

Research Article

Real-Time NURBS Interpolation under Multiple Constraints

Mingxing Nie , Yaping Wan , and Aijun Zhou 

School of Computer Science, University of South China, Hengyang 421001, China

Correspondence should be addressed to Yaping Wan; 2019000006@usc.edu.cn

Received 19 March 2022; Accepted 22 April 2022; Published 17 May 2022

Academic Editor: Kapil Sharma

Copyright © 2022 Mingxing Nie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

NURBS interpolation is superior to traditional linear or circular interpolation in terms of code size, surface quality, and machining efficiency. However, with the increasing demands for high-accuracy and efficient machining, NURBS interpolation has faced a growing number of challenges. Many researchers are actively involved in this field with great interest. Due to the special form of NURBS curve, there is a nonlinear relationship between its curve and arc length; feed fluctuations and mechanical shocks which are caused during the interpolation process will seriously affect the surface accuracy and quality of machined parts. To solve these problems, a real-time NURBS interpolation is proposed under multiple constraints (RNIC) in this paper. First, the formulas of the constrained feedrate under geometric errors, kinematic constraints, drive constraints, and contour errors are given. Then, the two stages for the proposed interpolation are established. The former stage is offline preprocessing stage, which aims to quickly find feedrate sensitive areas (FSAs), while the latter online stage is the real-time interpolation, which is responsible for smoothing the velocity. In the preprocessing stage, we utilized FSA scan module and feedrate adjustment module to detect the FSAs and adjust the feedrate at the start/end of each subsegment by a bidirectional scanning algorithm. Each segment contains acceleration and deceleration (some contains uniform speed) stages, which can be well matched with the processing process of acceleration and deceleration. Finally, according to the proposed method and the adaptive speed adjustment method, the simulation of a “butterfly-shaped” NURBS curve using the S-shaped ACC/DEC algorithm is carried out, which verifies the reliability and effectiveness of the proposed algorithm.

1. Introduction

As one of the international industrial standards of CAD system, NURBS curves play an increasing role in the application fields of computer-aided geometric design (CAGD) with the wide application of this geometric model. Profiting from the ability of changing the shape of local contours and describing complex contours by low-order equations, NURBS interpolations have been increasingly developed in CNC machining. NURBS interpolation is superior to traditional linear and circular interpolation in terms of code size, surface quality, and machining efficiency. However, there are still some challenges in building a successful NURBS interpolator. First, the feedrate along the machining path may be limited by many factors including geometrical, kinematic, and axis dynamics constraints in large curvature areas; these

constraints should be reasonably and smoothly adjusted to ensure good machining quality and accuracy. Second, abrupt curvature change will cause ACC/DEC and jerk out of tolerance. Last, there is no mathematical expression between the NURBS curve parameters and displacement, which inevitably leads to a fluctuation between the expected and actual feedrates.

In order to improve machining accuracy and efficiency, Yeh and Hsu [1] adaptively adjust the feedrate considering the constrained chord error, which improves the contour accuracy of parametric curve interpolation. Yong and Narayanaswami [2] proposed the concept of feedrate sensitive corners to confine acceleration and deceleration in offline processing. Nam and Liu [3, 4] further limited the effect of jerk on machining in NURBS interpolation. While these efforts are effective, many key constraints are overlooked; various advanced algorithms have been further

developed by many researchers. For example, Sencer et al. [5] solved the analytical solution of feedrate in the form of cubic B-spline, but the calculation efficiency of this process is low. Sun et al. [6] proposed a feedrate scheduling method based on drive constraints. Xavier et al. [7] developed a velocity profile optimization method based on iterative calculation of intersection points with given constraints. However, this method ignores the influence of contour accuracy. Jia et al. [8] proposed a NURBS interpolator under kinetic characteristics and contour error constraints, in which the feedrate remained constant in most areas and only changed smoothly in areas with large curvature, but the production efficiency was degraded when utilizing a constant low feedrate without making full use of the motion performance in this method. Ma et al. [9] also considered, from the perspective of axis restriction, maintaining constant feedrate in the sensitive area and smooth transition in the non-sensitive area. Liu et al. [10] used a linear mathematical model to explore the relationship between feedrate and geometric constraints. Liu et al. [11] also developed a new feedrate optimization method under multiple constraints and finally fitted to the smooth spline curve. However, these two methods only considered the influence of geometric and axis dynamics errors. Zhong et al. [12] further limited the axial velocity, axial acceleration, jerk, and contour errors through a look-ahead module.

Moreover, to mitigate the feed fluctuation to further improve the machining accuracy, many methods have been proposed for NURBS interpolation. Liu et al. [13] proposed a NURBS interpolation using approximation errors, but the calculation coefficient polynomial equation is time-consuming. Liu et al. [14] proposed a highly accurate interpolation method by establishing an improved quartic equation related to the curve parameter increment to reduce the feedrate fluctuation caused by approximation errors. However, this method obtains a quartic equation by ignoring the higher-order term (order >4), which has limited adaptability. Ni et al. [15] proposed a round-off error-free feedrate scheduling method. However, this method limits the feedrate to meet the various constraints by increasing the interpolation time, which is not efficient. Song et al. [16] compensates the interpolation parameters in order to improve the calculation accuracy of the parameters for the next interpolation point. Jiang et al. [17] presented an iterative feedback interpolator that consists of estimator and corrector stages to restrain the feedrate fluctuation. Chen et al. [18] developed a NURBS interpolation method for deriving parametric curves based on Steffensen's iterative method to avoid unnecessary feedrate fluctuations. Ji et al. [19] calculated the interpolation point parameters through iterative correction to obtain the accurate position and reduce the feedrate fluctuates. However, these methods have difficulty guaranteeing convergence when using iterative methods. To reduce feedrate fluctuations, Wang et al. [20] developed a bi-interpolation algorithm using two second-order Taylor expansions in one interpolation cycle. However, the time cost of this algorithm is relatively large.

In addition, much effort has been made to develop feedrate optimization techniques, and many fruitful feedrate

scheduling methods have been proposed. The feedrate profile of the linear acceleration and deceleration algorithm [21–23] is not smooth, and machine vibration will inevitably occur during the movement. In the early days, NURBS interpolation was realized by using the speed adaptive adjustment methods [1, 24], but these methods could not meet the tolerance limits of chord error, acceleration, and jerk. Du et al. [25] developed a complete S-shaped NURBS interpolation under confined jerk, acceleration limits, which adopted compensate on strategies to make the feedrate more continuous and the arc increment more precise. Lin et al. [26] used a look-ahead module to generate smooth feedrate profile to accommodate NURBS curve dynamics. Liu et al. [27] proposed the idea of NURBS curve segmentation to construct a new NURBS interpolator to improve contour error accuracy. Sun et al. [28] optimize short-line segmented machining using S-shaped ACC/DEC algorithm and time optimization function to obtain better machining accuracy and efficiency. Liu and Leng [14, 29] proposed a cubic polynomial NURBS interpolation. However, the cubic polynomial ACC/DEC algorithm is inefficient, and the jerk profile is discontinuous, which will impact the machine tool. Liu et al. [30] further improved and proposed the jerk-continuous NURBS interpolation method, which improved the contour machining accuracy and improved the low flexibility at small curvature sharp corners. However, these methods only consider the influence of the geometric characteristics of the curve and ignore the influence of the dynamic performance and contour error. Ni et al. [31] proposed a jerk-continuous NURBS interpolation. However, the computational load of this method is high, as the algorithm needs to deal with the coexistence of trigonometric functions and polynomial functions at the same time. Zhao et al. [32] applied a jerk-limited method to smooth the sharp corners of the final feedrate profile, which improved the feed correction polynomial to reduce the feed fluctuation. Furthermore, to address the problem of feedrate curve fluctuations in regions of large curvature, Su et al. [33] made full use of the local geometric information and adopted the method of sliding arc tube to optimize the target feedrate.

Traditional method is to subsegment the curve and then perform acceleration and deceleration processing based on local geometric information. In fact, when the arc length is less than the shortest requisite ACC/DEC distance, there are many types of ACC/DEC involved, and the processing is very complicated, which is not conducive to the realization of real-time interpolation. This article innovatively proposes a novel bidirectional scanning algorithm. For the curve subsegment with shorter arc length, directly process it at a uniform speed, discarding the complicated ACC/DEC process; for the long curve subsegment, readjust the acceleration end point or deceleration start point according to the acceleration or deceleration distance to meet the S-curve ACC/DEC requirements. This method is simple and efficient to process shorter curves, which is in line with the actual processing process of the special difficult-to-machine parts of the workpiece. Therefore, it can effectively improve the indicated processing quality of the processed part without increasing the hardware cost. Although the processing time

is extended to a certain extent, generally speaking, because the arc length of the curve subsegment is shorter, it is meaningful to sacrifice a certain amount of time to obtain higher real-time performance and processing effect.

