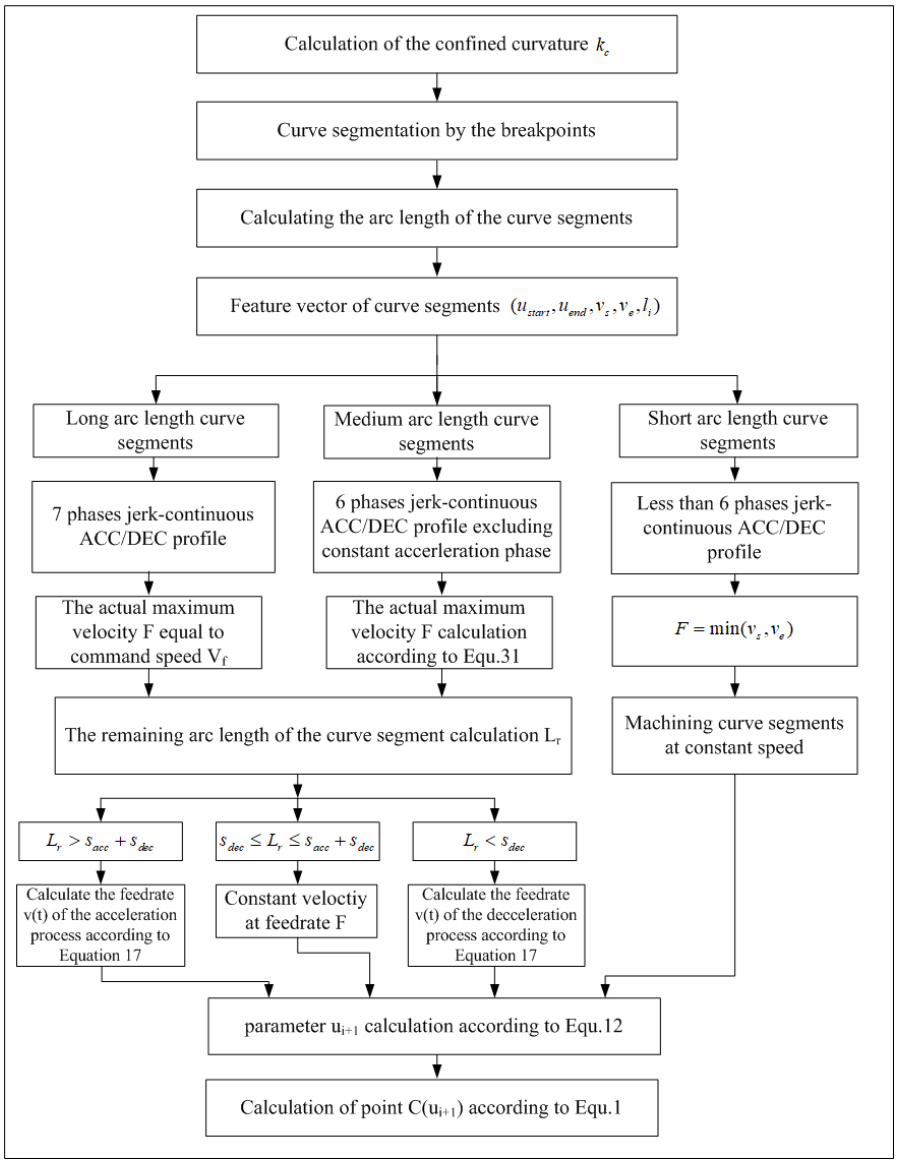

FIGURE 3. The flowchart of the ACC/DEC realization.

$$
\begin{aligned}
L &= \frac{(F+v_s)(T_1+\frac{F-v_s}{A}-T_1+T_3)+(F+v_e)(T_5+\frac{F-v_e}{A}-T_5+T_7)}{2} \\
&= \frac{(F+v_s)(\frac{\pi}{2J}A+\frac{F-v_s}{A})+(F+v_e)(\frac{\pi}{2J}A+\frac{F-v_e}{A})}{2} \\
&= \frac{2F^2+F\frac{\pi A^2}{J}+v_s\frac{\pi A^2}{2J}+v_e\frac{\pi A^2}{2J}-v_s^2-v_e^2}{2A}
\end{aligned} \tag{24}
$$

$$
F = \begin{cases} V_f & \text{if } L > L_{med} \\ \dfrac{-\frac{\pi A^2}{J}+\sqrt{(\frac{\pi A^2}{J})^2-8(v_s\frac{\pi A^2}{2J}+v_s\frac{\pi A^2}{2J}+v_e\frac{\pi A^2}{2J}-v_s^2-v_e^2-2AL)}}{4} & \text{if } L_{\min} \le L \le L_{med} \\ \min(v_s, v_e) & \text{if } L < L_{\min} \end{cases} \tag{29}
$$

4.2.3 The criterion length $L_{\min}$

The criterion length derivation process will be given as follows:

According to Eq.14, we have

$$F = v_3 = v_s + \tfrac{A}{2}T_1 + AT_2 + \tfrac{A}{2}T_3 \tag{30}$$

where $T_1 = T_3 = T_5 = T_7 = \frac{\pi A}{2J} > 0, and\ T_2 \geq 0$

Then:

$$F = v_s + \tfrac{A}{2}T_1 + AT_2 + \tfrac{A}{2}T_3 \geq v_s + AT_1 = v_s + A\tfrac{\pi A}{2J} \tag{31}$$

According to Eq. 28, we can get

$$F = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \geq v_s + \frac{\pi A^2}{2J} \tag{32}$$

where $a = 2, b = \frac{\pi A^2}{J}, c = v_s\frac{\pi A^2}{2J} + v_s\frac{\pi A^2}{2J} + v_e\frac{\pi A^2}{2J} - v_s^2 - v_e^2 - 2AL$

Substituting temporary variables a, b, and c into Equ.32 gives the following inequality

$$L \geq \frac{2\left(v_s + \frac{3\pi A^2}{4J}\right)^2 - \left(v_s - \frac{\pi A^2}{4J}\right)^2 - \left(v_e - \frac{\pi A^2}{4J}\right)^2}{2A} \tag{33}$$

Please refer to Appendix B for the detailed derivation process.

Thus, the criterion length $L_{\min}$ can be represented as

$$L_{\min} = \frac{2\left(v_s + \frac{3\pi A^2}{4J}\right)^2 - \left(v_s - \frac{\pi A^2}{4J}\right)^2 - \left(v_e - \frac{\pi A^2}{4J}\right)^2}{2A} \tag{34}$$

The criterion lengths are given by

$$\begin{cases} L_{med} = \frac{(V_f+v_s)(T_1+T_2+T_3)+(V_f+v_e)(T_5+T_6+T_7)}{2} \\ L_{\min} = \frac{2\left(v_s+\frac{3\pi A^2}{4J}\right)^2-\left(v_s-\frac{\pi A^2}{4J}\right)^2-\left(v_e-\frac{\pi A^2}{4J}\right)^2}{2A} \end{cases} \tag{35}$$

C. ACCELERATION AND DECELERATION REALIZATION

As shown in Fig.3, the confined curvature k_c of the NURBS curve is calculated by substituting the commanded velocity V_f into Eq. 13. Then, the curvature-sensitive regions of the NURBS curve are naturally partitioned by the confined curvature, and the local curvature extreme points can be found as the critical points for each curvature-sensitive area. Finally, the NURBS curve is segmented by critical points, and each curve segment can be expressed as $(u_s, u_e, v_s, v_e, l_i)$.

After that, the interpolator cyclically scans the segmented curve feature vector until all curve segments are processed. Firstly, the acceleration and deceleration of the curve segment are determined by comparing the segment length with the length of the confined curve segment. If $l_i > L_{med}$, the curve segment is long enough to belong to the long curve segment and will contain all 7 phases of the jerk-continuous method, and the feedrate can reach the command velocity. If $L_{\min} \leq l_i \leq L_{med}$, the curve segment belongs to the type of medium segment, the feedrate cannot reach the command velocity, and the actual maximum velocity F should be calculated according to Eq. 28. If $l_i \leq L_{\min}$, the curve segment belongs to the short segment, while the jerk and acceleration cannot reach the maximum value. The minimum feedrate at both ends of the short segment will be kept constant throughout the short curve segment interpolation to reduce the processing complexity.

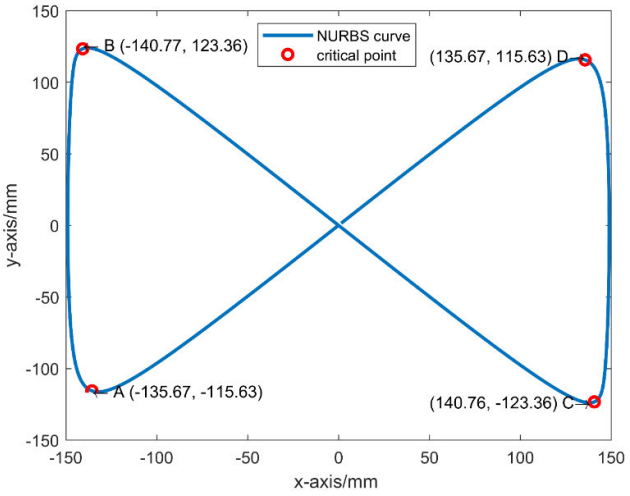

FIGURE 4. The horizontal-8-shaped NURBS curve.

