IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON EMERGING TECHNOLOGIES FOR DEVICE TO DEVICE COMMUNICATIONS

Received September 3, 2018, accepted October 3, 2018, date of publication October 15, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875403

A Bidirectional Adaptive Feedrate Scheduling Method of NURBS Interpolation Based on S-Shaped ACC/DEC Algorithm

HEPENG NI[1,2,3], CHENGRUI ZHANG[1,2,3], SHUAI JI[4], TIANLIANG HU[1,2,3], QIZHI CHEN[1,2,3], YANAN LIU[5], AND GONGCHENG WANG[1,2,3]

[1]School of Mechanical Engineering, Shandong University, Jinan 250061, China
[2]Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Shandong University, Ministry of Education, Jinan 250061, China
[3]National Demonstration Center for Experimental Mechanical Engineering Education, Shandong University, Jinan 250061, China
[4]School of Mechanical and Electronic Engineering, Shandong Jianzhu University, Jinan 250101, China
[5]Bristol Robotics Laboratory, University of Bristol, Bristol BS8 1TH, U.K.

Corresponding author: Chengrui Zhang (crzhang@sdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51405270 and in part by the Special Foundation for National Integrated Standardization and New Model of Intelligent Manufacturing, China, under Grant Z135060009002-132.

ABSTRACT To improve the interpolation accuracy for the non-uniform rational B-spline (NURBS) curves, this paper proposes a novel bidirectional adaptive feedrate scheduling (BAFS) method based on S-shaped acceleration/deceleration (ACC/DEC) algorithm. The NURBS interpolator with the proposed BAFS method is also presented. The BAFS method has two meanings. First, to consider the continuous constraints of feedrate especially in the areas near the endpoints of each NURBS curve segment, an adaptive feedrate scheduling method is designed, which can improve the interpolation accuracy compared with the conventional method only considering the constraints of endpoints. Second, a bidirectional adaptive interpolation strategy including an adaptive task scheduling method and a meeting processing method is proposed to conduct the acceleration and deceleration interpolation stages of each NURBS segment separately in two directions. The two directions interpolation can be realized orderly by the adaptive task scheduling and the smoothness of feedrate profile can be guaranteed by the meeting processing method whose solvability is also proved. Therefore, the feedrate constraints at both the start and end points areas can be considered, which can improve the interpolation accuracy. Meanwhile, the round-off errors caused by cycle sampling are also considered and compensated in the bidirectional feedrate scheduling process to maintain the motion smoothness. In addition, an optimized look-ahead strategy is designed to ensure the correctness of the endpoints feedrate. Finally, a series of simulations and experiments are conducted to validate the good performance and applicability of the proposed method.

INDEX TERMS NURBS interpolation, bidirectional adaptive feedrate scheduling, S-shaped ACC/DEC algorithm, meeting processing method, round-off error compensation.

I. INTRODUCTION

Computer numerical control (CNC) machining plays an important role in modern manufacturing fields with growing demands for high-speed and high-accuracy machining [1]. Conventional CNC machine tools only with straight line and circular interpolation suffer from frequent acceleration and deceleration transformation and abrupt direction changes resulting in low accuracy and poor surface finish. In contrast, parametric curve interpolation has lots of advantages in surface quality, machining efficiency, memory consumption and motion smoothness [2]. Non-uniform rational B-splines (NURBS) has gotten the most attention as it offers a common mathematical form for the precise presentation of standard analytical shapes as well as free-form curves and surfaces [3]. And parametric interpolation for NURBS curves has become a crucial feature of high-end CNC machine tools.

Feedrate scheduling is one of the most important factors in NURBS interpolation and determines the smoothness, accuracy and stability of the machining process. However, there is no analytical relation between the arc length, curvature and

parameter of NURBS curves. It is more difficult to realize the feedrate scheduling of NURBS interpolation than conventional analytical curves such as straight line and circular arc. Therefore, many methods have been developed to schedule the feedrate. Some simple constant feedrate methods were proposed in [4]–[6]. However, constant feedrate profile without considering the curvature features of NURBS curves will cause large chord errors especially at the break points with only C0 continuity and the critical points with large curvature. Later, some improvements were made in [7]–[9] for the constant feedrate profile. But these methods did not consider the kinematic and dynamic constraints of machine tools such as the acceleration and jerk, which can cause vibration and shock. Therefore, three constraints need to be satisfied in feedrate scheduling: chord error, acceleration (including tangential and centripetal acceleration) and jerk.

Considering these constraints, some methods have been proposed with different acceleration/deceleration (ACC/DEC) algorithms. Annoni *et al.* [10] proposed a real-time configurable interpolation algorithm with limited acceleration, jerk and chord error. Heng and Erkorkmaz [11] proposed a numerically efficient feedrate modulation strategy based on the trapezoidal acceleration profile to guarantee that the final trajectory is jerk-limited in all axes. Dong and Stori [12] proposed a time-optimal feedrate scheduling algorithm based on the dynamics of machine tools and the capabilities of individual motion axis but the jerk constraint is not considered. Beudaert et al. [13] took full account of the kinematic characteristics of the machine tools to obtain time-optimal feedrate profiles. However, the chord error constraint might not be satisfied.

Although there are many kinds of ACC/DEC algorithms including the polynomial and trapezoidal jerk-limited methods [14]–[18] and the jerk-continuous methods [19]–[21], the S-shaped ACC/DEC algorithm is widely used for its simplicity, smoothness and high motion efficiency [22]–[25]. Lin *et al.* [22] proposed a dynamic-based interpolator with look-ahead algorithm (DBLA) considering the confined chord error and curvature of the curve detected by a geometric module. Liu *et al.* [23] developed a NURBS interpolator with feedrate scheduling (NIFS) which employs curve spitting and feedrate look-ahead for each segment. Du *et al.* [24] used a bidirectional scanning strategy to obtain the feedrates of critical points and also proposed a round-off error compensation strategy. However, the compensation strategy cannot maintain the continuous of the acceleration profile as well as the method given in [16], [26], and [27]. Ni *et al.* [25] proposed a composite round-off error compensation method for feedrate scheduling, which can guarantee the continuous of the acceleration profile. However, these methods only considered the constraints at the endpoints of each NURBS segment. But the feedrate in the middle part of each segment cannot be guaranteed to satisfy the multiple constraints because the constraints are continuous but the scheduled feedrate only compliances with standard ACC/DEC profile, which reduces the interpolation accuracy.