2. The Presented Interpolation Architecture

2.1. NURBS Parameter Discretization. NURBS curves can be represented by parameters as follows:

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

where i is the index $i=0, 1, \dots, n$; $\{P_i\}$ is the set of control points; $\{\omega_i\}$ is the set of weights; and $\{N_{i,p}(u)\}$ is the set of p -order B-spline basic functions defined on the non-uniform knot vector \mathbf{U} , $U = \{u_0, u_1, \dots, u_{n+p+1}\}$. The B-spline basic function $N_{i,p}(u)$ is calculated as

$$\begin{cases} N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u), \\ N_{i,0}(u) = \begin{cases} 1, & u_i < u < u_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \end{cases} \quad (2)$$

The first-order and second-order derivatives of B-spline basic functions and NURBS curves are, respectively, given as

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

The first derivative and second derivative of $c(u)$ can be calculated as

$$\begin{cases} C'(u) = \frac{\sum_{i=0}^n N'_{i,p}(u)\omega_i p_i - c(u) \cdot \sum_{i=0}^n N'_{i,p}\omega_i}{\sum_{i=0}^n N_{i,p}\omega_i}, \\ C''(u) = \frac{\sum_{i=0}^n N''_{i,p}(u)\omega_i p_i - 2c'(u) \cdot \sum_{i=0}^n N'_{i,p}\omega_i - c(u) \cdot \sum_{i=0}^n N''_{i,p}\omega_i}{\sum_{i=0}^n N_{i,p}\omega_i}, \end{cases} \quad (4)$$

For a point $c(u)$ on the NURBS curve, the curvature radius $\rho(u)$ can be calculated by

$$\rho(u) = \frac{\|C'(u)\|^3}{\|C'(u) \times C''(u)\|}, \quad (5)$$

where $\|\cdot\|$ stands for the Euclidean norm and \times means the exterior product.

In this paper, the Adams-Bashforth-Moulton method (ABM) is adopted to discretize the parameter vector for NURBS interpolation. The ABM method is as follows:

$$\begin{cases} u_{m+1} = u_m + \frac{T}{24}(55f_k - 59f_{k-1} + 37f_{k-2} - 9f_{k-3}), \\ f_k = \left| \frac{du}{dt} \right| t = t_m = \frac{v(u_m)}{\|C'(u_m)\|} = \frac{v(u_m)}{\sqrt{(dC_x(u_m)/du_m)^2 + (dC_y(u_m)/du_m)^2 + (dC_z(u_m)/du_m)^2}}, \end{cases} \quad (6)$$

where $v(u_m)$, T , and $C'(u_m)$ are the velocity, the interpolation period, and the first derivatives of the NURBS curve; $C_x(u_m)$, $C_y(u_m)$, and $C_z(u_m)$ denote the parameter curve; u_m corresponds to each axis component of the interpolation point $C(u_m)$ on the curve.

The NURBS interpolation process is continuously affected by geometric errors, motion errors, drive errors, and contour errors. Aiming at the above problems, in the preprocessing stage, under the premise of considering multiple constraints, a real-time NURBS interpolation under multiple constraints (RNIC) is proposed, which can accurately obtain the constrained interpolation feedrate and optimal interpolation points.

2.2. Architecture of RNIC NURBS Interpolation. A systematic NURBS interpolator with smoothing feedrate at the FSA under multiple constraints is developed to enhance the interpolator performance with respect to the machining precision and machining quality. The overall structure of the interpolator is shown in Figure 1. The interpolator has two stages, namely, the preprocessing stage and the real-time interpolation stage. The former preprocessing stage aims to find the FSA quickly, while the latter real-time interpolation implements velocity smoothing and generation of interpolated toolpath points.

The interpolator comprehensively considers various constraints, breaking through the geometric constraints,

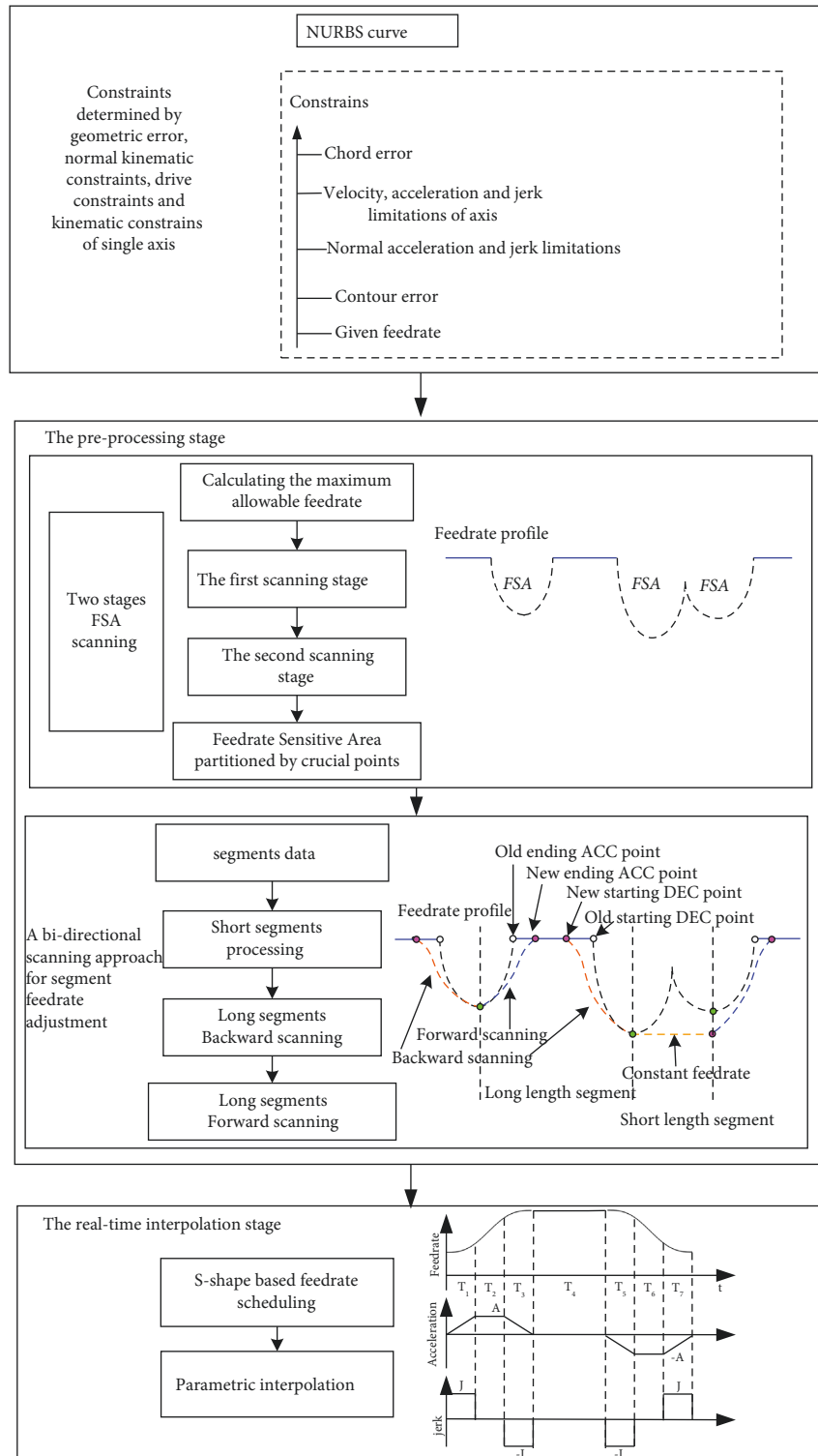


FIGURE 1: The structure of the FSA NURBS interpolator.

kinematic feature constraints, and machine performance constraints of NURBS interpolation, including various constraints such as chord error, velocity, normal acceleration and jerk, contour error, and command feedrate. There are two modules in the preprocessing stage, namely, the FSA scan module and the segment feedrate adjustment module,

which are used to obtain the geometric features required by the NURBS curve as the input for the next step of real-time interpolation.