Now, the actual feedrate can be calculated. For both long and medium curve segments, the displacements of the remaining uninterpolated curve segment are first calculated, and the stage in which the acceleration and deceleration of the current interpolation point is located is determined based on the remaining displacements. If $L_r > s_{acc} + s_{dec}$, the velocity $v(t)$ at this point should gradually increase to the end velocity, which should be calculated according to Eq.13. If $s_{dec} \leq L_r \leq s_{acc} + s_{dec}$, it indicates that the velocity of the current interpolation point belongs to the constant velocity phase, while the feedrate $v(t)$ should be maintained at the maximum velocity F of the curve segment. If $L_r < s_{dec}$, it indicates that the velocity of the current interpolation point belongs to the acceleration decreasing stage, while the velocity $v(t)$ at this point should gradually decelerate to the endpoint velocity, which should be calculated according to Eq.13. Besides, for the short segment, the velocity should be maintained at a constant velocity, smaller than the velocity between the beginning and end points.

So far, the feedrate can be obtained for three different curve segments at any interpolation cycle. The parameter u_{i+1} can be calculated from Eq.8.

V. SIMULATION AND ANALYSIS

As shown in Fig. 4 and Fig. 6, horizontal-8-shaped and butterfly-shaped NURBS curves are simulated to verify the effectiveness of the proposed method. Appendix C lists the parameters for the two typical curves, such as degrees, control points, and weight. The interpolation parameters for the simulation are given in Table 2.

TABLE 2. The interpolation parameters for the simulation.

Parameters	Symbol	Units
Command feedrate	V_f	2×10^{-1} mm/ms
Maximum normal ACC	a_n	1×10^{-3} mm/ms^2
Maximum normal jerk	j_n	1×10^{-4} m/ms^3
Maximum tangential ACC	A_t	1×10^{-3} mm/ms^2
Maximum tangential jerk	J_t	1×10^{-4} mm/ms^3
Maximum chord error	δ_{max}	1×10^{-3} mm
Maximum contour error	ε_{max}	5×10^{-2} mm
Interpolation priod	T	2×10^{-3} s

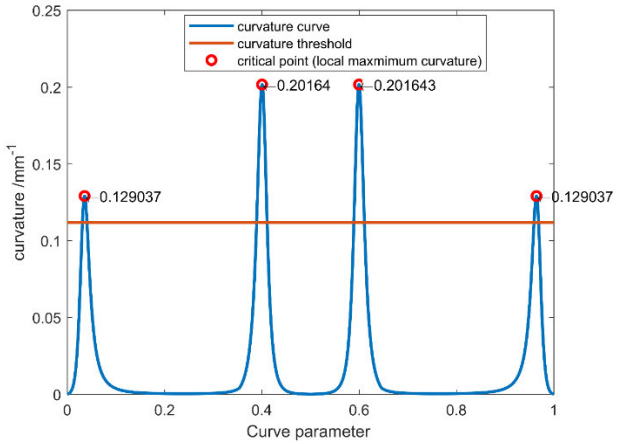

FIGURE 5. The curvature and the critical points of the horizontal-8-shaped NURBS curve.

A. SIMULATION ANALYSIS OF THE HORIZONTAL-8-SHAPED NURBS CURVE

As shown in Fig.4, the curve is divided into four segments by four critical points, marked with A, B, C, and D. Critical points on the curve are marked with ''o'', and their corresponding coordinate values are displayed next to them. The parameter information of the curve segments is shown in Table 3. Fig. 5 shows the curvature corresponding to the horizontal-8-shaped curve. It can be seen that there are four abrupt change areas in the curvature of the curve, each of which represents a curvature-sensitive region. The confined curvature k_c can be obtained by substituting the parameters in Table 2 into Eq.10. The confined curvature is finally calculated as $k_c = 0.112$. This confined curvature is represented by a line parallel to the x-axis in Fig. 5, marked as the curvature threshold, and divides the horizontal-8-shaped curve into four curvature-sensitive regions with curvature higher than the portion above the confined curvature line. The local curvature maximum point of each curvature-sensitive area is regarded as a critical point and marked as ''o'', which corresponds to the critical point in Fig. 4. The number next to each critical point in Fig.5 represents the point curvature.

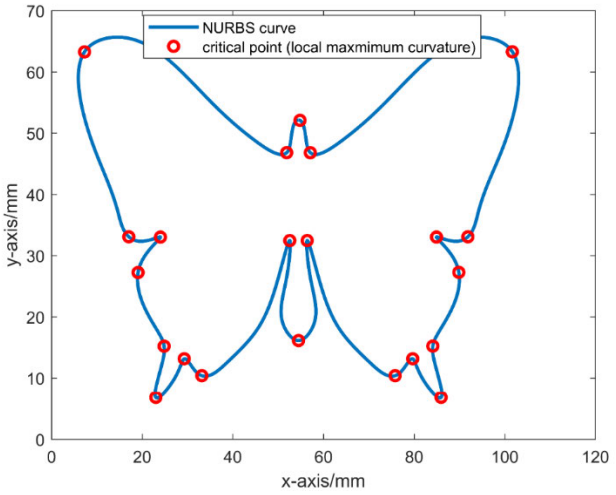

FIGURE 6. The butterfly-shaped NURBS curve.

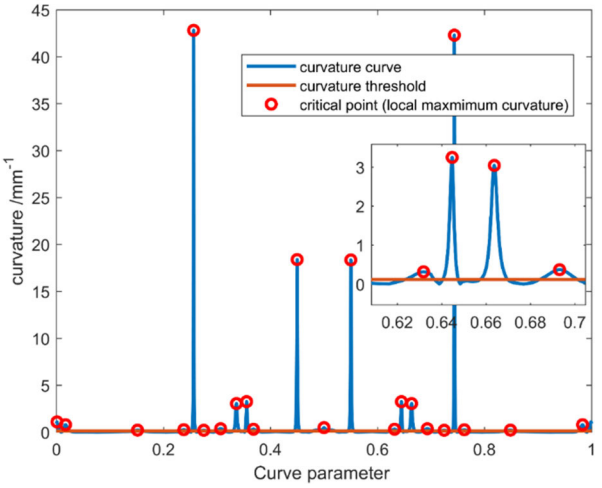

FIGURE 7. The curvature and the critical points of the butterfly-shaped NURBS curve.

The simulation results of the proposed method are shown in Figs.8-15. The proposed method's velocity and tangential acceleration/jerk are well confined to the maximum values. At the same time, the tangential acceleration profile and tangential jerk profile are continuous, indicating that the velocity is smooth naturally. Fig. 11 shows that the chord error is always kept below the maximum chord error during the interpolation, indicating a high machining accuracy. The normal acceleration and normal jerk are also involved in the simulation, as shown in Figs.12-13, and the results indicate that both metrics are within the maximum range. In particular, the normal jerk is significantly lower than the maximum normal jerk. The influence of contour error is also considered in the simulation, and the simulation results are shown in Fig. 14. The results show that the method can well limit the contour error throughout the interpolation process, which is several orders of magnitude smaller than the maximum allowable contour error.

TABLE 3. Segments of the horizontal-8-shaped curve.

SN of the segments	Parameters $(u_s, u_e, v_s, v_e, l_i)$	Max velocity(mm/ms)	Types(L:long,M:midum,S:short)
S0	(u_s =0.000 u_e =0.036 v_s =0.000 v_e =0.088 l_0 =180.735)	0.2	L
S1	(u_s =0.036 u_e =0.400 v_s =0.088 v_e =0.070, l_1 =243.730)	0.2	L
S2	(u_s =0.400 u_e =0.600 v_s =0.088 v_e =0.070, l_2 =378.608)	0.2	L
S3	(u_s =0.600 u_e =1.000 v_s =0.070 v_e =0.000, l_3 =243.764)	0.2	L

TABLE 4. Segments of the butterfly-shaped curve.