For the curve segment being machining, if the continuous constraints of the remaining part which have not been interpolated can be obtained, the interpolation accuracy can be improved. In order to obtain the constraint information of the remaining part, some similar adaptive feedrate scheduling strategies based on different deceleration profiles were proposed in [28]–[31]. They considered both the endpoints of each segment and their adjacent areas which have chord error constraint. But the centripetal acceleration constraint was still not included. Moreover, it is difficult to satisfy both the chord error and centripetal acceleration constraints through a standard ACC/DEC profile. Luo *et al.* [32] proposed a bidirectional optimization interpolation (BOI) method creatively on the condition of unknown arc length. However, the meeting processing based on the maximum feedrate proposed in [32] cannot generate continuous acceleration profile. In addition, the round-off error was not considered and compensated in these papers, which can affect the machining accuracy and motion smoothness.

In this paper, a novel bidirectional adaptive feedrate scheduling (BAFS) method based on S-shaped ACC/DEC algorithm is proposed to improve the machining accuracy. Correspondingly, the NURBS interpolator with the proposed BAFS method is also illustrated. The BAFS method has two meanings. Firstly, an adaptive feedrate scheduling method is designed to consider the continuous constraints of feedrate and improve the interpolation accuracy. Secondly, a bidirectional interpolation strategy including an adaptive task scheduling method and a meeting processing method is proposed to realize the acceleration and deceleration stages of each NURBS segment interpolation separately in two directions. Therefore, the feedrate constraints at both the start and end point areas are considered and the interpolation accuracy can be improved. Meanwhile, the round-off errors caused by cycle sampling are also considered and compensated in the bidirectional feedrate scheduling process while maintaining the continuity of acceleration. In addition, an optimized look-ahead strategy is designed to ensure the correctness of the endpoints feedrate.

The remainder of this paper is organized as follows. In section II, preliminaries including the definition of the NURBS curve, arc length calculation method and the S-shaped ACC/DEC algorithm are introduced. Section III illustrates the NURBS interpolator based on the proposed feedrate scheduling method. Simulation and experimental results are analyzed and compared with previous research works in section IV. The conclusions are given in section V.

II. PRELIMINARIES

A. DEFINITION OF A NURBS CURVE

A NURBS curve $C(u)$ can be defined as follows [3]:

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

with the control points $\{P_i\}$, weights $\{\omega_i\}$, the number of control points $(n+1)$, the degree p and the p-th degree B-spline basis functions $\{N_{i,p}(u)\}$ defined on the knot vector $U = \{u_0, u_1, \ldots, u_m\} = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b, \ldots, b}_{p+1}\}$. $(m+1)$ is the number of knots. Generally, we assume that $a = 0$, $b = 1$ and $\omega_i > 0$ for all i. There is a relation between p, $(n+1)$ and $(m+1)$ as follows:

$$m = n + p + 1 \tag{2}$$

The p th-degree B-spline basis functions $\{N_{i,p}(u)\}$ are defined as follows:

$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & otherwise \end{cases} \tag{3}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (i = 0, 1, \ldots, n) \tag{4}$$

B. ARC LENGTH CALCULATION OF A NURBS CURVE

The arc length of NURBS curve is necessary for feedrate look-ahead and real-time interpolation. In general, the arc length of a NURBS curve $C(u)$ over parameter interval $[a, b]$ can be expressed as follows:

$$l(a, b) = \int_a^b \left\| C'(u) \right\| du \tag{5}$$

where $||C'(u)||$ is the norm of $C'(u)$ and denoted by $g(u) = \left\| C'(u) \right\|$. However, it is difficult to calculate the arc length with this integral formula directly because there is no analytical expression for $g(u)$. Hence, some numerical integration methods are proposed to approximately estimate the arc length of a NURBS curve. Considering the computational efficiency and accuracy, the adaptive quadrature method [33] based on Simpson rule is employed in this paper. According to the theoretical conclusions derived in [34], the estimated values can be kept within a specific tolerance of its true value. By means of a composite Simpson rule, the initial interval $[u_l, u_u]$ is divided into two equal subintervals (i.e. $[u_l, \frac{u_u+u_l}{2}]$ and $[\frac{u_u+u_l}{2}, u_u]$). And the estimated arc length of each interval can be expressed as follows:

$$l'(u_l, u_u) = \frac{u_u - u_l}{6}[g(u_l) + 4g(\frac{u_l + u_u}{2}) + g(u_u)] \tag{6}$$

$$l'(u_l, \frac{u_u + u_l}{2}) = \frac{u_u - u_l}{12}[g(u_l) + 4g(\frac{3u_l + u_u}{4}) + g(\frac{u_l + u_u}{2})] \tag{7}$$

$$l'(\frac{u_u + u_l}{2}, u_u) = \frac{u_u - u_l}{12}[g(\frac{u_l + u_u}{2}) + 4g(\frac{u_l + 3u_u}{4}) + g(u_u)] \tag{8}$$

Assuming that the tolerance δ_{arc} is specified, if

$$\frac{1}{10}\left| l'(u_l, \frac{u_u+u_l}{2}) + l'(\frac{u_u+u_l}{2}, u_u) - l'(u_l, u_u) \right| < \delta_{arc} \tag{9}$$

then it can be prove that

$$\left| l'(u_l, \frac{u_u + u_l}{2}) + l'(\frac{u_u + u_l}{2}, u_u) - l(u_l, u_u) \right| < \delta_{arc} \tag{10}$$

Thus the arc length $l(u_l, u_u)$ over the parameter interval $[u_l, u_u]$ can be approximated by $l'(u_l, \frac{u_u+u_l}{2}) + l'(\frac{u_u+u_l}{2}, u_u)$ when Eq. (9) is satisfied. If Eq. (9) fails to be satisfied, the two subintervals $[u_l, \frac{u_u+u_l}{2}]$ and $[\frac{u_u+u_l}{2}, u_u]$ should be further divided for calculation and judgment until Eq. (9) holds in each subinterval of $[u_l, u_u]$. Finally, the whole arc length of a NURBS curve can be obtained by summing up the length of each subinterval as follows:

$$l(a, b) = \sum_{i=1}^{TotNum} l(u_{li}, u_{ui}) \tag{11}$$

where $TotNum$ denotes the total number of subintervals in $[a, b]$.