The goal of the FSA scan module is to efficiently obtain FSAs of the NURBS curve by introducing the first FSA scan algorithm, then find the critical point in each FSA in the

second FSA scan algorithm and divide the FSA curve blocks into subsegments, and further get more detailed information about the geometric features of subsegments. The goal of the segment feedrate adjustment module is to optimize the feedrate profile for segments data generated by the former FSA scan module, which includes short segments processing, long segments backward scanning, and long segments forward scanning. During short segments processing, update the feedrate of the start/end point at the beginning and end of the segment to the same smaller speed, and then keep the entire short segment interpolation speed constant; during long segments processing, use backward and forward directions. In the forward scanning process, in order to simplify the processing flow, first set the start/end speed of the beginning and end to the smaller speed, and then obtain the actual maximum allowable feedrate of each segment; in the backward scanning process, the starting point of deceleration for a long segment is obtained.

In the real-time interpolation stage, it is mainly to obtain the feedrate and calculate the precise coordinates of the next interpolation point. The s-shape based feedrate scheduling is adopted to achieve smooth feedrate profile which can greatly improve the accuracy of interpolation.

3. Scheduling of the FSA under Multiple Constraints

This paragraph describes the scheduling method of FSA. First, determine the allowable feedrate on the NURBS curve under the constraints of geometric characteristics, dynamic characteristics, etc., and give its corresponding mathematical function formula. Then, the FSAs are detected quickly with a two-stage scanning method, which employs FSA scan and feedrate adjustment modules. The goal of the FSA scan module is to quickly find the FSA, including the maximum allowable feedrate calculation, the first scanning stage, the second scanning stage, and the FSA partition. The goal of the FSA feedrate adjustment is to optimize the start/end feedrate of the acceleration/deceleration phase, which includes intersection processing, backward scanning, and forward scanning.

3.1. Allowable Feedrate under Multiple Constraints

3.1.1. Allowable Feedrate for Chord Error Constraint.

Considering the chord $\overline{C(u_i), C(u_{i+1})}$ is a tiny segment in Figure 2, the arc approximation method can be used to calculate the chord error. In Figure 2, $P(u_i)$ is a point on the circle, $C(u_i)$ is a point on the curve, $P(u_i)$ and $C(u_i)$ are the coincident interpolation points, and the parameter vectors are both u_i . $P(u_{i+1})$, $C(u_{i+1})$ are the next estimated interpolation points, and their parameter vectors are both u_{i+1} .

Considering $L_i = \overline{C(u_i), C(u_{i+1})}$, we can get $\|P(u_{i+1}) - P(u_i)\| = L_i$, the approximated feedrate $v(u_i) = L_i/T$, and the formula for calculating chord error δ is

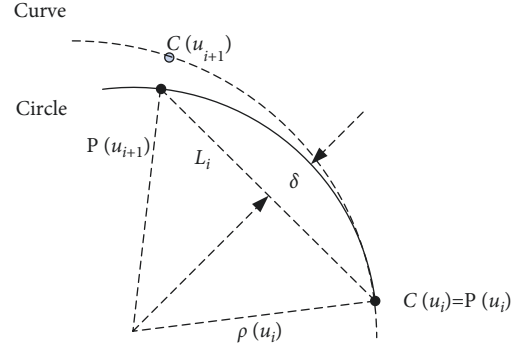


FIGURE 2: The chord error representation.

$$\delta = \rho(u_i) - \sqrt{\rho(u_i)^2 - \left(\frac{v(u_i)T}{2}\right)^2}. \quad (7)$$

$\rho(u_i)$ is the curvature radius of the interpolated point $C(u_i)$, and $v(u_i)$ is the velocity during the sampling time T . The allowable feedrate $v_{c1}(\rho(u_i))$ of the interpolated point $C(u_i)$ for the set maximum chord error δ_{\max} is

$$v_{c1}(\rho(u_i)) = \frac{2\sqrt{2\rho(u_i)\delta_{\max} - \delta_{\max}^2}}{T}. \quad (8)$$

3.1.2. Allowable Feedrate for the Normal Acceleration and Normal Jerk Constraints. Acceleration is the rate at which the speed and direction of an object are changing; it can be broken into its tangent and normal components. Here we first consider the normal acceleration; it can be expressed as

$$a_n = \frac{v^2}{\rho}. \quad (9)$$

Therefore, the allowable feedrate $v_{c2}(\rho(u_i))$ by the normal acceleration constraint is given as

$$v_{c2}(\rho(u_i)) = \sqrt{\rho(u_i)a_{n,\max}}, \quad (10)$$

where $a_{n,\max}$ is the maximum normal acceleration.

The normal jerk is calculated as

$$\left\{ \begin{array}{l} \Delta a_n = \|a_{n+} - a_{n-}\| = 2a_n \sin\left(\frac{\Delta\varphi}{2}\right) = 2\frac{v^2}{\rho} \sin\left(\frac{\Delta\varphi}{2}\right), \\ \Delta t = \frac{\rho\Delta\varphi}{v}, \\ j_n = \lim_{\Delta t \rightarrow 0} \frac{\Delta a_n}{\Delta t} = \lim_{\Delta\varphi \rightarrow 0} \frac{2v^2/\rho \sin(\Delta\varphi/2)}{\rho\Delta\varphi/v} = \frac{v^3}{\rho^2}. \end{array} \right. \quad (11)$$

Hence, the allowable feedrate $v_{c3}(\rho(u_i))$ for the normal jerk constrained can be derived as

$$v_{c3}(\rho(u_i)) = \sqrt[3]{\rho(u_i)^2 j_{n,\max}}. \quad (12)$$

where $j_{n,\max}$ is the maximum normal jerk.

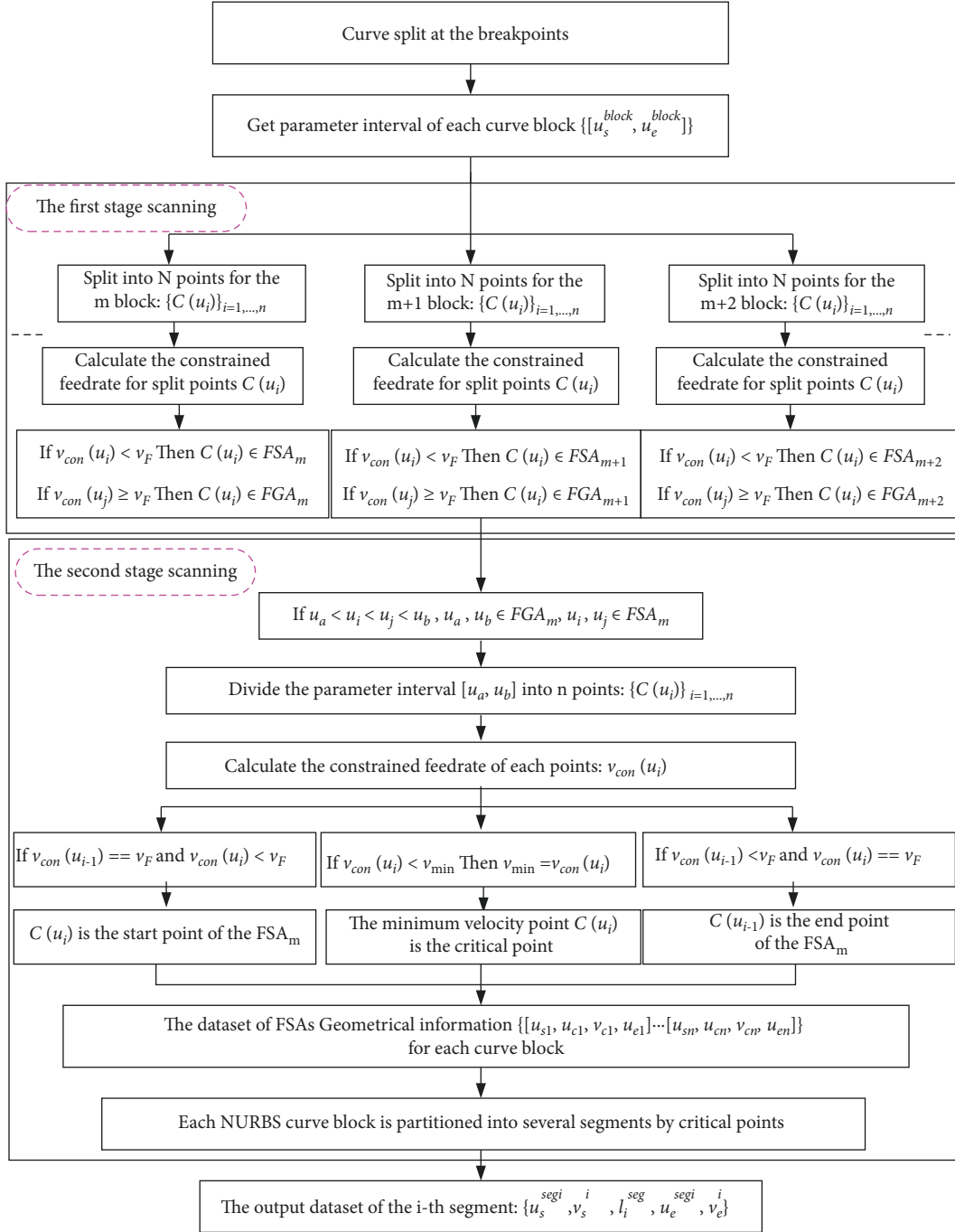


FIGURE 3: The flow chart of the two-stage FSA scan.