SN of the segments	Parameters $(u_s, u_e, v_s, v_e, l_i)$	Max velocity(mm/ms)	Types(L:long,M:midum,S:short)
S_0	(u_s =0.000 u_e =0.001 v_s =0.000 v_e =0.030 l_0 =0.344)	0.017	M
S_1	(u_s =0.001 u_e =0.017 v_s =0.030 v_e =0.036 l_1 =6.103)	0.074	M
S_2	(u_s =0.017 u_e =0.152 v_s =0.036 v_e =0.068 l_2 =50.678)	0.2	L
S_3	(u_s =0.152 u_e =0.238 v_s =0.068 v_e =0.062 l_3 =33.200)	0.183	M
S_4	(u_s =0.238 u_e =0.256 v_s =0.062 v_e =0.005 l_4 =7.142)	0.085	M
S_5	(u_s =0.256 u_e =0.275 v_s =0.005 v_e =0.071 l_5 =7.737)	0.091	M
S_6	(u_s =0.275 u_e =0.307 v_s =0.071 v_e =0.052 l_6 =13.757)	0.122	M
S_7	(u_s =0.307 u_e =0.336 v_s =0.052 v_e =0.018 l_7 =8.731)	0.091	M
S_8	(u_s =0.336 u_e =0.355 v_s =0.018 v_e =0.018 l_8 =9.295)	0.089	M
S_8	(u_s =0.355 u_e =0.368 v_s =0.018 v_e =0.057 l_9 =4.824)	0.070	M
S_9	(u_s =0.368 u_e =0.450 v_s =0.057 v_e =0.007 l_9 =31.123)	0.172	M
S_{10}	(u_s =0.450 u_e =0.500 v_s =0.007 v_e =0.046 l_{10} =18.485)	0.131	M
S_{11}	(u_s =0.500 u_e =0.551 v_s =0.046 v_e =0.007 l_{11} =18.502)	0.131	M
S_{12}	(u_s =0.551 u_e =0.632 v_s =0.007 v_e =0.057 l_{12} =31.115)	0.172	M
S_{13}	(u_s =0.632 u_e =0.645 v_s =0.057 v_e =0.018 l_{13} =4.827)	0.018	M
S_{14}	(u_s =0.645 u_e =0.664 v_s =0.018 v_e =0.018 l_{14} =9.303)	0.089	M
S_{15}	(u_s =0.664 u_e =0.693 v_s =0.018 ve=0.052 l_{15} =8.716)	0.091	M
S_{16}	(u_s =0.693 u_e =0.725 v_s =0.052 v_e =0.071 l_{16} =13.744)	0.121	M
S_{17}	(u_s =0.725 u_e =0.744 v_s =0.071 v_e =0.005 l_{17} =7.757)	0.091	M
S_{18}	(u_s =0.744 u_e =0.762 v_s =0.005 v_e =0.063 l_{18} =7.203)	0.085	M
S_{19}	(u_s =0.762 u_e =0.848 v_s =0.063 v_e =0.068 l_{19} =33.126)	0.183	M
S_{20}	(u_s =0.848 u_e =0.983 v_s =0.068 v_e =0.036 l_{20} =50.712)	0.2	L
S_{21}	(u_s =0.983 u_e =1.000 v_s =0.036 v_e =0.000 l_{21} =6.435)	0.075	M

Besides, Fig.15 shows the velocity planning results using the jerk-continuous method for the long type segment. The horizontal-8-shaped test curve is divided into four long curve segments. Therefore, the paper intercepts one long segment. The long curve segment parameters are ($u_s = 0.036, u_e = 0.400, v_s = 0.088mm/ms, v_e = 0.070mm/ms, l_i = 243.73mm$). As shown in Fig. 15a, the curvature of the tested curve segment is high at both ends and low in the middle.

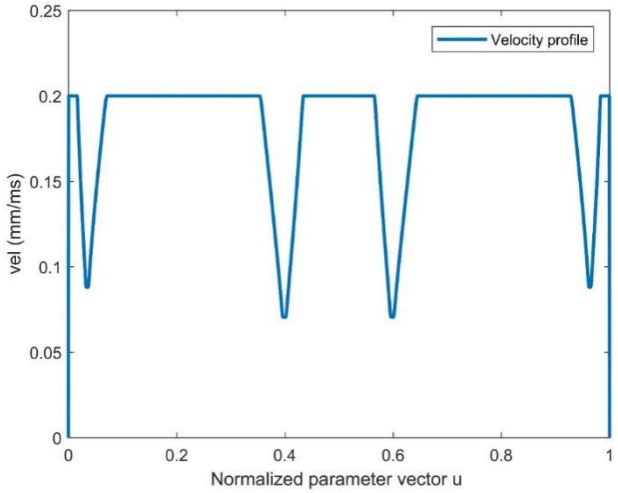

FIGURE 8. The velocity simulation result of the horizontal-8-shaped NURBS curve.

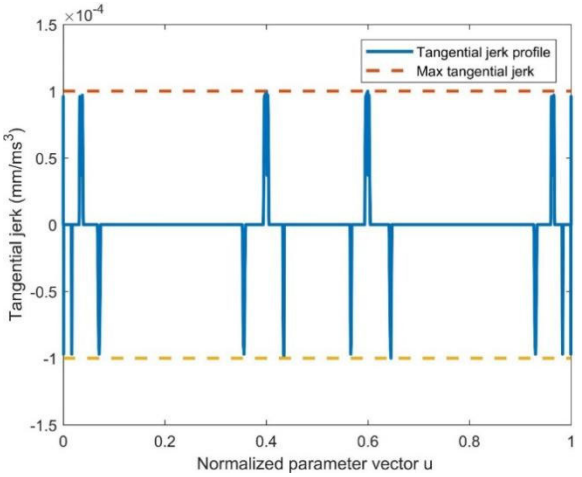

FIGURE 10. The tangential jerk simulation result of the horizontal-8-shaped NURBS curve.

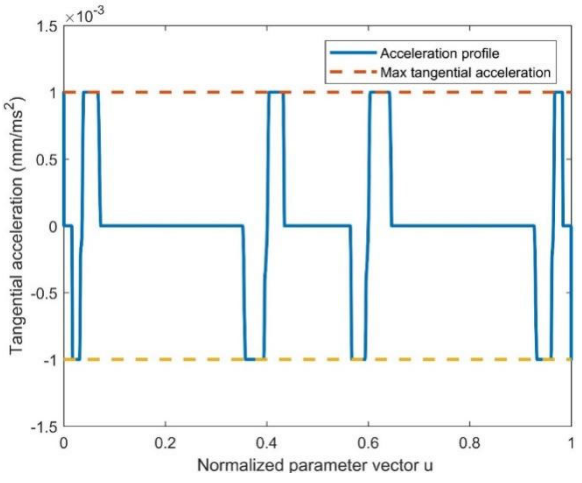

FIGURE 9. The tangential acceleration simulation result of the horizontal-8-shaped NURBS curve.

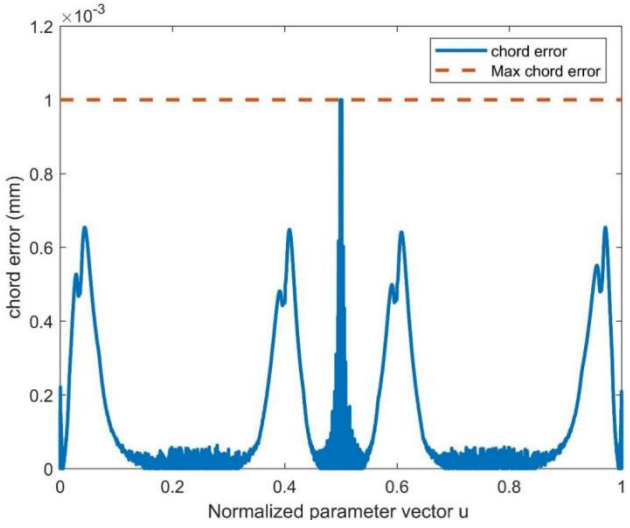

FIGURE 11. The chord error simulation result of the horizontal-8-shaped NURBS curve.

Fig. 15b shows the scheduled velocity profile, including three stages of acceleration, constant velocity, and deceleration, while the velocity profile changes continuously. Fig.15c and Fig. 15d show the corresponding tangential acceleration and jerk curves, all within the maximum allowable value range of the system, and meet the continuous characteristics of a jerk. At the same time, Fig. 15e and Fig. 15f show that the normal acceleration and normal jerk also meet the tolerance. Significantly, the change of the normal jerk is well limited, further reducing the vibration in the normal direction. The chord and contour errors of the curve segment under the jerk-continuous method are shown in Fig. 15g and Fig. 15h. The chord error generally meets the maximum allowable error requirements, the chord error value at the beginning and end of the curve segment is high, with smaller values in the middle part. The contour error is several orders of magnitude smaller than the maximum allowable error, indicating that the machining accuracy can be well guaranteed.