C. S-SHAPED ACC/DEC ALGORITHM

As shown in Fig. 1, the S-shaped ACC/DEC profile consists of seven phases. The phase time $\{[T_1, T_2, T_3, T_4, T_5, T_6, T_7]\}$ can be calculated according to the given motion parameters including the maximum jerk $J_{\max}$, acceleration $a_{\max}$, command feedrate F, displacement l, start and end feedrates v_s and v_e. Then, based on the phase time and the kinematic characteristics of S-shaped ACC/DEC algorithm, the start feedrate $\{[v_1, v_2, v_3, v_4, v_5, v_6, v_7]\}$ and the cumulative displacement $\{[S_1, S_2, S_3, S_4, S_5, S_6, S_7]\}$ of each phase can be obtained. Therefore, the feed length in each interpolation period can be calculated easily.

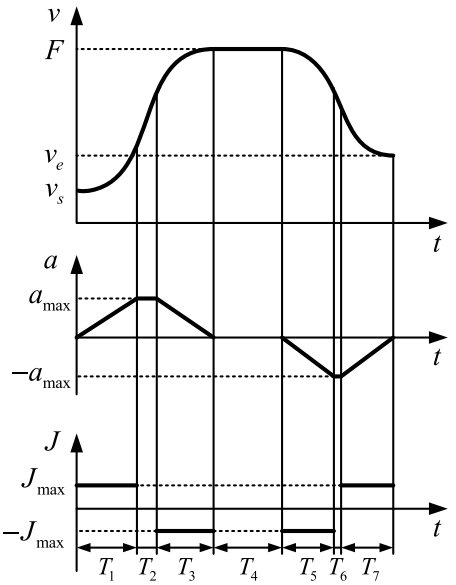


FIGURE 1. The S-shaped ACC/DEC profile with seven sections.

III. NURBS INTERPOLATOR WITH THE BIDIRECTIONAL ADAPTIVE FEEDRATE SCHEDULING METHOD

In this section, the NURBS interpolation procedures with the proposed feedrate scheduling method is described in detail. The overall structure of NURBS interpolator is shown in Fig. 2, which consists of the pre-processing stage and real-time interpolation stage.

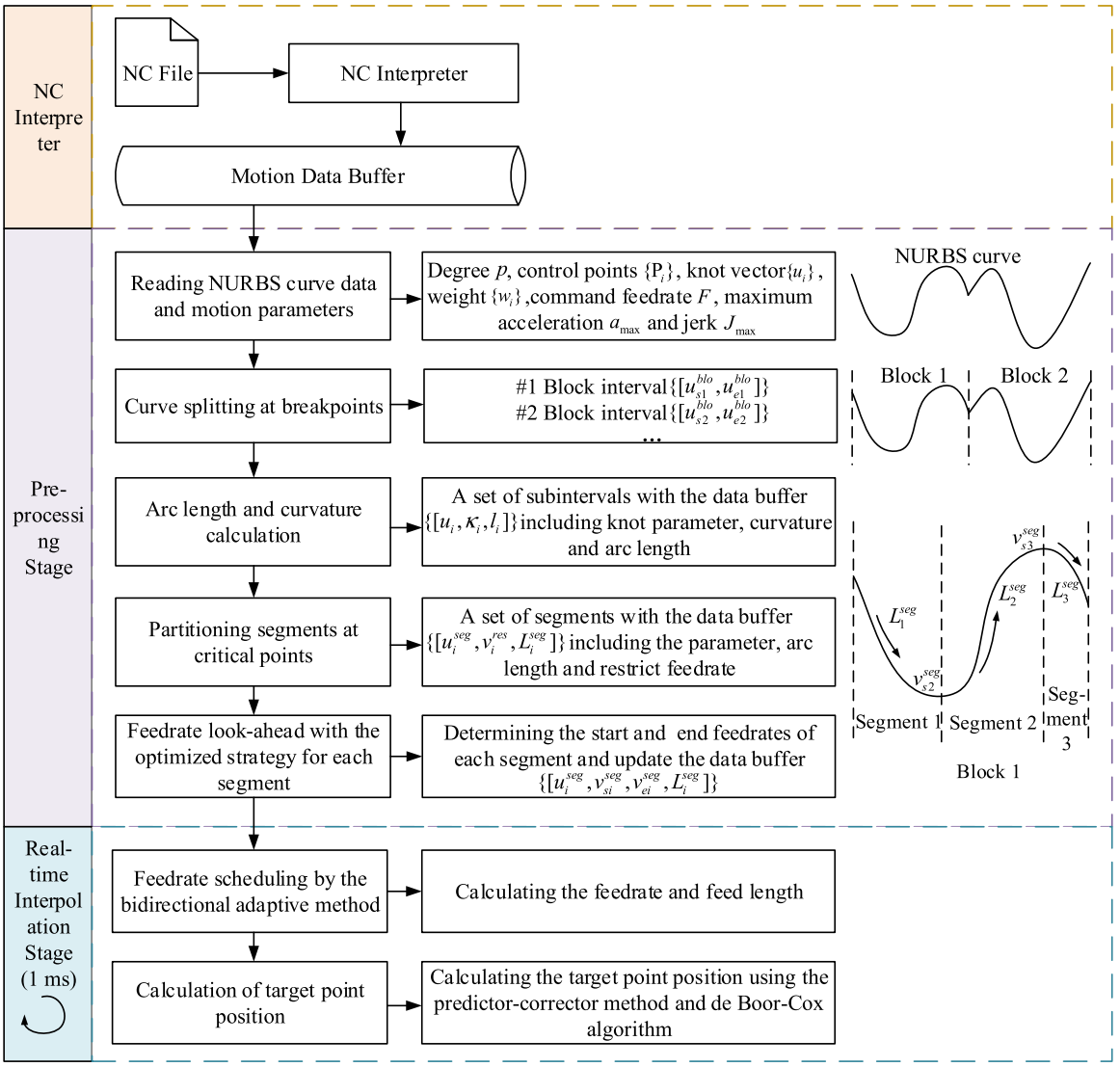


FIGURE 2. The overall structure of NURBS interpolator with BAFS method.

A. PRE-PROCESSING STAGE

In the stage of pre-processing, four modules which are curve splitting at breakpoints, arc length and curvature calculation, partitioning segments at critical points and feedrate look-ahead with the optimized strategy will be performed to obtain the necessary features of NURBS curves and prepare for real-time interpolation.