3.1.3. *Allowable Feedrate for the Contour Error Constraint.* The contour error directly affects the machining effect of the workpiece. For this reason, we include the contour error constraint in the preprocessor process to effectively limit the contour error from the source. Accurate estimation of contour errors has a significant impact on the quality of contour control. The contour error should satisfy the following formula:

$$\varepsilon < \varepsilon_{lim}, \quad (13)$$

where ε_{lim} is the given contour error.

The servo-lag induced contour error in real machining is considered as an additional constraint accompanied by the geometric and drive constraints during feedrate scheduling, so that the contour error can be bounded from the source. According to the literature [8], the feedrate constraint formula under the limitation of contour error is introduced as follows:

$$v_{c4}(\rho(u_i)) = \frac{\sqrt{2\rho(u_i)\varepsilon_{max}}}{T}, \quad (14)$$

Thus, the constrained interpolation feedrate $v_{\text{cons}}(u)$ can be obtained as

$$v_{\text{con}}(u_i) = \min(v_{c1}(\rho(u_i)), v_{c2}(\rho(u_i)), v_{c3}(\rho(u_i)), v_{c4}(\rho(u_i)), v_F). \quad (15)$$

$v_{\text{con}}(u_i)$ is the allowable feedrate under multiple constraints. It can be seen that the constrained feedrate is only related to the curvature of the point. However, due to the continuous influence of curve curvature fluctuation, NURBS interpolation will still be affected by the tolerance of tangential acceleration and jerk in the large curvature part. Then, a novel curve subsegmentation method was proposed during the preprocessing stage using a two-stage FSA scan method to obtain sufficient geometric characteristics of the curve. Compared with the existing methods, this method not only considers the optimal speed under various constraints, but also uses the preprocessing process to obtain enough curve geometric characteristic information without adding additional hardware.

3.2. The Preprocessing Stage

3.2.1. Curve Splitting at Breakpoints. During acceleration and deceleration on real-time interpolation of NURBS curves, the remaining curve distance needs to be calculated. The adaptive Simpson method can be used to calculate the length of the subsegment of the curve [8, 27]. But if there are breakpoints that appear to be discontinuous, C1 continuity is not satisfied which is a necessary condition for adaptive Simpson method. Therefore, the key is to find the breakpoints on the curve and segment the curve with the breakpoint as the boundary.

There are roughly two types of breakpoints [3]. The first type is the points of C0 continuity. The second type is the points with C1 continuity when there are multiple corresponding control points.

Then, the NURBS curve can be segmented into blocks according to these two kinds of breakpoints, and the buffer $\{[u_s^{\text{block}i}, u_e^{\text{block}i}]\}$ represented as the parameter interval of i -th block.

3.2.2. The Two-Stage FSA Scan. Although NURBS curves are roughly divided into blocks according to breakpoints, the interpolation of each block will continue to be affected by the change of curve curvature. In order to realize smoothness interpolation, ACC/DEC feedrate profile needs to be adopted in the interpolation. The more geometrical characteristic obtained, the better the acceleration and deceleration effect.

This paper creatively proposes a two-stage feedrate sensitive area (FSA) scanning method that can efficiently obtain sufficient geometrical characteristic without adding extra burden to real-time interpolation module. FSA represents a special area of the curve, which reflects the area where the allowable feedrate on the curve is less than the commanded feedrate.

TABLE 1: Forward scanning algorithm.

Forward scanning algorithm
Input: $\{u_s^{\text{segi}}, v_s^j, l_i^{\text{segi}}, u_e^{\text{segi}}, v_e^j\}$, a, J, N
Output: $\{u_s^{\text{segi}}, v_s^j, l_i^{\text{segi}}, V_{\text{max}}, u_e^{\text{segi}}, v_e^j\}$.
1: Initialize: Set $i = 0$
2: While $i < N$
3: Calculate $S_{\text{acc}}^{\text{req}}(v_s^j, v_F), S_{\text{dec}}^{\text{req}}(v_F, v_e^j), S_{\text{acc/dec}}^{\text{min}}(v_s^j, v_e^j)$.
4: If $l_i^{\text{segi}} \geq S_{\text{acc/dec}}^{\text{min}}(v_s^j, v_e^j)$.
5: Set $V_{\text{max}} = v_F$.
6: else
7: $v_s^j = v_e^j = \min(v_s^j, v_e^j)$.
8: Calculate $V_{\text{max}-i}^{\text{act}}$ to satisfy $S(v_s^j, V_{\text{max}-i}^{\text{act}}) = l_i^{\text{segi}}/2$.
9: Set $V_{\text{max}} = V_{\text{max}-i}^{\text{act}}$.
10: $i++$

TABLE 2: Backward scanning algorithm.

Backward scanning algorithm
Input: $\{u_s^{\text{segi}}, v_s^j, l_i^{\text{segi}}, V_{\text{max}}, u_e^{\text{segi}}, v_e^j\}$, N
Output: $\{u_s^{\text{segi}}, v_s^j, V_{\text{max}}, u_{\text{dec}}, u_e^{\text{segi}}, v_e^j\}$.
1: Initialize: Set $i = 0$
2: While $i < N$
3: If $V_{\text{max}} == v_F$.
4: Obtained u_{dec}^i according to reference 26
5: else if $V_{\text{max}} < v_F$.
6: $s_{\text{dec}} = l_i^{\text{segi}}/2$.
7: Obtained u_{dec}^i .
8: $i++$

The first stage of an FSA scanning method is to obtain the FSAs of the curve, and the second stage is to get more detailed information of FSAs, such as the parameter of the start/end point, critical point, and the corresponding constrained feedrate of the FSAs.

In the first stage scanning, the input data is the curve block parameter interval $[u_s^{\text{block}}, u_e^{\text{block}}]$ which is the output of Section 3.1. Firstly, the parameter interval is discretized into a set of elements $\{u_i\}$ by step size Δu . Then, the constrained feedrate $v_{\text{con}}(u_i)$ of each point $C(u_i)$ is calculated according to (15). If $v_{\text{con}}(u_i) < v_F$, then the point $C(u_i)$ belongs to the feedrate sensitive area (FSA); otherwise, it belongs to the feedrate insensitive area (FIA).

Therefore, the points of the curve block can be classified to FSA and FIA datasets through the first stage scanning. If the FSA set is not empty, it indicates that there are some feedrate sensitive areas in the curve block, and the constrained feedrates of the points in FSA are lower than the command feedrate. If the FIA set is not empty, it indicates that there are some feedrate insensitive areas in the curve block, and the constrained feedrates of the points in the FIA can reach the command value v_F .