B. SIMULATION ANALYSIS OF THE BUTTERFLY-SHAPED NURBS CURVE

The simulation results of the butterfly-shaped NURBS curve are shown in Fig. 6, while the critical points on the curve are marked with 'o'. The thresholds of the medium segment L_{med} and short segment $L_{\min}$ are 42.922mm and 0.274mm, respectively, which can be calculated by Eq.39. Next, the curve can be divided into 22 segments according to the threshold, of which 2 are long segments and 20 medium segments. Table 4 shows the detailed parameters of each curve segment. The curvature of the butterfly-shaped NURBS curve is listed in Fig. 7. The right middle area in Fig. 7 is a partially enlarged view, indicating some details that are not apparent in the small scaling. The sharp corners with larger curvature are the curvature-sensitive regions in Fig.7. Each curvature-sensitive area's maximum local curvature point is regarded as a critical point and marked as ''o'',

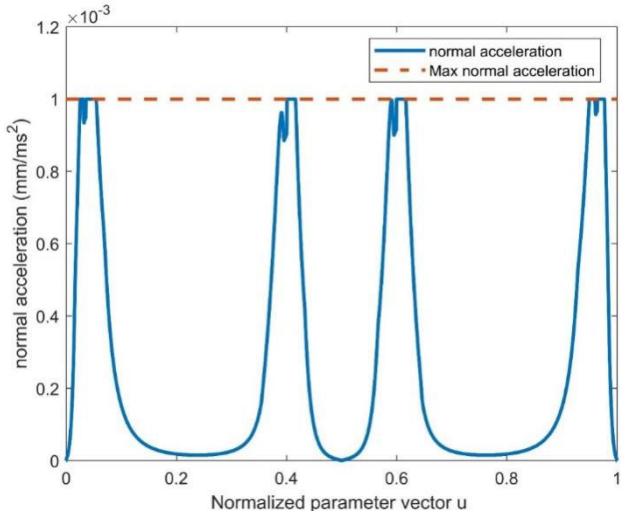

FIGURE 12. The normal acceleration simulation result of the horizontal-8-shaped NURBS curve.

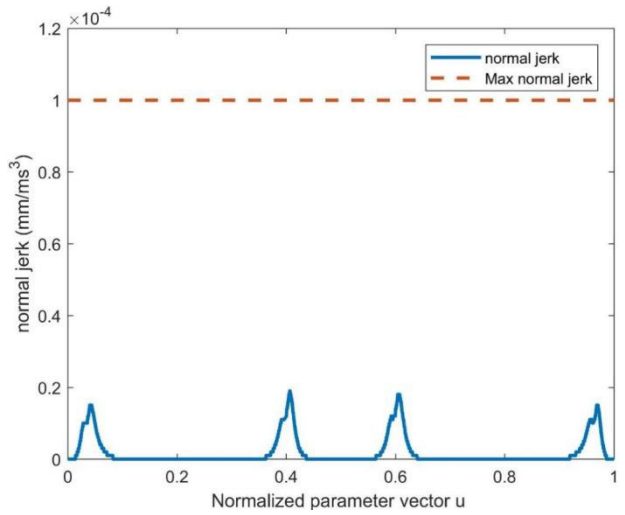

FIGURE 13. The normal jerk simulation result of the horizontal-8-shaped NURBS curve.

which corresponds to the critical point in Fig. 6. Compared with the horizontal 8-shaped curve, the butterfly-shaped curve is divided into more curve segments, and the feedrate scheduling is more difficult.

The simulation results of the proposed method are shown in Figs.16-22. According to the simulation results of the horizontal 8-type curve, Figs. 16-18 show that the proposed method's velocity, tangential acceleration, and tangential jerk values are limited to the maximum values. At the same time, it is clear that the tangential acceleration and jerk profiles are continuous, indicating that the velocity is smooth naturally. The normal acceleration and jerk are always kept below the maximum value in Figs.19-20. The influence of chord and contour errors are also considered in the simulation, and the simulation results are shown in Figs.21-.22. The results indicate that the proposed method can well limit the chord and contour errors in the whole interpolation process.

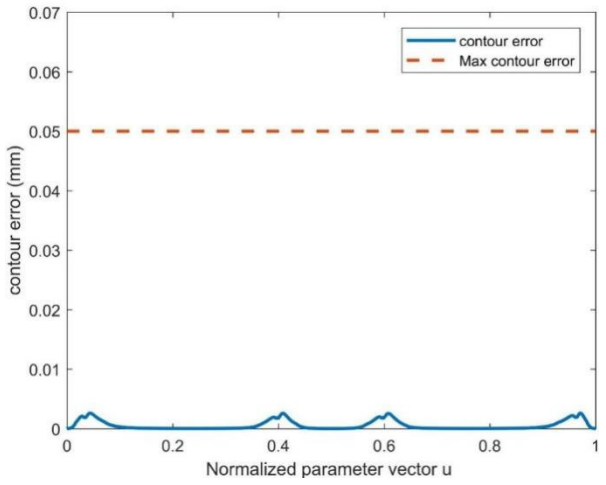

FIGURE 14. The contour error simulation result of the horizontal-8-shaped NURBS curve.

TABLE 5. Comparison of butterfly-shaped curve simulation results.

Interpolation methods	Maximum chord error (*mm*)	Maximum contour error (*mm*)	Computational time (*us*)	Interpolation time (*s*)
JCFS Proposed	1.000e-3 (↓23.7%)	5.700e-3 (↓17.2%)	658.36 (↓14.7%)	8.712 (↓22.8%)
DICT in [16]	1.31e-3	4.862e-2	771.54	11.29
CSFA in [38]	4.819e-2	2.592e-1	-	5.547
DBLA in [39]	3.75e-2	3.450e-1	-	5.397

At the same time, to better observe the simulation results, the results of one curve segment are taken from each of the two types of 22-segment curve segments for display. As shown in Fig. 23, the curve segment is long, corresponding to s2 in Table 4. The detailed parameters of the curve segment can be found in the s2 item. Fig. 23 shows that the acceleration and deceleration processes include seven phases, and the maximum velocity can reach the command velocity. As shown in Fig. 23a, the curvature of the tested curve segment is high at both ends and low in the middle. Fig. 23b shows the feedrate scheduled profile, including three stages of acceleration, constant velocity, and deceleration, while the velocity profile changes continuously. Fig. 23c and Fig. 23d show the corresponding tangential acceleration and jerk curves, all within the maximum allowable value range of the system, and meet the continuous characteristics of the jerk. Fig. 23e and Fig. 23f show that the normal acceleration and jerk also meet the tolerance, especially the change of the normal jerk is well limited. As shown in Fig. 23g and Fig. 23h, the chord and contour errors are also limited.

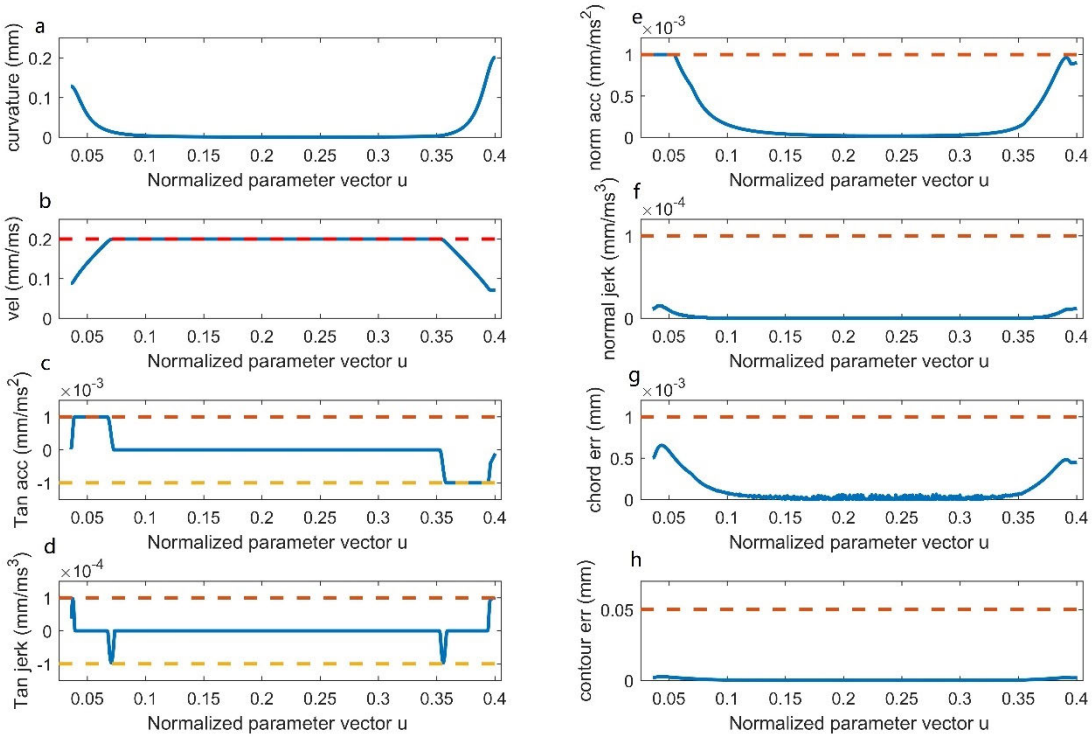

FIGURE 15. The simulation results of the long-length type curve segment of the horizontal-8-shaped NURBS curve. a Curvature. b Scheduled velocity. c Tangential acceleration. d Tangential jerk. e Normal acceleration. F Normal Jerk. g Chord error. h Contour error.