1) CURVE SPLITTING AT BREAKPOINTS

Breakpoints are the points at where it is visually discontinuous on the curve. The feedrate of the breakpoint is always set to zero. In addition, C1 continuity is the essential condition of calculating curve arc length precisely by the adaptive quadrature method and composite Simpson rule given in subsection II.B. Therefore, it is necessary to find the breakpoints and split the whole curve into several blocks.

There are two situations for detecting the breakpoints on the curve [23]. Firstly, the points with only C0 continuity are the breakpoints. Secondly, it is possible for the points with C1 continuity to be classified as the breakpoints when the corresponding control points are multiple [3].

With these two kinds of breakpoint detection, the whole curve can be divided into several blocks and the each block buffer $\{[u_{si}^{blo}, u_{ei}^{blo}]\}$ can be obtained and stored where u_{si}^{blo} and u_{ei}^{blo} are the start and end knots of i-th block, respectively.

2) ARC LENGTH AND CURVATURE CALCULATION

With the adaptive quadrature method given in subsection II.B, the arc length can be obtained approximately with a series of parameter subinterval $[u_{li}, u_{ui}]$. In addition, the curvature κ is needed for segment partitioning of each block and feedrate scheduling. Therefore, during the calculation of arc length, the curvature of each subinterval boundary knot should be

obtained. The curvature $\kappa(u_j)$ at the point with parameter u_j of a NURBS curve $C(u)$ can be expressed as follows [20]:

$$\kappa(u_j) = \frac{\left\| C'(u_j) \times C''(u_j) \right\|}{\left\| C'(u_j) \right\|^3} \tag{12}$$

Through the arc length and curvature calculation module, a series of feature data buffer $\{[u_i, \kappa_i, l_i]\}$ of each block can be generated where u_i is the end point parameter of i-th subinterval, κ_i is the curvature and l_i is the cumulative arc length.

3) PARTITIONING SEGMENTS AT CRITICAL POINTS

Sharp corners with large curvatures in a NURBS curve block have chord error and centripetal acceleration constraints on feedrate. Therefore, each block should be further partitioned to several segments by the sharp corners which are called critical points. And each segment is the basic unit of feedrate scheduling with one acceleration and deceleration process. To find the critical points, three steps should be applied. Firstly, the candidate points which are the boundary knots with larger curvature than the two neighbor boundary knots are found through scanning the feature data buffer $\{[u_i, \kappa_i, l_i]\}$ of each block. Secondly, the restricted feedrate of each candidate point under chord error and centripetal acceleration constraints can be calculated. As for chord error, the feedrate should satisfy the inequality as follows [22]:

$$v_i^{res} \le \frac{2}{T_s}\sqrt{\frac{1}{\kappa_i^2} - (\frac{1}{\kappa_i} - \delta_{cho})^2} \tag{13}$$

where v_i^{res} is the restricted feedrate of i-th candidate point, κ_i is the corresponding curvature, δ_{cho} is the maximum allowable chord error and T_s is the interpolation period. As for centripetal acceleration, the feedrate should satisfy the inequality [23]:

$$v_i^{res} \le \sqrt{\frac{a_{\max c}}{\kappa_i}} \tag{14}$$

where $a_{\max c}$ is the maximum allowable centripetal acceleration. Therefore, v_i^{res} can be expressed as follows:

$$v_i^{res} = \min(\frac{2}{T_s}\sqrt{\frac{1}{\kappa_i^2} - (\frac{1}{\kappa_i} - \delta_{cho})^2}, \sqrt{\frac{a_{\max c}}{\kappa_i}}) \tag{15}$$

Finally, the candidate point whose restricted feedrate is smaller than the command feedrate F is the critical point. Hence, each NURBS curve block is partitioned into several segments. Meanwhile, for i-th segment, the data buffer $\{[u_i^{seg}, v_i^{res}, L_i^{seg}]\}$ can be obtained with the end point parameter u_i^{seg}, the corresponding restricted feedrate v_i^{res} and the arc length of the segment L_i^{seg}.

4) FEEDRATE LOOK-AHEAD FOR EACH SEGMENT WITH THE OPTIMIZED LOOK-AHEAD STRATEGY

In order to conduct the feedrate scheduling, the start and end feedrates of each segment should be calculated at first through look-ahead. The bidirectional feedrate look-ahead method [23] which consists of back-trace module and forward module is employed in this paper. Firstly, the end feedrate of each block is assumed to be 0 mm/s. And the back-trace module is performed based on S-shaped ACC/DEC algorithm to calculate the start feedrate of each segment from the last segment to the start segment. Secondly, the forward module is applied to recalculate the end feedrate of each segment from the start segment to the last segment. Through these two modules, it can be sure that the start feedrate can speed up or slow down to the end feedrate in each segment according to standard S-shaped ACC/DEC algorithm. Furthermore, the segment feature data buffer should be updated by $\{[u_i^{seg}, v_{si}^{seg}, v_{ei}^{seg}, L_i^{seg}]\}$ where v_{si}^{seg} and v_{ei}^{seg} are the start and end feedrates of i-th segment, respectively.

However, with the bidirectional look-ahead method and standard S-shaped ACC/DEC algorithm, only the feedrate constraints at the endpoints of each segment can be considered. Nevertheless, the multiple constraints are existing continuously especially in the areas near the endpoints. In some situations, the look-ahead results might affect the motion smoothness. For example, if $v_{si}^{seg} > v_{ei}^{seg}$, v_{ei}^{seg} might not be able to accelerate to v_{si}^{seg} in reverse direction under the continuous constraints, which means that the feedrate profile is not smooth enough in forward interpolation. To cope with this problem, an optimized look-ahead strategy is proposed and its flowchart is shown in Fig. 3.

A back-trace correction module is introduced to correct the start feedrates of each segment in the same direction of back-trace module. For the k-th segment where $v_{sk}^{seg} > v_{ek}^{seg}$, the backward interpolation is conducted step by step based on the adaptive feedrate scheduling method which is introduced in subsection III.B.2 until the maximum backward feedrate v_{dk}^{back} with zero acceleration is bigger or equal to v_{sk}^{seg}. At the same time, the remaining arc length l_k^r between current interpolation point and start point of current segment is updated in every interpolation step. If $l_k^r \ge 0$ and $v_{dk}^{back} \ge v_{sk}^{seg}$, v_{sk}^{seg} can be accepted. Otherwise, v_{dk}^{back} needs to be adjusted downwards to guarantee that $l_k^r = 0$ and v_{sk}^{seg} should be updated by v_{dk}^{back}. Therefore, the data buffer $\{[u_i^{seg}, v_{si}^{seg}, v_{ei}^{seg}, L_i^{seg}]\}$ can be corrected after the backward correction module. The backward interpolation results should also be stored to the data buffer $\{[u_j, C(u_j)]\}$. Meanwhile, it is of great significance to apply the back-trace correction module to ensure the real-time performance of the real-time interpolation stage given in subsection III.B.