The step size Δu is important in the first stage scanning; the larger the step size is, the fewer the points are. If the step size is very large, some FSAs will be lost, and if the step

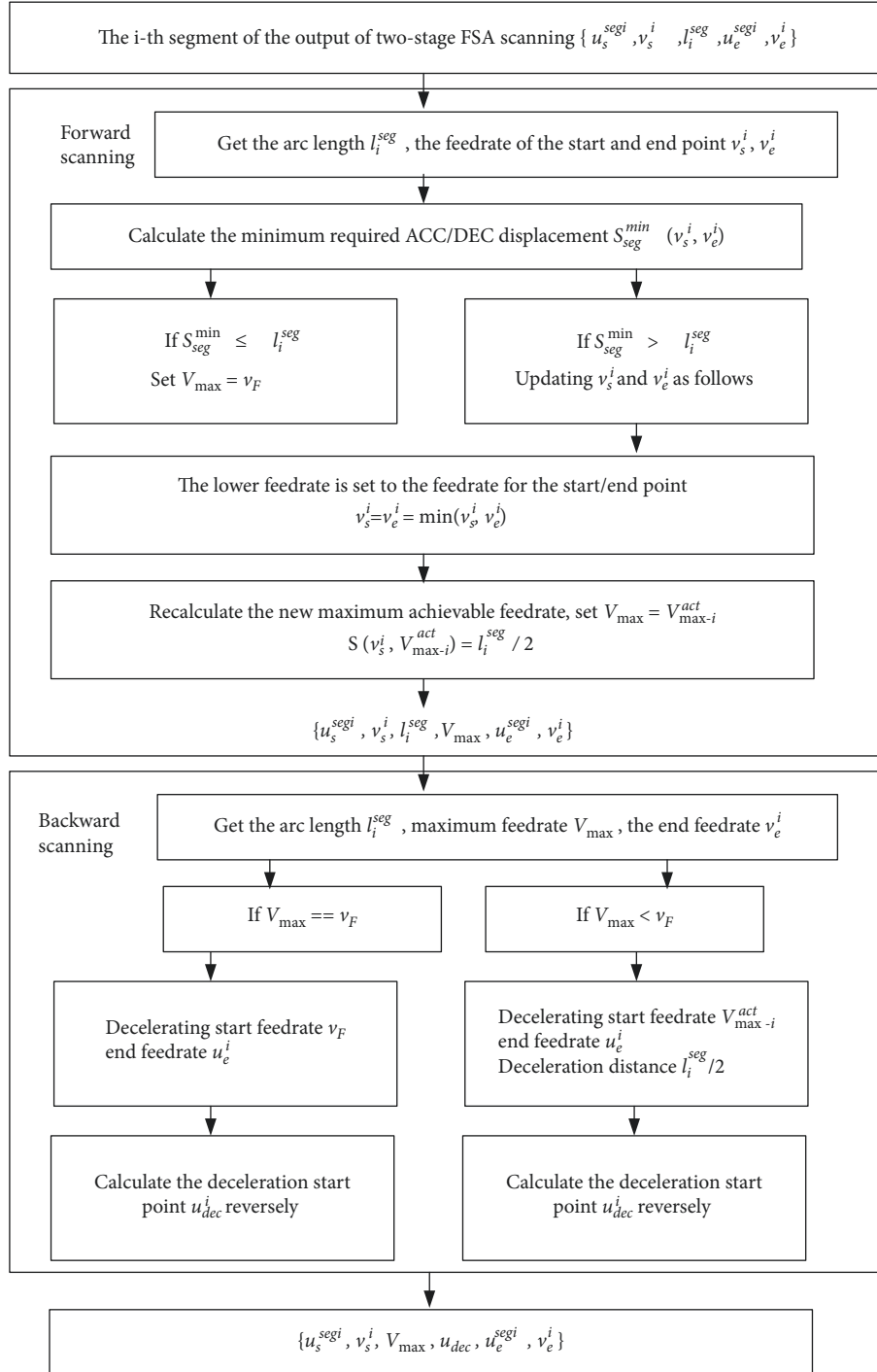


FIGURE 4: The bidirectional scanning method.

size is too small, the processing time will be increased. In actual processing, the step size shall be flexibly selected according to experience and curve characteristics.

After that, the second scanning stage is performed; the input data is the FSA and FIA datasets of the first stage scanning. In this stage, the curve blocks are further divided into subsegment, and it is necessary to obtain the geometrical characteristics information, such as the start/end point of the FSA.

Suppose there are four points $C(u_a), C(u_b), C(u_i), C(u_j)$ on a curve block, where the parameters $u_a, u_b \in FGA_m$, and $u_i, u_j \in FIA_m$, $C(u_a)$ and $C(u_b)$ are the adjacent points. If $u_a < u_i < u_j < u_b$, there must exist a feedrate sensitive area in the parameter interval $[u_a, u_b]$. So the second stage scanning is carried out for the detailed information of the FSA.

In the second stage scanning, the parameter interval $[u_a, u_b]$ is discretized into a set of elements $\{u_i\}_{i=1 \dots n}$ by step size Δu , and the corresponding curve is divided into n

points. The constrained velocity of each point $C(u_i)$ is calculated according to equation (6). If $v_{con}(u_{i-1}) == v_F$ and $v_{con}(u_i) < v_F$, it means that the velocity of the previous point $C(u_{i-1})$ is exactly equal to the command velocity v_F and the velocity of the current point $C(u_i)$ is lower than the command velocity v_F , and the current point $C(u_i)$ is considered to be the start point of the FSA; if $v_{con}(u_i) < v_F$ and $v_{con}(u_{i+1}) == v_F$, it means that the velocity of the next point $C(u_{i+1})$ is exactly equal to the command velocity v_F and the velocity of the current point $C(u_i)$ is lower than the command velocity v_F ; the current point $C(u_i)$ is considered to be the end point of the FSA; if the velocity of point $C(u_i)$ is the point with the lowest velocity of all points in the parameter interval $[u_a, u_b]$, then point $C(u_i)$ is considered to be the critical point of the FSA which has the lowest velocity and minimum radius of curvature.

So far, the detailed geometrical feature information of the FSA is obtained, such as the start/end point of the FSA. So the quaternion tuple $[u_s, u_c, v_c, u_e]$ corresponding to a FSA is got with the parameter u_s of the start point, the parameter u_c of the critical point and its constrained feedrate v_c , and the parameter u_e of the end point. To obtain accurate parameter of the critical point, a smaller step size Δu in the second stage scanning is required, which should be much smaller than the step size Δu in the first stage scanning.

The critical point has minimum radius of curvature and lowest feedrate in an FSA. Therefore, the critical point can be regarded as the start/end point of the acceleration/deceleration process in ACC/DEC feedrate profile. Between the two adjacent critical points, there is only one acceleration process and one deceleration according to the characteristics of NURBS curve. Hence, the curve block can be divided into several subsegments by critical points. This will greatly simplify feedrate planning for the feedrate profile of each subsegment has only one acceleration and one deceleration process.

After the seconde stage scanning, The NURBS curve block can be partitioned into subsegments by critical points with the datasets $\{u_s^{segi}, v_s^i, l_i^{seg}, u_e^{segi}, v_e^i\}$, where u_s^{segi} , u_e^{segi} are the parameters of the start point and end point, with the corresponding feedrate v_s and v_e , and l_i^{seg} is the arc length for the parameter interval $[u_s^{segi}, u_e^{segi}]$, which can be calculated by the adaptive simpson method [8, 27]. The flow chart of the two-stage scanning method is shown in Figure 3.

3.3. A Bidirectional Scanning Approach for the Subsegments.

The curve is divided into subsegments according to the critical point following the above method. If the arc length is not long enough, the feedrate cannot reach the command feedrate before deceleration. In addition, when the arc length is too short, even if the feedrate keeps accelerating or decelerating, it cannot reach the end velocity before reaching the endpoint of the subsegment. To deal with the short curve segments, there are seventeen types of feedrate profiles in [25], leading to the complex and time consume processing. In another paper, the authors proposed a new method that the scheduled feedrate keeps constant at most areas [8]; this

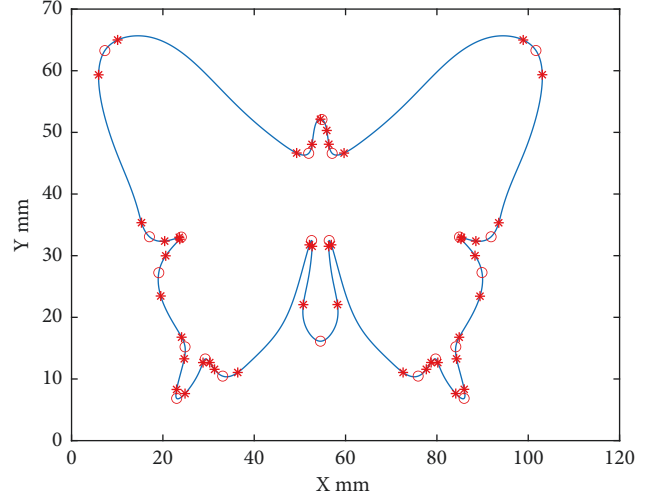


FIGURE 5: The butterfly-shaped curve.

TABLE 3: The parameters of the simulation.

Parameters	Symbol	Unit
Command feedrate	v_{max}	0.2 mm/ms
Normal acceleration	a_n	0.001 mm/ms ²
Normal jerk	j_n	0.0001 mm/ms ³
Tangential acceleration	A_t	0.001 mm/ms ²
Tangential jerk	J_t	0.0001 mm/ms ³
Max chord error	δ_{max}	0.001 mm
Max contour error	ϵ_{max}	0.05 mm

method greatly simplifies the processing process, but the machining efficiency is also greatly reduced.