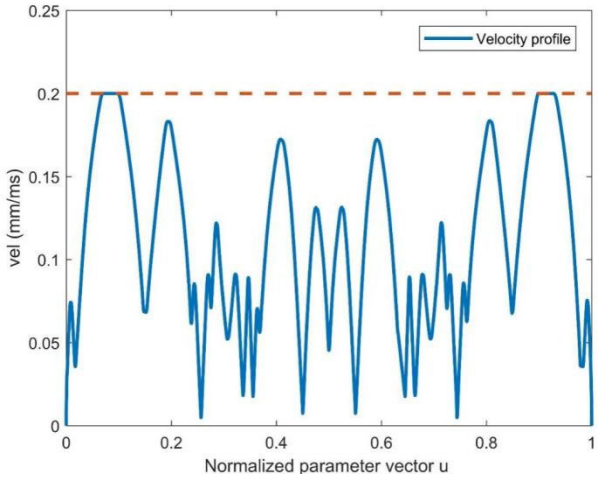

FIGURE 16. The scheduled velocity of butterfly-shaped NURBS curve.

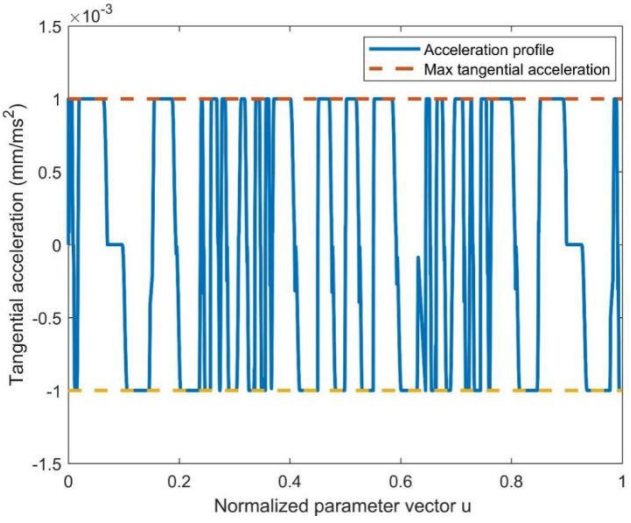

FIGURE 17. The tangential acceleration profile of butterfly-shaped NURBS curve.

Besides, the s12 segment in Table 4 is taken for further discussion, which belongs to the medium segment. Since the length of the curve segment is not enough to accelerate to the commanded velocity, the difference from s2 is inevitable. Besides, the feedrate planning process is somewhat different from that in Fig.23. The most important thing is that the velocity profile does not have a constant velocity phase, and the maximum velocity is less than the command velocity, as shown in Fig. 24b. However, from the simulation results of tangential acceleration and jerk, normal acceleration and jerk, chord and contour errors, the design requirements can be met.

In order to evaluate the effectiveness of the proposed method, it is compared with a double interpolation algorithm based on the cosine theorem (DICT) [16], a complete S-shape feedrate scheduling approach (CSFA) [38], and a dynamic-based interpolator with real-time look-ahead

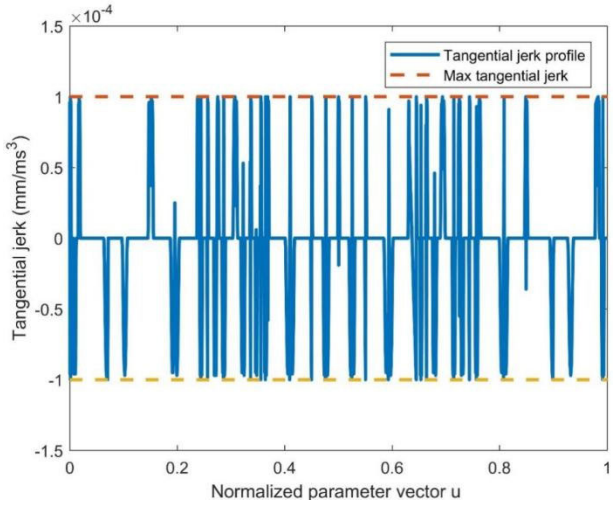

FIGURE 18. The tangential jerk profile of butterfly-shaped NURBS curve.

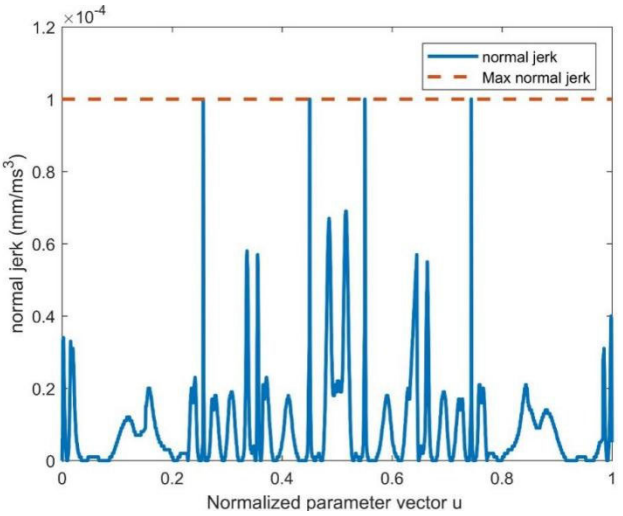

FIGURE 20. The normal jerk profile of butterfly-shaped NURBS curve.

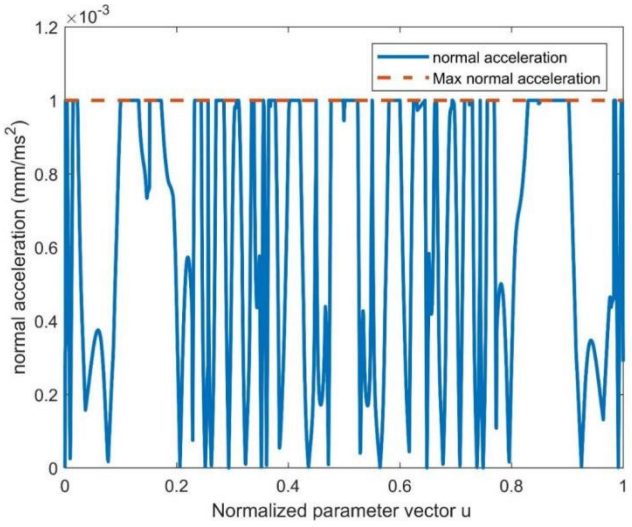

FIGURE 19. The normal acceleration profile of butterfly-shaped NURBS curve.

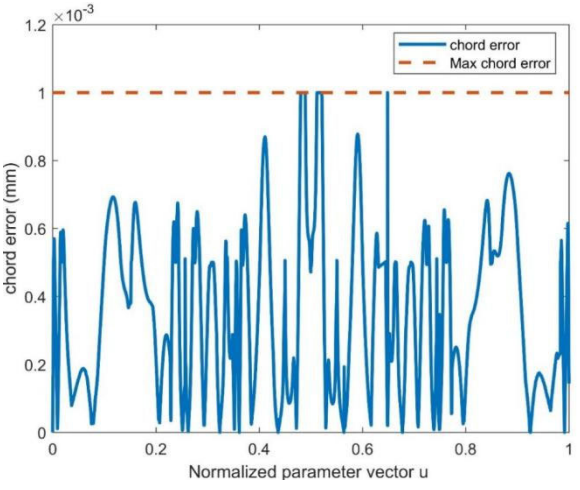

FIGURE 21. The chord error of butterfly-shaped NURBS curve.

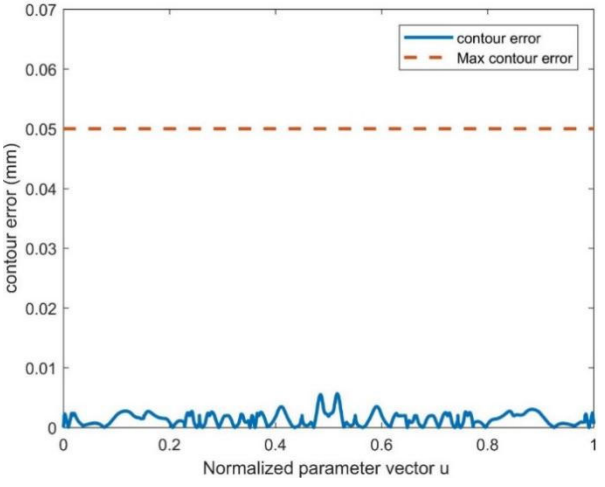

FIGURE 22. The contour error of butterfly-shaped NURBS curve.

algorithm (DBLA) [39]. Several parameters are compared, such as maximum chord error, maximum contour error, computational time [16], and interpolation time. Table 5 shows that although DBLA and CSFA methods consume less computational time than DICT and the proposed method, their maximum chord and contour errors are higher than DICT and the proposed method. Using DICT as the benchmark, the proposed method reduces the maximum chord and contour errors by 23.7% and 17.2%, respectively. The proposed method can achieve the best accuracy and surface quality compared with other methods. At the same time, the computational time of the proposed method is about 658.36*us* for NURBS curve interpolation, which is 14.7% lower than that of DICT, which can better meet the real-time interpolation period (2*ms*) requirement. Moreover, the interpolation time is 8.712s, which is 22.8% lower than that of DICT, providing a better processing efficiency. The proposed method significantly reduces the computational overhead and improves the processing quality.