B. REAL-TIME INTERPOLATION STAGE WITH THE BIDIRECTIONAL ADAPTIVE FEEDRATE SCHEDULING METHOD

In real-time interpolation stage, the main tasks are to obtain the feed length and calculate the corresponding target interpolation point in each period. A novel bidirectional adaptive feedrate scheduling method is proposed and implemented in real-time to improve the interpolation accuracy. In addition, the round-off errors caused by cycle sampling are also

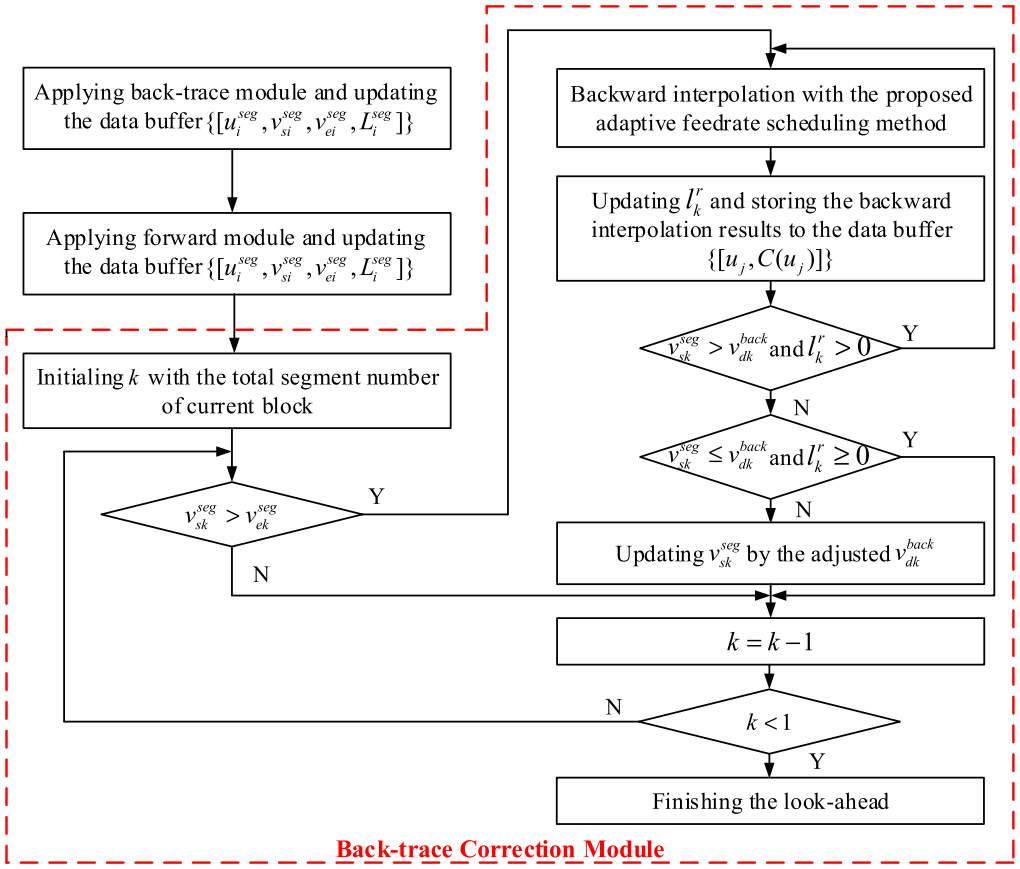


FIGURE 3. The optimized feedrate look-ahead strategy.

considered and compensated to maintain the smoothness of the feedrate profiles.

1) THE FUNDAMENTAL THEORY OF BIDIRECTIONAL ADAPTIVE FEEDRATE SCHEDULING

Bidirectional adaptive feedrate scheduling has two meanings. Firstly, the feedrate is scheduled adaptively in real-time under the various constraints, which is introduced in subsection III.B.2. Therefore, the continuous constraints of feedrate can always be considered especially in the areas near the endpoints of each NURBS segment, which improves the interpolation accuracy. Secondly, the feedrate scheduling and interpolation for the acceleration and deceleration stages of each NURBS segment is conducted separately in two directions, which is presented in subsection III.B.3. The acceleration stage is realized by forward interpolation and the deceleration stage is performed by backward acceleration interpolation. At the meeting point of two directions interpolation, the forward and backward feedrates can be adjusted to be equal to each other with zero acceleration to generate smooth feedrate profile. Therefore, compared with the unidirectional feedrate scheduling, the various constraints in the terminal areas of each NURBS segment can be considered and the interpolation accuracy can be improved.

To reduce the computational load, a special output strategy of the two directions interpolation results is adopted. Before the meeting of two directions interpolation, only the forward interpolation results are output and the results of backward interpolation are stored. After the meeting, the stored backward interpolation data is taken out in reverse order as forward output until the interpolation of current segment is completed.