For dataset $\{u_s^{segi}, v_s^i, l_i^{seg}, u_e^{segi}, v_e^i\}$ of the i -th subsegment, the requirement ACC/DEC displacements can be deduced as follows:

$$S_{acc/dec}^{req}(v_s^i, v_e^i) = S_{acc}^{req}(v_s^i, v_F) + S_{dec}^{req}(V_{max}, v_e^i) + nTv_F, \quad (16)$$

where $S_{acc/dec}^{req}(v_s^i, v_e^i)$ is the requirement ACC/DEC displacements, $S_{acc}^{req}(v_s^i, V_{max})$ is the acceleration displacements, $S_{dec}^{req}(V_{max}, v_e^i)$ is the deceleration displacements, and v_F is the command feedrate. There will exits a constant feedrate profile between acceleration and deceleration when $n > 0$ for. The minimum required ACC/DEC displacement is obtained if $n = 0$ as follows:

$$S_{acc/dec}^{min}(v_s^i, v_e^i) = S_{acc}^{req}(v_s^i, v_F) + S_{dec}^{req}(v_F, v_e^i). \quad (17)$$

And if

$$l_i^{seg} < S_{acc/dec}^{min}(v_s^i, v_e^i), \quad (18)$$

it means that the arc length l_i^{seg} is shorter than minimum required ACC/DEC displacement, and the feedrate profile of the segment cannot achieve the command feedrate v_F during the ACC/DEC procedure. If the arc length l_i^{seg} is longer than minimum required ACC/DEC displacement, the feedrate profile can accelerating from v_s^i to v_F and decelerating from v_F to v_e^i . In this case, the start point of the deceleration should be decided precisely for the real-time interpolation.

TABLE 5: Segments information of the butterfly curve.

Number	U_start	U_end	V_start	V_end	Required_arc length	Acture_arclength	Complete S curve profile
Segment1	0.001	0.0169	0.060443	0.071035	38.307739	6.10257	NO
Segment2	0.0169	0.151502	0.071035	0.136423	31.208744	50.677979	YES
Segment3	0.151502	0.237903	0.136423	0.122992	26.428003	33.198769	YES
Segment4	0.237903	0.256303	0.122992	0.009457	35.054008	7.142123	NO
Segment5	0.256303	0.2753	0.009457	0.142006	32.629776	7.737445	NO
Segment6	0.2753	0.306796	0.142006	0.104414	27.698153	13.756968	NO
Segment7	0.306796	0.336292	0.104414	0.036195	36.596878	8.730752	NO
Segment8	0.336292	0.35529	0.036195	0.03501	41.08812	9.29499	NO
Segment9	0.35529	0.368188	0.03501	0.113086	35.733379	4.824294	NO
Segment10	0.368188	0.449777	0.113086	0.014613	36.137478	31.123468	NO
Segment11	0.449777	0.499971	0.014613	0.091119	38.27055	18.485075	NO
Segment12	0.499971	0.550179	0.091119	0.014629	38.270397	18.501719	NO
Segment13	0.550179	0.631792	0.014629	0.113006	36.145996	31.115427	NO
Segment14	0.631792	0.644695	0.113006	0.035021	35.741718	4.826924	NO
Segment15	0.644695	0.663698	0.035021	0.036196	41.087765	9.303009	NO
Segment16	0.663698	0.693103	0.036196	0.104468	36.591499	8.716472	NO
Segment17	0.693103	0.724608	0.104468	0.141766	27.725693	13.74367	NO
Segment18	0.724608	0.743611	0.141766	0.009517	32.662388	7.756817	NO
Segment19	0.743611	0.762214	0.009517	0.125237	34.786385	7.203535	NO
Segment20	0.762214	0.848428	0.125237	0.135905	26.228525	33.126568	YES
Segment21	0.848428	0.983151	0.135905	0.071057	31.275175	50.711964	YES
Segment22	0.983151	1.0	0.071057	0	39.830734	6.435459	NO

Step 4. If $l_i^{seg} < S_{seg}^{\min}(v_s^i, v_e^i)$, go to Step 5; otherwise, set $V_{\max} = v_F$, go to Step 7.

Step 5. Update v_s^i and v_e^i by equation $v_s^i = v_e^i = \min(v_s^i, v_e^i)$. If the velocity at the starting/ending point of the previous subsegment is updated, the velocity at the starting/ending point of the adjacent subsegment should be updated to a lower velocity by an iterative method.

Step 6. Recalculate the new maximum achievable feedrate $V_{\max-i}^{act}$ to satisfy $S(v_s^i, V_{\max-i}^{act}) = l_i^{seg}/2$ that accelerated from v_s^i to $V_{\max-i}^{act}$ and the displacement is half of the subsegment arc length, set $V_{\max} = V_{\max-i}^{act}$.

Step 7. If $i < N$, N represents the number of subsegments, set $i = i + 1$; then go to Step 2; otherwise, end the program.

(B) Backward scanning:

Step 1. Set $i = N$; N represents the number of subsegments.

Step 2. Get the arc length l_i^{seg} , maximum feedrate V_{\max} , and the end feedrate v_e^i of the i -th subsegment from the forward scanning output.

Step 3. If $V_{\max} == v_F$ (command Feedrate), take the velocity of the ending point as the starting velocity, and v_F as ending velocity. The deceleration start point u_{dec}^i calculated reversely from u_e^i is the deceleration start point under S-shape ACC/DEC method described in reference 26; if $V_{\max} < v_F$, take the velocity of the ending point as starting velocity, and V_{\max} as ending velocity, $l_i^{seg}/2$ as the displacement. Calculate the deceleration start point u_{dec}^i from u_e^i reversely.

Step 4. If $i = 1$, program ends; otherwise, set $i = i - 1$, and go to Step 2.

After bidirectional scanning, the dataset of the final preinterpolation is obtained as $\{u_s^{segi}, v_s^i, V_{\max}, u_{dec}, u_e^{segi}, v_e^i\}$, where u_s^{segi} and u_e^{segi} are the parameters of the start/end point for the i -th subsegment, v_s^i and v_e^i are the corresponding feedrate, V_{\max} is the maximum achieved feedrate, and u_{dec} is the parameter of the start point of deceleration. The flow chart of the bidirectional scanning approach is shown in Figure 4.

4. Simulation and Results

The proposed NURBS interpolator is simulated on an Intel Core i7-8565U 1.8 GHz laptop. The butterfly-shaped curve is used for the proposed method shown in Figure 5. The proposed real-time NURBS interpolation under multiple constraints (RNIC) and adaptive feedrate interpolation (AFI) are adopted for comparison. The two methods are implemented for the butterfly-shaped curves, and the parameters are shown in Tables 3 and 4.

The curvature of the test curve is shown Figure 6. The test curve has high curvature and low curvature regions. In the low curvature area, due to the constraint of curvature, the velocity will not be able to feed at the command velocity, which can be a good verification of our proposed method.

Both preprocessing and online interpolation routines are developed in C. Additionally, we use MATLAB software to quantitatively analyze the actual velocity, tangential acceleration, and tangential jerk, which can be expressed as follows:

$$\left\{ \begin{array}{l} v_{\text{actual},i} = \frac{|C(u_{i+1}) - C(u_i)|}{T}, \\ a_{\text{actual},i} = \frac{v_{i+1} - v_i}{T}, \\ j_{\text{actual},i} = \frac{a_{i+1} - a_i}{T}. \end{array} \right. \quad (19)$$

Figure 5 shows FSAs detected by the proposed scheme (RNIC) indicated by asterisk and hollow circle, where the asterisk marked points are the starting/ending point of a FSA, and the hollow circle marked points are the minimum velocity point of the FSAs, called the critical points. The NURBS curve block can be partitioned into subsegments by critical points. All information of segments for the test curve is listed in Table 5. There are 21 segments on the test curve, where U_{start} and U_{end} are the parameters of starting/ending point of segment, and V_{start} and V_{end} are the velocity of starting/ending point of the segment. Required arc length is the minimum length for the complete S curve acc/dec profile from U_{start} to U_{end} at the test curve. Acture arclength is the length of the curve from the U_{start} to U_{end} at the test curve. The complete S curve profile field shows that the curve segment can complete a complete acceleration and deceleration process when the value is 1, or cannot complete when the value is 0.