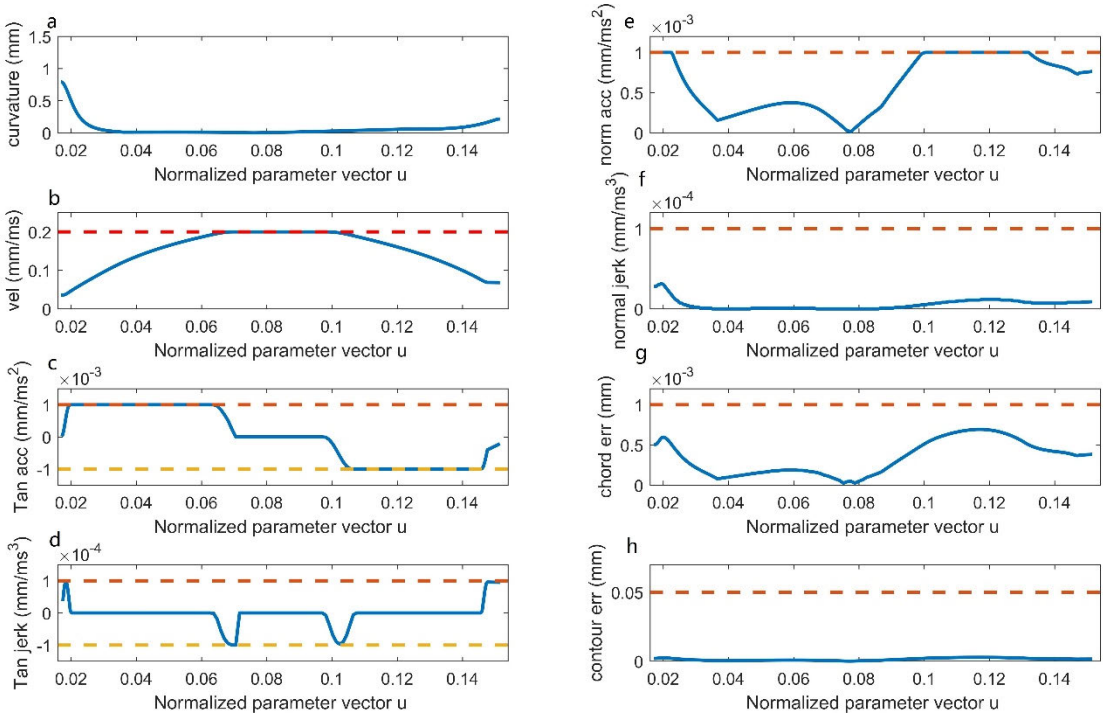

FIGURE 23. The simulation results of the long-length type curve segment of the butterfly-shaped NURBS curve. a Curvature. b Scheduled velocity. c Tangential acceleration. d Tangential jerk. e Normal acceleration. f Normal Jerk. g Chord error. h Contour error.

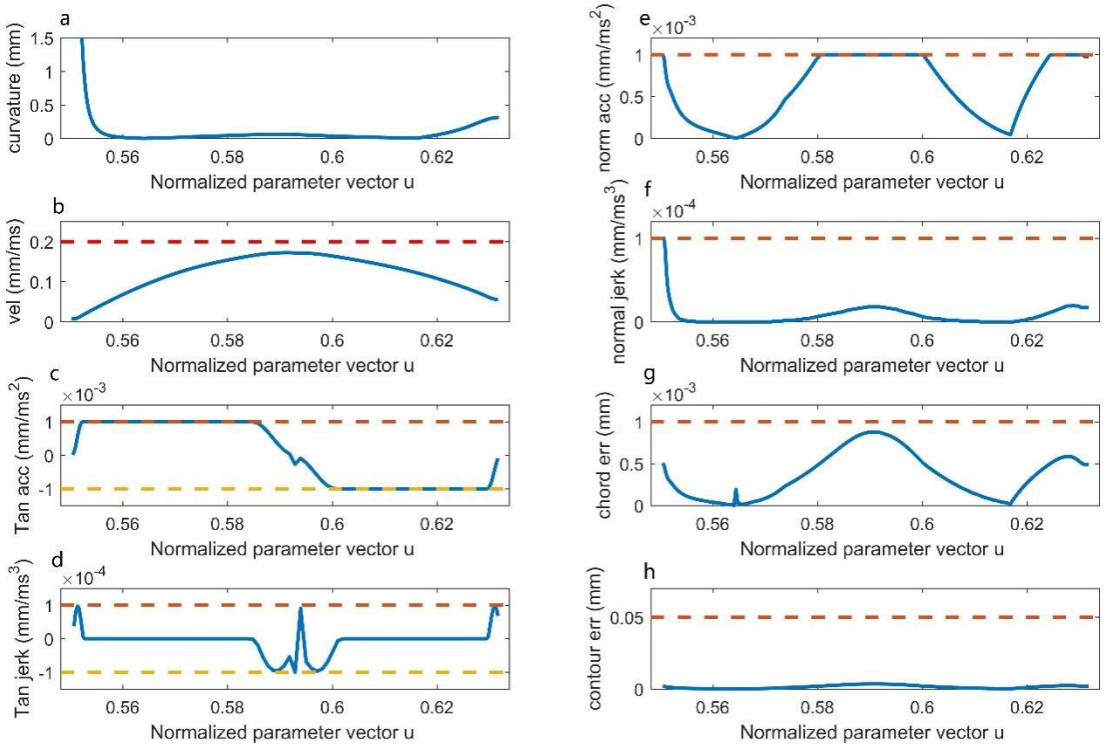

FIGURE 24. The simulation results of the medium-length type curve segment of the butterfly-shaped NURBS curve. a Curvature. b Scheduled velocity. c Tangential acceleration. d Tangential jerk. e Normal acceleration. f Normal Jerk. g Chord error. h Contour error.

The simulation results indicate that the proposed method has reached the expectation in all aspects for more complex butterfly-shaped curves, demonstrating the feasibility and effectiveness of the proposed method.

C. BRIEF DISCUSSION ON THE JERK LIMITATION

The proposed method considers the normal and tangential jerk limitations during interpolation. The proposed method obtains the confined curvature through maximum chord error, maximum contour errors, maximum normal acceleration and normal jerk and obtains the curvature-sensitive areas. The above parameters (including the maximum normal jerk) restrict the feedrate of each point in the curvature-sensitive area so that the maximum normal jerk tolerance can be met at any time during the interpolation process, as shown in Fig. 13 and Fig. 20. At the same time, the trigonometric-function-based jerk-continuous method is utilized to confine the tangential jerk, so that the tangential jerk meets the maximum tangential jerk tolerance and maintains the continuity of the tangential jerk everywhere, as shown in Fig. 10 and Fig. 18.

However, there are some differences in different interpolation curves, such as the simulated horizontal-8-shaped NURBS and butterfly-shaped NURBS curves. Since the curvature profile of the former curve is much lower than that of the latter, as shown in Fig. 5 and Fig. 7, the actual normal jerk profile of the former is much lower than the maximum normal jerk tolerance (as shown in Fig. 13). In contrast, the actual normal jerk profile of the latter is larger than the former in terms of the fluctuation range and amplitude (as shown in Fig. 20). In terms of a tangential jerk, since the former has fewer curvature sensitive areas than the latter, the change in the actual tangential jerk profile of the former is simpler than that of the latter (as shown in Fig. 10 and Fig. 18).

VI. CONCLUSION

In this paper, a jerk-continuous feedrate optimization method is proposed. Firstly, the entire trajectory is subdivided into segments based on the curve segmentation idea. Then, curve segment classification and threshold length calculation methods are constructed to reduce the complexity of feedrate optimization. Next, the analytic expression of the maximum achievable feedrate for different-length curve segments is also established. Finally, two typical horizontal-8-shaped and butterfly-shaped NURBS curves are simulated. The results indicate that the chord and contour errors can be kept within the given tolerance. Besides, the values of normal acceleration/jerk and tangential acceleration/jerk are confined to match the machine tool's dynamic capabilities. Compared with the DICT method, the computational and interpolation time are reduced by 17.2% and 22.8%, respectively, demonstrating the method's feasibility and effectiveness. Further research should be performed to apply the proposed method to five-axis machine tools, which will face more significant challenges.

APPENDIX A

$$2F^2 + F\frac{\pi A^2}{J} + v_s\frac{\pi A^2}{2J} + v_e\frac{\pi A^2}{2J} - v_s^2 - v_e^2 - 2AL = 0$$

Let $a = 2$, $b = \frac{\pi A^2}{J}$, $c = v_s\frac{\pi A^2}{2J} + v_s\frac{\pi A^2}{2J} + v_e\frac{\pi A^2}{2J} - v_s^2 - v_e^2 - 2AL$

TABLE 6. The horizontal-8-shaped curve parameters.

Parameters	Values
Control Points	{0,0},{-150,-150},{-150,150},{0,0},{150,-150},{150,150},{0,0}
Weights	{1,25,25,1,25,25,1};
knot vector	{0,0,0,0,0.354,0.5,0.646,1,1,1,1};

TABLE 7. The parameters of the butterfly-shaped curve.