2) ADAPTIVE FEEDRATE SCHEDULING METHOD BASED ON S-SHAPED ACC/DEC ALGORITHM

Since both forward and backward interpolation are acceleration processes, only forward feedrate scheduling is discussed in this subsection. During machining, the kinematic parameters of each interpolation point should satisfy various constraints. Firstly, the inequalities should be satisfied as follows:

$$v_i \le F \tag{16}$$

$$a_i \le a_{\max t} \tag{17}$$

$$J_i \le J_{\max} \tag{18}$$

where v_i, a_i and J_i are the target feedrate, tangential acceleration and jerk of i-th interpolation period respectively, F is the command feedrate, $a_{\max t}$ is the allowable maximum

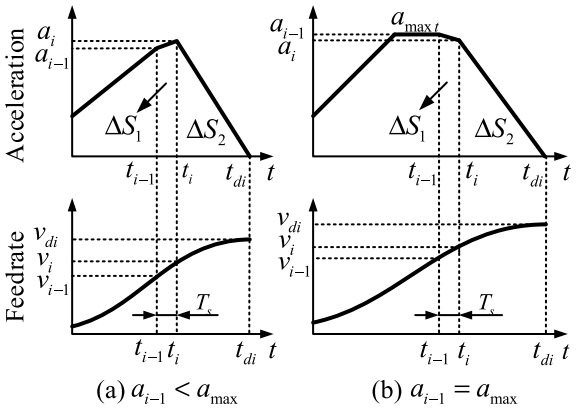


FIGURE 4. Feedrate scheduling based on S-shaped ACC/DEC algorithm.

tangential acceleration, $J_{\max}$ is the allowable maximum jerk. Secondly, v_i should satisfy the chord error and centripetal acceleration constraints shown in Eq. (15). Thirdly, as shown in Fig. 4, v_i, a_i and J_i can be derived based on S-shaped ACC/DEC algorithm as follows:

$$v_i = v_{i-1} + \Delta S_1 = v_{i-1} + \frac{a_i + a_{i-1}}{2} T_s \tag{19}$$

$$a_i = a_{i-1} + J_i T_s \tag{20}$$

$$J_i = \begin{cases} J_{\max}, & phase\ 1 \\ 0, & phase\ 2\ or\ phase 4 \\ -J_{\max}, & phase\ 3 \end{cases} \tag{21}$$

where phase 1, 2, 3 and 4 indicate the increasing acceleration phase, constant acceleration phase, decreasing acceleration phase and constant feedrate phase shown in Fig. 1, respectively. Before $a_{i-1} = a_{\max t}$, the interpolation should be in phase 1 and $J_i = J_{\max}$. When $a_{i-1} = a_{\max t}$, the interpolation should enter phase 2 and $J_i = 0$. The conditions of entering phase 3 and 4 will be analyzed later.

Based on Eq. (15)-(21), the target acceleration a_i and feedrate v_i can be expressed as follows:

$$a_i = \min[a_{\max t}, \max(a_{i-1} + J_i T_s, 0)] \tag{22}$$

$$v_i = \min(F, \sqrt{\frac{a_{\max c}}{\kappa_i}}, \frac{2}{T_s}\sqrt{\frac{1}{\kappa_i^2} - (\frac{1}{\kappa_i} - \delta_{cho})^2}, v_{i-1} + \frac{a_i + a_{i-1}}{2} T_s) \tag{23}$$

Considering the computational load, since the feed length in one interpolation period is small and the curvature profile of a NURBS curve is continuous, the curvature of target point κ_i can be replaced by the current point value κ_{i-1} rather than through a complex iterative process to calculate. Furthermore, the target acceleration a_i should be updated because the final feedrate v_i might not be obtained by Eq. (19). Therefore, a_i can be updated as follows based on Eq. (19):

$$a_i = 2\frac{v_i - v_{i-1}}{T_s} - a_{i-1} \tag{24}$$

The corresponding target feed length of i-th period Δl_i can be derived based on Eq. (20)-(21) and (23)-(24) as follows:

$$\Delta l_i = v_{i-1} T_s + \frac{a_{i-1}}{2} T_s^2 + \frac{J_i}{6} T_s^3 \tag{25}$$

where J_i is updated by $J_i = \frac{a_i - a_{i-1}}{T_s}$.

Finally, if the interpolation is in phase 1 or 2, it should be guaranteed that the kinematic parameters will be within allowable range when a_i is reduced to zero. As shown in Fig. 4, the corresponding decreasing acceleration time t_{di} and the maximum feedrate v_{di} can be expressed as follows:

$$t_{di} = \frac{a_i}{J_{\max}} \tag{26}$$

$$v_{di} = v_i + \Delta S_2 = v_i + \frac{a_i^2}{2J_{\max}} \tag{27}$$

The corresponding length from v_i to v_{di} can be derived as follows:

$$\Delta l_{di} = v_i t_{di} + \frac{a_i}{2} t_{di}^2 - \frac{J_{\max}}{6} t_{di}^3 \tag{28}$$

Since Δl_{di} might be much bigger than Δl_i, the corresponding κ_{di} cannot be replaced by κ_{i-1}. However, accurate calculation of κ_{di} is time-consuming but not necessary. Therefore, the linear interpolation method based on the feature data buffer $\{[u_i, \kappa_i, l_i]\}$ given in subsection III.A.2 is utilized to estimate κ_{di} as follows:

$$\kappa_{di} = \kappa_{m+1}\frac{l_{i-1}^{cum} + \Delta l_i + \Delta l_{di} - l_m}{l_{m+1} - l_m} + \kappa_m \frac{l_{m+1} - \Delta l_i - \Delta l_{di} - l_{i-1}^{cum}}{l_{m+1} - l_m} \tag{29}$$

where $l_{i-1}^{cum} = \sum_{j=1}^{i-1} \Delta l_j$ and $l_m \le l_{i-1}^{cum} + \Delta l_i + \Delta l_{di} \le l_{m+1}$. If $v_{di} \le \min(F, \sqrt{\frac{a_{\max c}}{\kappa_{di}}}, \frac{2}{T_s}\sqrt{\frac{1}{\kappa_{di}^2} - (\frac{1}{\kappa_{di}} - \delta_{cho})^2})$, the target feedrate v_i can be accepted. Otherwise, v_i and a_i need to be adjusted downwards. In particular, if $v_{di} > F$, the feedrate scheduling should enter phase 3 in next period. In order to adjust v_i and a_i, an equation can be constructed as follows:

$$h(a''_i) = v_{di}(a''_i) - \min(F, \sqrt{\frac{a_{\max c}}{\kappa_{di}}}, \frac{2}{T_s}\sqrt{\frac{1}{\kappa_{di}^2} - (\frac{1}{\kappa_{di}} - \delta_{cho})^2}) = 0 \tag{30}$$

where a''_i is the adjusted target acceleration. From the calculation results of last period, v_{di-1} is in allowable range with the target acceleration a'_i where $a'_i = a_{i-1} - J_{\max} T_s$. Meanwhile, because of the monotonically increasing relation between a_i and v_{di} and no more than one stationary point of the curvature curve in one segment, there must exist a unique solution a''_i in $[a'_i, a_i)$ for Eq. (30). Since Eq. (30) is a high order equation, the dichotomy method can be utilized to solve a''_i and the flowchart is shown in Fig. 5 where δ_{dic} denotes the allowable error.