The optimized velocity profile by two methods is shown in Figures 7 and 8, and the corresponding tangential acceleration/tangential jerk profiles are presented in Figures 9–12. As depicted in Figures 7 and 8, the feedrate is under the command feedrate by two methods. However, as shown in Figure 7, the starting speed of the feedrate profile in AFI method is rising directly from 0 to the command feedrate, which will cause a large feedrate vibration and affects the processing quality. A similar situation exists at the ending point in Figure 7. In the AFI method, the feedrate can meet the contour error limit adapted by the chord error, but the tangential acceleration exceeds the given tolerance at most of the time as shown in Figure 9. The maximum tangential acceleration is 0.0076 mm/s^2 by AFI method, as shown in Table 5, which is seven times more than the maximum allowable value. At the same time, the fluctuation of acceleration also causes the tangential jerk to exceed the maximum allowable value. The maximum tangential jerk is 0.0382 mm/s^3 by AFI, as shown in Table 6, which is 380 times more than the maximum allowable value.

Compared with the AFI method, As shown in Figure 8, the initial speed of the feedrate profile in the proposed RNIC method starts at zero and then accelerates to the command speed. Correspondingly, at the end of the feedrate profile, the speed decelerates to zero. In the feedrate scheduling process, the feedrate is optimized under multiple constraints. Therefore, from Figures 10 and 12, the tangential acceleration and tangential jerk by the FSA method can met system requirements; all tangential kinematic constraints of the proposed method are satisfied perfectly as shown in Table 6. Thus, the smoothness of the speed of the interpolation process is ensured.

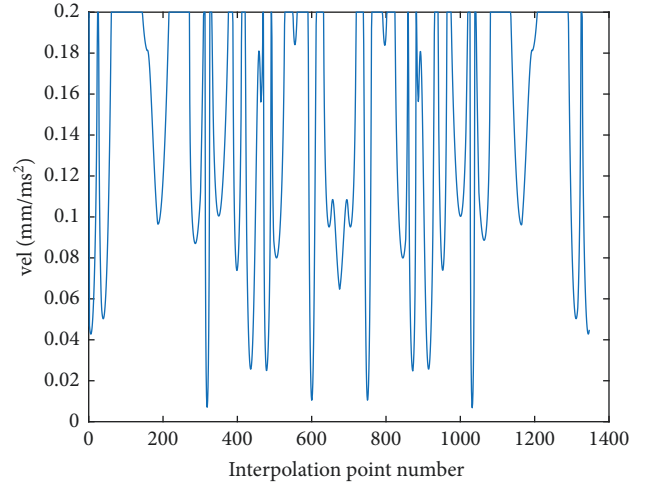


FIGURE 7: The velocity profile of the butterfly-shaped NURBS curve by AFI.

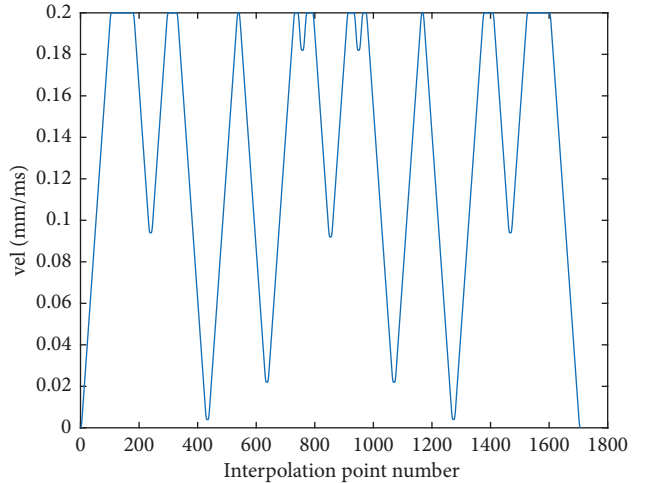


FIGURE 8: The velocity profile of the butterfly-shaped NURBS curve by RNIC.

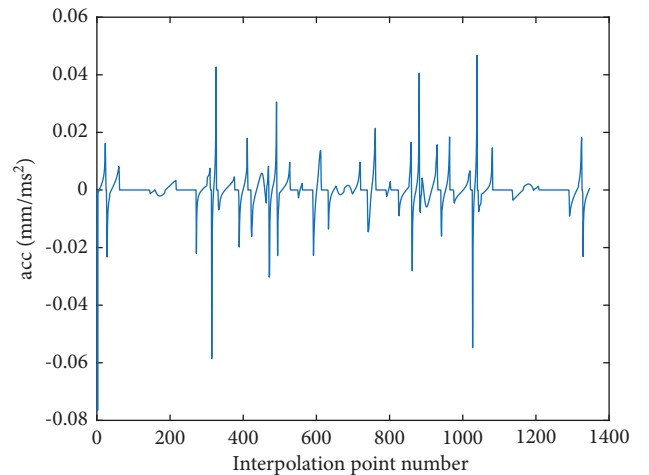


FIGURE 9: The ACC profile of butterfly-shaped NURBS curve by AFI.

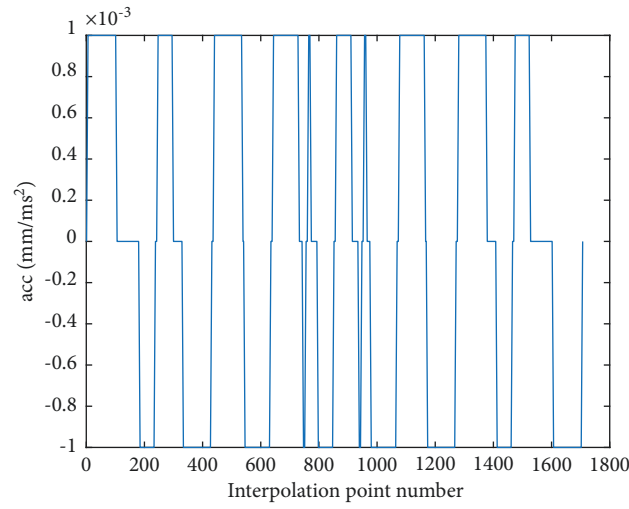


FIGURE 10: The ACC profile of butterfly-shaped NURBS curve by RNIC.

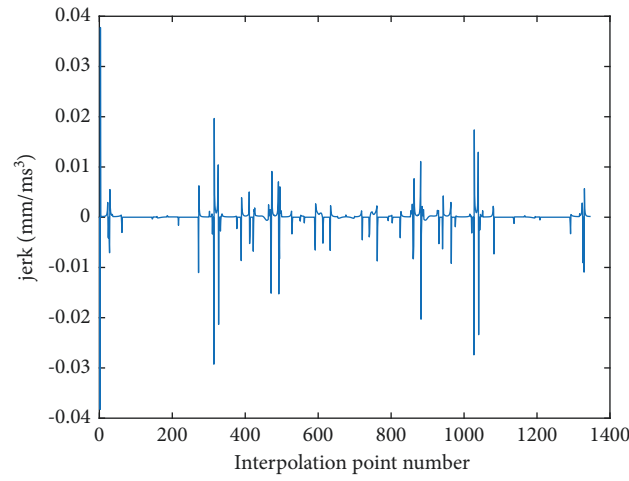


FIGURE 11: Jerk profile of butterfly-shaped NURBS curve by AFI.

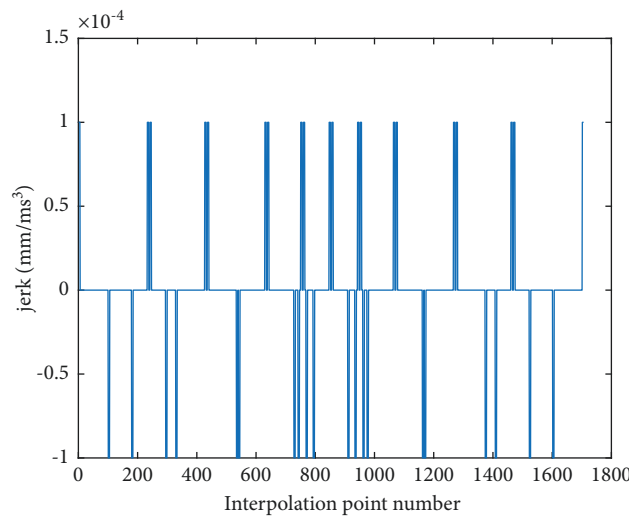


FIGURE 12: Jerk profile of butterfly-shaped NURBS curve by RNIC.

TABLE 6: The comparison of simulation results of the two methods.