Parameters	Values
Control Points	{54.493, 52.139}, {55.507, 52.139}, {56.082, 49.615},{56.780, 44.971}, {69.575, 51.358}, {77.786, 58.573},{90.526, 67.081}, {105.973, 63.801}, {100.400, 47.326},{94.567, 39.913}, {92.369, 30.485}, {83.440, 33.757}, {91.892, 28.509}, {89.444, 20.393}, {83.218, 15.446},{87.621, 4.830}, {80.945, 9.267}, {79.834, 14.535},{76.074, 8.522}, {70.183, 12.550}, {64.171, 16.865},{59.993, 22.122}, {55.680, 36.359}, {56.925, 24.995}, {59.765, 19.828}, {54.493, 14.940}, {49.220, 19.828},{52.060, 24.994}, {53.305, 36.359}, {48.992, 22.122},{44.814, 16.865}, {38.802, 12.551}, {32.911, 8.521},{29.152, 14.535}, {28.040, 9.267}, {21.364, 4.830}, {25.768, 15.447}, {19.539, 20.391}, {17.097, 28.512},{25.537, 33.750}, {16.602, 30.496}, {14.199, 39.803},{8.668, 47.408}, {3.000, 63.794}, {18.465, 67.084},{31.197, 58.572}, {39.411, 51.358}, {52.204, 44.971}, {52.904, 49.614} , {53.478, 52.139}, {54.493, 52.139}
Weights	{0, 0, 0, 0, 0.0083, 0.0150, 0.0361, 0.0855, 0.1293, 0.1509, 0.1931, 0.2273, 0.2435, 0.2561,0.2692, 0.2889, 0.3170, 0.3316, 0.3482, 0.3553, 0.3649, 0.3837, 0.4005, 0.4269, 0.4510, 0.4660, 0.4891, 0.5000,0.5109, 0.5340, 0.5489, 0.5731, 0.5994, 0.6163, 0.6351, 0.6447, 0.6518, 0.6683, 0.6830, 0.7111, 0.7307, .7439,0.7565, 0.7729, 0.8069, 0.8491, 0.8707, 0.9145, 0.9639, 0.9850, 0.9917, 1.0, 1.0, 1.0, 1.0};
knot vector	{1.0, 1.0, 1.0, 1.2, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 5.0, 3.0, 1.0, 1.1, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.1, 1.0, 3.0, 5.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.2, 1.0, 1.0, 1.0};

The quadratic equation for F can be described as:

$$aF^2 + bF + c = 0$$

$$\begin{aligned}\Delta &= b^2 - 4ac \\ &= (\tfrac{\pi A^2}{J})^2 - 4 * 2 * (v_s\tfrac{\pi A^2}{2J} + v_e\tfrac{\pi A^2}{2J} - v_s^2 - v_e^2 - 2AL) \\ &= \frac{\pi^2 A^4}{J^2} - 8(v_s(\frac{\pi A^2}{2J} - v_s) + v_e(\frac{\pi A^2}{2J} - v_e)) + 16AL \\ &= 8(v_s^2 - \frac{1}{2}v_s\frac{\pi A^2}{J}) + 8(v_e^2 - \frac{1}{2}v_e\frac{\pi A^2}{J}) + \frac{\pi^2 A^4}{J^2} + 16AL \\ &= 8(v_s^2 - \frac{1}{2}v_s\frac{\pi A^2}{J} + \frac{\pi^2 A^4}{16J^2})\end{aligned}$$

$$
\begin{aligned}
&\Rightarrow -8(v_s(AT_1 - v_s) + v_e(AT_5 - v_e) - 2AL) \geq 16(v_s + \frac{\pi A^2}{2J})^2 + 8A(T_1 + T_5)(v_s + \frac{\pi A^2}{2J}) \\
&\Rightarrow 16AL \geq 16(v_s + \frac{\pi A^2}{2J})^2 + 8A(T_1 + T_5)(v_s + \frac{\pi A^2}{2J}) + 8(v_s(AT_1 - v_s) + v_e(AT_5 - v_e)) \\
&\Rightarrow L \geq \frac{(v_s + \frac{\pi A^2}{2J})^2 + \frac{1}{2}A(T_1 + T_5)(v_s + \frac{\pi A^2}{2J})}{A} + \frac{(v_s(AT_1 - v_s) + v_e(AT_5 - v_e))}{2A} \\
&\Rightarrow L \geq \frac{(v_s + \frac{\pi A^2}{2J})^2 + \frac{1}{2}A(\frac{\pi A}{2J} + \frac{\pi A}{2J})(v_s + \frac{\pi A^2}{2J})}{A} + \frac{(v_s(A\frac{\pi A}{2J} - v_s) + v_e(A\frac{\pi A}{2J} - v_e))}{2A} \\
&\Rightarrow L \geq \frac{(v_s + \frac{\pi A^2}{2J})^2 + \frac{1}{2}\frac{\pi A^2}{J}(v_s + \frac{\pi A^2}{2J}) + \left(\frac{\pi A^2}{4J}\right)^2 - \left(\frac{\pi A^2}{4J}\right)^2}{A} \\
&\quad - \frac{\left(v_s^2 - v_s\frac{\pi A^2}{2J} + \left(\frac{\pi A^2}{4J}\right)^2 - \left(\frac{\pi A^2}{4J}\right)^2\right) + \left(v_e^2 - v_e\frac{\pi A^2}{2J} + \left(\frac{\pi A^2}{4J}\right)^2 - \left(\frac{\pi A^2}{4J}\right)^2\right)}{2A} \\
&\Rightarrow L \geq \frac{\left((v_s + \frac{\pi A^2}{2J}) + \frac{\pi A^2}{4J}\right)^2 - \left(\frac{\pi A^2}{4J}\right)^2}{A} - \frac{\left(v_s - \frac{\pi A^2}{4J}\right)^2 - \left(\frac{\pi A^2}{4J}\right)^2 + \left(v_e - \frac{\pi A^2}{4J}\right)^2 - \left(\frac{\pi A^2}{4J}\right)^2}{2A} \\
&\Rightarrow L \geq \frac{2\left((v_s + \frac{\pi A^2}{2J}) + \frac{\pi A^2}{4J}\right)^2 - \left(v_s - \frac{\pi A^2}{4J}\right)^2 - \left(v_e - \frac{\pi A^2}{4J}\right)^2 - 2\left(\frac{\pi A^2}{4J}\right)^2 + 2\left(\frac{\pi A^2}{4J}\right)^2}{2A} \\
&\Rightarrow L \geq \frac{2\left(v_s + \frac{3\pi A^2}{4J}\right)^2 - \left(v_s - \frac{\pi A^2}{4J}\right)^2 - \left(v_e - \frac{\pi A^2}{4J}\right)^2}{2A} \\
&L_{\min} = \frac{2\left(v_s + \frac{3\pi A^2}{4J}\right)^2 - \left(v_s - \frac{\pi A^2}{4J}\right)^2 - \left(v_e - \frac{\pi A^2}{4J}\right)^2}{2A}
\end{aligned}
$$

$$
\begin{aligned}
&+ 8(v_e^2 - \frac{1}{2}v_e\frac{\pi A^2}{J} + \frac{\pi^2 A^4}{16J^2}) + 16AL \\
&= 8(v_s - \frac{1}{4}\frac{\pi A^2}{J})^2 + 8(v_e - \frac{1}{4}\frac{\pi A^2}{J})^2 + 16AL
\end{aligned}
$$

APPENDIX B

$$
\begin{aligned}
&\frac{-b + \sqrt{b^2 - 4ac}}{2a} \geq v_s + \frac{\pi A^2}{2J} \\
&\Rightarrow \sqrt{b^2 - 4ac} \geq 2a(v_s + \frac{\pi A^2}{2J}) + b \\
&\Rightarrow b^2 - 4ac \geq 4a^2(v_s + \frac{\pi A^2}{2J})^2 + 4ab(v_s + \frac{\pi A^2}{2J}) + b^2 \\
&\Rightarrow -4ac \geq 4a^2(v_s + \frac{\pi A^2}{2J})^2 + 4ab(v_s + \frac{\pi A^2}{2J})
\end{aligned}
$$

where $a = 2$, $b = \frac{\pi A^2}{J}$, $c = v_s\frac{\pi A^2}{2J} + v_s\frac{\pi A^2}{2J} + v_e\frac{\pi A^2}{2J} - v_s^2 - v_e^2 - 2AL$, then the equation can be derived, as shown at the top of the page.

APPENDIX C

See Table 6 and 7.

REFERENCES

[1] X. Zhiming, C. Jincheng, and F. Zhengjin, "Performance evaluation of a real-time interpolation algorithm for NURBS curves," *Int. J. Adv. Manuf. Technol.*, vol. 20, no. 4, pp. 270–276, Aug. 2002.

[2] 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.

[3] 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 Des.*, vol. 43, pp. 612–628, Jun. 2011.

[4] 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.

[5] 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.