Method	Start/end speed	Percentage of scheduling max tan-acc to maximum tan-acc (%)	Percentage of scheduling max tan-jerk to maximum tan-jerk (%)
AFI	2 mm/ms	760	38200
RNIC (proposed)	0 mm/ms	100	100

The proposed RNIC method comprehensively considers the constraint of chord error, normal acceleration, normal jerk, and contour error in the preprocessing stage and directly introduces S-shaped ACC/DEC to the curve segments in the real-time interpolation. Compared with the AFI method, the proposed RNIC method adds a preprocessing module, which can realize the optimal feed under various constraints. It can be seen from the simulation results that the former has a greater improvement in speed smoothness than the latter. From the perspective of real-time processing, the former arranges non-real-time tasks to be completed in the preprocessing stage which can greatly reduce the working load of the real-time interpolation module, so that high-speed and high-precision machining with a smaller interpolation period can be achieved. In this sense, the real-time performance is significantly improved.

5. Conclusion

In this paper, a real-time NURBS interpolation is proposed under geometric errors, normal kinematic constraints, and drive constraints. The preprocessing process is used to solve the problem that the velocity in NURBS interpolation is difficult to deal with under the condition of multiple constraints. The comparison and simulation analysis of the proposed RNIC method and the AFI method show that the proposed method has greatly improved in terms of velocity smoothness, acceleration, and jerk continuity. The simulation results show that the proposed algorithm is effective and greatly improves the real-time performance of the NURBS interpolator.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Hunan Province under Grant no. 2021JJ30574 and was partly supported by the Research Foundation of Education Bureau of Hunan Province under Grant no. 21B0424.

References

- [1] S. S. Yeh and P. L. Hsu, "Adaptive-feedrate interpolation for parametric curves with a confined chord error," *Computer-Aided Design*, vol. 34, no. 3, pp. 229–237, 2002.
- [2] T. Yong and R. Narayanaswami, "A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining," *Computer-Aided Design*, vol. 35, no. 13, pp. 1249–1259, 2003.
- [3] S.-H. Nam and M.-Y. Yang, "A study on a generalized parametric interpolator with real-time jerk-limited acceleration," *Computer-Aided Design*, vol. 36, no. 1, pp. 27–36, 2004.
- [4] X. Liu, F. Ahmad, K. Yamazaki, and M. Mori, "Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 4–5, pp. 433–444, 2005.
- [5] B. Sencer, Y. Altintas, and E. Croft, "Feed optimization for five-axis CNC machine tools with drive constraints," *International Journal of Machine Tools and Manufacture*, vol. 48, no. 7–8, pp. 733–745, 2008.
- [6] Y. Sun, Y. Zhao, J. Xu, and D. Guo, "The feedrate scheduling of parametric interpolator with geometry, process and drive constraints for multi-axis CNC machine tools," *International Journal of Machine Tools and Manufacture*, vol. 85, pp. 49–57, 2014.
- [7] X. Beudaert, S. Lavernhe, and C. Tournier, "Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path," *International Journal of Machine Tools and Manufacture*, vol. 57, pp. 73–82, 2012.
- [8] Z. Y. Jia, D. N. Song, J. W. Ma, G. Q. Hu, and W. W. Su, "A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints," *International Journal of Machine Tools and Manufacture*, vol. 116, pp. 1–17, 2016.
- [9] J. W. Ma, Z. Y. Jia, F. Z. Qin, D. N. Song, J. W. Wen, and S. Y. Chen, "A five-axis dual-NURBS interpolator with constant speed at feedrate-sensitive regions under axial drive constraints," *Journal of Manufacturing Science and Engineering*, vol. 141, no. 6, Article ID 061002, 2019.
- [10] H. Liu, Q. Liu, and S. Yuan, "Adaptive feedrate planning on parametric tool path with geometric and kinematic constraints for CNC machining," *International Journal of Advanced Manufacturing Technology*, vol. 90, no. 5–8, pp. 1–8, 2017.
- [11] L. Huan, L. Qiang, Q. Liu, S. Yuan, and P. Sun, "The optimal feedrate planning on five-axis parametric tool path with geometric and kinematic constraints for CNC machine tools," *International Journal of Production Research*, vol. 55, no. 13, pp. 1–17, 2017.
- [12] W. Zhong, X. Luo, W. Chang, F. Ding, and Y. Cai, "A real-time interpolator for parametric curves," *International Journal of Machine Tools and Manufacture*, vol. 125, pp. 133–145, 2018.
- [13] H. Liu, Q. Liu, P. Sun, Q. Liu, and S. Yuan, "A polynomial equation-based interpolation method of NURBS tool path with minimal feed fluctuation for high-quality machining," *International Journal of Advanced Manufacturing Technology*, vol. 90, no. 9–12, pp. 1–9, 2017.
- [14] L. Qiang, L. Huan, and S. Yuan, "High accurate interpolation of NURBS tool path for CNC machine tools," *Chinese Journal of Mechanical Engineering*, vol. 29, no. 5, pp. 1–10, 2016.

- [15] H. Ni, C. Zhang, Q. Chen, Y. Liu, S. Ji, and T. Hu, "A novel time-rounding-up-based feedrate scheduling method based on S-shaped ACC/DEC algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 104, no. 1, 2019.
- [16] 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," *International Journal of Industrial and Systems Engineering*, vol. 31, no. 2, p. 278, 2019.
- [17] J. Jiang, F. Lin, Y. Zhang, P. Ye, and Z. Hui, "A real-time feedrate planning method and efficient interpolator with minimal feedrate fluctuation for parametric toolpath," *IEEE Access*, vol. 7, no. 99, p. 1, 2019.
- [18] M. Chen and Y. Sun, "Contour error-bounded parametric interpolator with minimum feedrate fluctuation for five-axis CNC machine tools," *International Journal of Advanced Manufacturing Technology*, vol. 103, no. 4, pp. 1–18, 2019.
- [19] S. Ji, T. Hu, Z. Huang, and C. Zhang, "A NURBS curve interpolator with small feedrate fluctuation based on arc length prediction and correction," *International Journal of Advanced Manufacturing Technology*, vol. 111, no. 7-8, pp. 1–10, 2020.
- [20] 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," *International Journal of Precision Engineering and Manufacturing*, vol. 21, no. 2, 2020.
- [21] J. W. Jae and Y. Y. H. Young, "A generalized approach for the acceleration and deceleration of industrial robots and CNC machine tools," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 133–139, 2000.
- [22] J. Hu, L. Xiao, Y. Wang, and Z. Wu, "An optimal feedrate model and solution algorithm for a high-speed machine of small line blocks with look-ahead," *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 9-10, pp. 930–935, 2006.
- [23] Z. LiXian, S. RuiYong, H. Li, and X. Gao, "High speed interpolation for micro-line trajectory and adaptive real-time look-ahead scheme in CNC machining," *Science China Technological Sciences*, vol. 54, no. 6, pp. 1481–1495, 2011.
- [24] M.-Y. Cheng, M.-C. Tsai, and J.-C. Kuo, "Real-time NURBS command generators for CNC servo controllers," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 7, pp. 801–813, 2002.
- [25] X. Du, J. Huang, and L.-M. Zhu, "A complete S-shape feed rate scheduling approach for NURBS interpolator," *Journal of Computational Design and Engineering*, vol. 2, no. 4, pp. 206–217, 2015.
- [26] M.-T. Lin, M.-S. Tsai, and H.-T. Yau, "Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm," *International Journal of Machine Tools and Manufacture*, vol. 47, no. 15, pp. 2246–2262, 2007.
- [27] 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," *International Journal of Machine Tools and Manufacture*, vol. 87, pp. 1–15, 2014.
- [28] S. Sun, H. Lin, and L. Zheng, "A real-time and look-ahead interpolation methodology with dynamic B-spline transition scheme for CNC machining of short line subsegments," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 5-8, pp. 1359–1370, 2016.
- [29] H. B. Leng, Y. J. Wu, and X. H. Pan, "Research on cubic polynomial acceleration and deceleration control model for high speed NC machining," *Journal of Zhejiang University - Science*, vol. 9, no. 3, pp. 358–365, 2008.
- [30] L. Xinhua, P. Junquan, S. Lei, and Z. Wang, "A novel approach for NURBS interpolation through the integration of acc-jerk-continuous-based control method and look-ahead algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 88, no. 1-4, 2016.
- [31] 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, p. 1, 2018.
- [32] K. Zhao, S. Li, and Z. Kang, "Smooth minimum time trajectory planning with minimal feed fluctuation," *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9-12, pp. 1–13, 2019.
- [33] Z. Su, H. Zhou, P. Hu, and W. Fan, "Three-axis CNC machining feedrate scheduling based on the feedrate restricted interval identification with sliding arc tube," *International Journal of Advanced Manufacturing Technology*, vol. 99, no. 12, 2018.