[6] M. Shpitalni, Y. Koren, and C. C. Lo, "Realtime curve interpolators," *Comput.-Aided Des.*, vol. 26, no. 11, pp. 832–838, Nov. 1994.

[7] D. Li, W. Zhang, W. Zhou, T. Shang, and J. Fleischer, "Dual NURBS path smoothing for 5-Axis linear path of flank milling," *Int. J. Precis. Eng. Manuf.*, vol. 19, no. 12, pp. 1811–1820, Dec. 2018.

[8] Y. Sang, C. Yao, Y. Lv, and G. He, "An improved feedrate scheduling method for NURBS interpolation in five-axis machining," *Precis. Eng.*, vol. 64, pp. 70–90, Jul. 2020.

[9] L. Chen, Z. Wei, and L. Ma, "Five-axis tri-NURBS spline interpolation method considering compensation and correction of the nonlinear error of cutter contacting paths," *Int. J. Adv. Manuf. Technol.*, vol. 119, pp. 2043–2057, Dec. 2021.

[10] X. Wang, B. Liu, X. Mei, D. Hou, Q. Li, and Z. Sun, "Global smoothing for five-axis linear paths based on an adaptive NURBS interpolation algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 114, nos. 7–8, pp. 2407–2420, Jun. 2021.

[11] D. C. Yang and T. Kong, "Parametric interpolator versus linear interpolator for precision CNC machining," *Comput.-Aided Des.*, vol. 26, no. 3, pp. 225–234, 1994.

[12] K. Erkorkmaz and Y. Altintas, "Quintic spline interpolation with minimal feed fluctuation," *J. Manuf. Sci. Eng.*, vol. 127, no. 2, pp. 339–349, May 2005.

[13] 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.
[14] 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.
[15] Y. Sun, Y. Zhao, Y. Bao, and D. Guo, "A novel adaptive-feedrate interpolation method for NURBS tool path with drive constraints," *Int. J. Mach. Tools Manuf.*, vol. 77, pp. 74–81, Feb. 2014.
[16] T.-Y. Wang, Y.-B. Zhang, J.-C. Dong, R.-J. Ke, and Y.-Y. Ding, "NURBS interpolator with adaptive smooth feedrate scheduling and minimal feedrate fluctuation," *Int. J. Precis. Eng. Manuf.*, vol. 21, no. 2, pp. 273–290, Feb. 2020.
[17] 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.
[18] H. Ni, C. Zhang, S. Ji, T. Hu, Q. Chen, Y. Liu, and G. Wang, "A bidirectional adaptive feedrate scheduling method of NURBS interpolation based on S-shaped ACC/DEC algorithm," *IEEE Access*, vol. 6, pp. 63794–63812, 2018.
[19] P. Guo, Y. Wu, G. Yang, Z. Shen, H. Zhang, P. Zhang, F. Lou, and H. Li, "A feedrate planning method for the NURBS curve in CNC machining based on the critical constraint curve," *Appl. Sci.*, vol. 11, no. 11, p. 4959, May 2021.
[20] J. Jahanpour and M. R. Alizadeh, "A novel acc-jerk-limited NURBS interpolation enhanced with an optimized S-shaped quintic feedrate scheduling scheme," *Int. J. Adv. Manuf. Technol.*, vol. 77, nos. 9–12, pp. 1889–1905, Apr. 2015.
[21] Y. Sun, M. Chen, J. Jia, Y.-S. Lee, and D. Guo, "Jerk-limited feedrate scheduling and optimization for five-axis machining using new piecewise linear programming approach," *Sci. China Technol. Sci.*, vol. 62, no. 7, pp. 1067–1081, Jul. 2019.
[22] K. Erwinski, R. Szczepanski, and T. Tarczewski, "Nature inspired optimization of jerk limited feedrate profile for NURBS toolpaths in CNC machines," in *Proc. IOP Conf., Mater. Sci. Eng.*, 2021, vol. 1140, no. 1, Art. no. 012031.
[23] K. Erwinski, A. Wawrzak, and M. Paprocki, "Real-time jerk limited feedrate profiling and interpolation for linear motor multiaxis machines using NURBS toolpaths," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7560–7571, Nov. 2022.
[24] P. Boscariol, A. Gasparetto, and R. Vidoni, "Planning continuous-jerk trajectories for industrial manipulators," in *Proc. ASME 11th Biennial Conf. Eng. Syst. Design Anal.*, vol. 44861, 2012, pp. 127–136.
[25] Y. Fang, J. Qi, J. Hu, W. Wang, and Y. Peng, "An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints," *Mechanism Mach. Theory*, vol. 153, Nov. 2020, Art. no. 103957.
[26] G. Y. Zhao, Y. G. Zhao, and R. G. Hou, "Research on motion profile smooth control algorithm based on continuous jerk," in *Applied Mechanics and Materials*. New York, NY, USA: Trans Tech Publ, vol. 29, 2010, pp. 2002–2007.
[27] L. Xinhua, P. Junquan, S. Lei, and W. Zhongbin, "A novel approach for NURBS interpolation through the integration of acc-jerk-continuous-based control method and look-ahead algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 88, nos. 1–4, pp. 961–969, Jan. 2017.
[28] Y. Zhang, P. Ye, H. Zhang, and M. Zhao, "A local and analytical curvature-smooth method with jerk-continuous feedrate scheduling along linear toolpath," *Int. J. Precis. Eng. Manuf.*, vol. 19, no. 10, pp. 1529–1538, Oct. 2018.
[29] H. Ni, J. Yuan, S. Ji, C. Zhang, and T. Hu, "Feedrate scheduling of NURBS interpolation based on a novel jerk-continuous ACC/DEC algorithm," *IEEE Access*, vol. 6, pp. 66403–66417, 2018.
[30] Y. Zhang, M. Zhao, P. Ye, and H. Zhang, "A G4 continuous B-spline transition algorithm for CNC machining with jerk-smooth feedrate scheduling along linear segments," *Comput.-Aided Des.*, vol. 115, pp. 231–243, Oct. 2019.
[31] J. Yang, D. Li, C. Ye, and H. Ding, "An analytical C^3 continuous tool path corner smoothing algorithm for 6R robot manipulator," *Robot. Comput.-Integr. Manuf.*, vol. 64, Aug. 2020, Art. no. 101947.
[32] H. Sun, J. Yang, D. Li, and H. Ding, "An on-line tool path smoothing algorithm for 6R robot manipulator with geometric and dynamic constraints," *Sci. China Technol. Sci.*, vol. 64, no. 9, pp. 1907–1919, Sep. 2021.
[33] D. N. Song, Z. Y. Jia, J. W. Ma, N. Zhang, and L. K. Si, "A precise direct parametric interpolation method for NURBS-toolpath generation," *Int. J. Ind. Syst. Eng.*, vol. 31, no. 2, pp. 278–286, 2019.
[34] D. Du, Y. Liu, X. Guo, K. Yamazaki, and M. Fujishima, "An accurate adaptive NURBS curve interpolator with real-time flexible acceleration/deceleration control," *Robot. Comput.-Integr. Manuf.*, vol. 26, no. 4, pp. 273–281, Aug. 2010.
[35] M. Nie, Y. Wan, and A. Zhou, "Real-time NURBS interpolation under multiple constraints," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–15, May 2022.
[36] J. Huang and L.-M. Zhu, "Feedrate scheduling for interpolation of parametric tool path using the sine series representation of jerk profile," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 231, no. 13, pp. 2359–2371, Nov. 2017.
[37] Y. Hu, X. Jiang, G. Huo, C. Su, B. Wang, H. Li, and Z. Zheng, "A novel S-shape based NURBS interpolation with acc-jerk- continuity and round-off error elimination," 2021, *arXiv:2103.14433*.
[38] X. Du, J. Huang, and L.-M. Zhu, "A complete S-shape feed rate scheduling approach for NURBS interpolator," *J. Comput. Design Eng.*, vol. 2, no. 4, pp. 206–217, Oct. 2015.
[39] M.-T. Lin, M.-S. Tsai, and H.-T. Yau, "Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 15, pp. 2246–2262, Dec. 2007.

MINGXING NIE received the bachelor's degree in computer science, the master's degree in computer system structure, and the Ph.D. degree in traffic information engineering and control from Central South University, Hunan, China, in 2004, 2007, and 2015, respectively. He is currently an Associate Professor with the School of Computer Science, University of South China. His current research interests include computer numerical control, NURBS interpolation, optimization technique, motor control, and the IoT.

LIWEI ZOU received the bachelor's degree in software engineering from Sichuan Normal University, Sichuan, China, in 2022. He is currently pursuing the master's degree in electronic information with the School of Computer Science, University of South China. His current research interests include optimization techniques and the Internet of Things.

TAO ZHU (Member, IEEE) received the B.E. degree from Central South University, in 2009, and the Ph.D. degree from the University of Science and Technology of China, in 2015. He was a Postdoctoral Researcher and a Lecturer with the School of Computer and Communication Engineering, University of Science and Technology Beijing. He is currently an Associate Professor with the University of South China. His research interests include evolutionary computation and the Internet of Things.