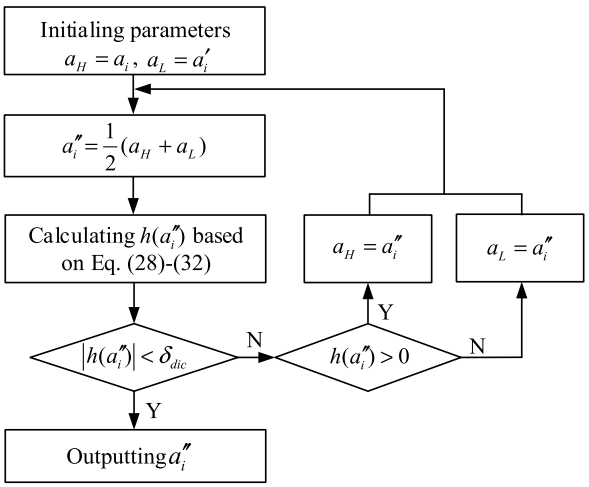


FIGURE 5. The flowchart of solution based on dichotomy method.

In general, the feedrates in phase 3 scheduled from v_i to v_{di} with negative jerk can always satisfy the multiple constraints. Therefore, to simplify the calculation, the feedrates between v_i and v_{di} are not judged whether the various constraints can be satisfied. During the interpolation in phase 3, if $a_i = 0$, the feedrate scheduling should enter phase 4 in next period.

3) BIDIRECTIONAL ADAPTIVE INTERPOLATION STRATE-GY

There are three main tasks to realize the two directions interpolation orderly with high efficiency and smooth feedrate profile. Firstly, an adaptive task scheduling method is needed to determine whether to make forward or backward interpolation in a period. Secondly, a meeting processing method should be proposed to adjust the forward and backward feedrates to equal with zero acceleration at the meeting point to guarantee the motion smoothness. Thirdly, the round-off error caused by cycle sampling needs to be compensated while maintaining the smoothness of feedrate profile. Therefore, a bidirectional adaptive interpolation strategy with three sub-stages is proposed and its flowchart is shown in Fig. 6.

3.1) Sub-stage 1: bidirectional feedrate scheduling and interpolation based on the forward maximum feedrate v_{di}^f and backward maximum feedrate v_{dj}^b.

In this sub-stage, the two directions interpolation are performed synchronously in phase 1, 2, 6 and 7 of S-shaped ACC/DEC profile. But only forward interpolation point $C(u_i)$ will be output to servo system and the backward interpolation results should be stored in the data buffer $\{[u_j, C(u_j)]\}$.

In order to perform the two directions interpolation orderly, an adaptive task scheduling method is proposed where v_{di}^f and v_{dj}^b are selected as the scheduling conditions. In each period, forward interpolation takes a step with the target feedrate v_i^f. The corresponding feed length Δl_i^f and the predicted decreasing acceleration length Δl_{di}^f can also be calculated. Meanwhile, if $v_{di}^f \geq v_{dj}^b$, backward interpolation takes a few steps further with v_j^b, Δl_j^b and Δl_{dj}^b until $v_{di}^f < v_{dj}^b$. Otherwise, backward interpolation is not implemented in this period. Therefore, through this adaptive scheduling principle, it can be guaranteed that the forward interpolation is applied once in every period and the maximum feedrates of two directions always satisfy the inequality after every interpolation period as follows:

$$v_{di}^f < v_{dj}^b \tag{31}$$

which is an important condition for the meeting processing. It can be seen that the optimized feedrate look-ahead algorithm given in subsection III.A.4 is very useful to ensure the real-time performance especially when the start feedrate is much bigger than the end feedrate of NURBS segment.

In order to prevent overshoot of two directions interpolation and generate smooth feedrate profile at the meeting point, a novel meeting processing method is proposed which consists of two modules namely meeting detection and feedrate adjustment. After every forward or backward feedrate scheduling, the meeting detection module should be applied with the obtained parameters. If

$$\Delta l_i^f + \Delta l_{di}^f + \Delta l_j^b + \Delta l_{dj}^b + \Delta l_{con} < l_{ij}^r \tag{32}$$

where l_{ij}^r is the remaining length of the current segment and $\Delta l_{con} = \max(v_{di}^f, v_{dj}^b)T_s$ which is of great significance to round-off error compensation and will be introduced in Substage 2, it means that the two directions interpolation cannot meet each other under the current scheduled feedrate. Therefore, the corresponding target point position $C(u_i)$ or $C(u_j)$ can be calculated. Otherwise, the meeting will happen and the backward interpolation should be stopped. In particular, if $\Delta l_i^f + \Delta l_{di}^f + \Delta l_j^b + \Delta l_{dj}^b + \Delta l_{con} = l_{ij}^r$ and $v_{di}^f = v_{dj}^b$, the feedrate scheduling results can be accepted exactly. But if $\Delta l_i^f + \Delta l_{di}^f + \Delta l_j^b + \Delta l_{dj}^b + \Delta l_{con} > l_{ij}^r$ or $v_{di}^f \neq v_{dj}^b$, the feedrate adjustment module should be performed.

Considering the feedrate scheduling direction which is being processed and the relation between v_{di}^f and v_{dj}^b, there are various situations of feedrate adjustment. As shown in Fig. 7, the situation that $v_{di}^f > v_{dj}^b$ and forward feedrate is under scheduling is discussed and the other situations have similar analysis steps and adjustment results. The feedrate adjustment module consists of three sub-steps and can be described as follows:

- Step 1: adjust v_{di}^f to v'^f_{di} where $v'^f_{di} = v_{dj}^b$

The adjusted target feedrate can be expressed as follows:

$$v'^f_i = v_{i-1}^f + \frac{a_{i-1}^f + a'^f_i}{2} T_s \tag{33}$$

where a'^f_i is the adjusted target acceleration. The maximum feedrate v'^f_{di} can be derived correspondingly as follows:

$$v'^f_{di} = v_{dj}^b = v'^f_i + \frac{(a'^f_i)^2}{2J_{\max}} \tag{34}$$

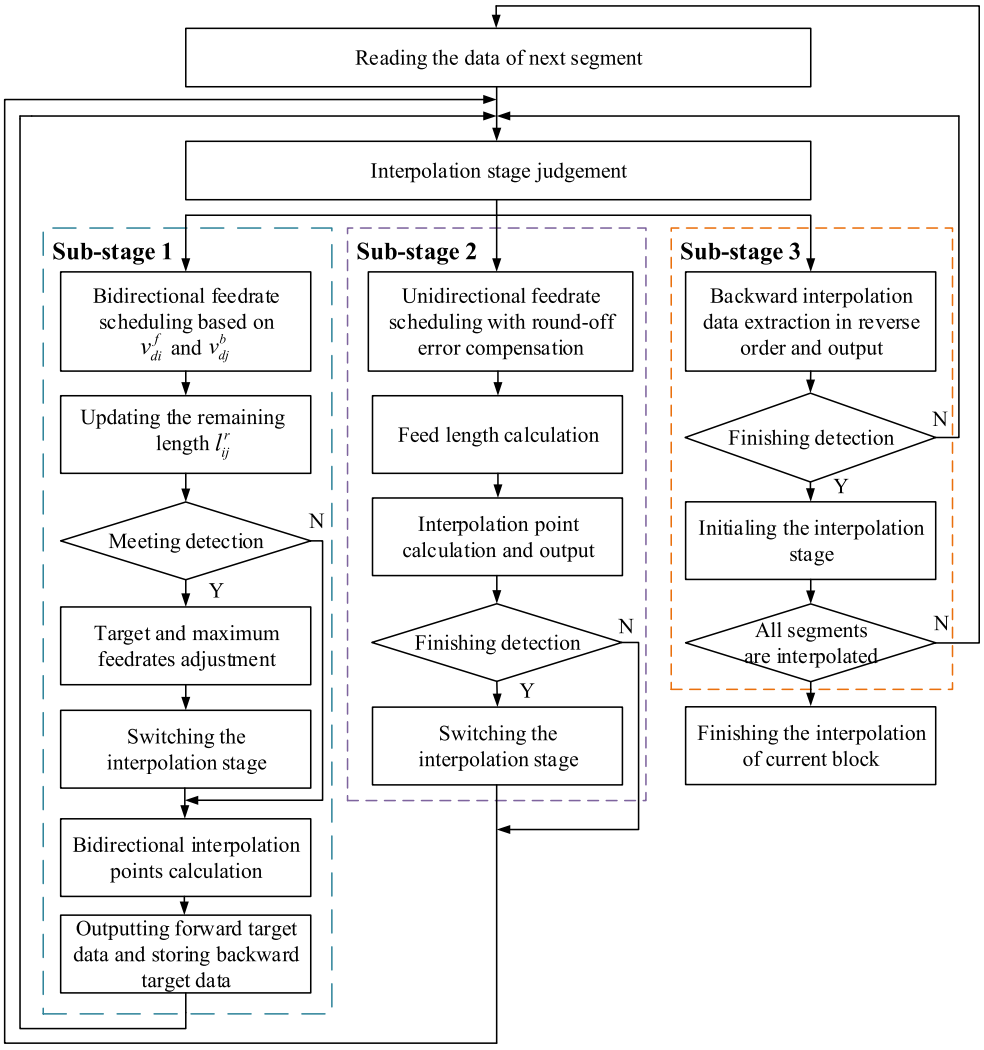


FIGURE 6. Flowchart of the bidirectional adaptive interpolation strategy.

Therefore, the constructed equation can be obtained based on Eq. (33)-(34) as follows:

$$\frac{1}{2J_{\max}}(a'^{f}_{i})^2+\frac{T_s}{2}a'^{f}_{i}+v^{f}_{i-1}+\frac{T_s}{2}a^{f}_{i-1}-v^{b}_{dj}=0 \quad (35)$$

Eq. (35) is a quadratic equation. Therefore, a'^{f}_{i} can be solved directly as follows:

$$a'^{f}_{i}=-\frac{J_{\max}T_s}{2}+J_{\max}\sqrt{\frac{T_s^2}{4}-\frac{2}{J_{\max}}(v^{f}_{i-1}+\frac{T_s}{2}a^{f}_{i-1}-v^{b}_{dj})} \quad (36)$$

Correspondingly, the adjusted target length $\Delta l'^{f}_{i}$ and the predicted decreasing acceleration length $\Delta l'^{f}_{di}$ can be obtained based on Eq. (25)-(28). Then, the comparison should be conducted as follows:

if $\Delta l'^{f}_{i}+\Delta l'^{f}_{di}+\Delta l^{b}_{j}+\Delta l^{b}_{dj}+\Delta l'_{con}\le l^{r}_{ij}$, go to Step 3;
if $\Delta l'^{f}_{i}+\Delta l'^{f}_{di}+\Delta l^{b}_{j}+\Delta l^{b}_{dj}+\Delta l'_{con}> l^{r}_{ij}$, go to Step 2;
where $\Delta l'_{con}=v^{b}_{dj}T_s$.

- Step 2: readjust v'^{f}_{di} and v^{b}_{dj}

In this situation, a'^{f}_{i} and a^{b}_{j} should be further reduced to maintain $v''^{f}_{di}=v''^{b}_{dj}$ where v''^{f}_{di} and v''^{b}_{dj} are the readjusted maximum feedrates. According to Eq. (33)-(34), v''^{f}_{di} and v''^{b}_{dj} can be expressed as follows:

$$v''^{f}_{di}=v^{f}_{i-1}+\frac{a^{f}_{i-1}+a''^{f}_{i}}{2}T_s+\frac{(a''^{f}_{i})^2}{2J_{\max}} \quad (37)$$

$$v''^{b}_{dj}=v^{b}_{j-1}+\frac{a^{b}_{j-1}+a''^{b}_{j}}{2}T_s+\frac{(a''^{b}_{j})^2}{2J_{\max}} \quad (38)$$

where a''^{f}_{i} and a''^{b}_{j} are the adjusted target acceleration of two directions, respectively. The corresponding lengths can be derived as follows based on Eq. (25)-(28):

$$\Delta l''^{f}_{i}+\Delta l''^{f}_{di}=v^{f}_{i-1}T_s+\frac{a^{f}_{i-1}}{3}T_s^2+\frac{a''^{f}_{i}}{6}T_s^2+v''^{f}_{i}t''^{f}_{di}+\frac{a''^{f}_{i}}{2}(t''^{f}_{di})^2-\frac{J_{\max}}{6}(t''^{f}_{di})^3 \quad (39)$$

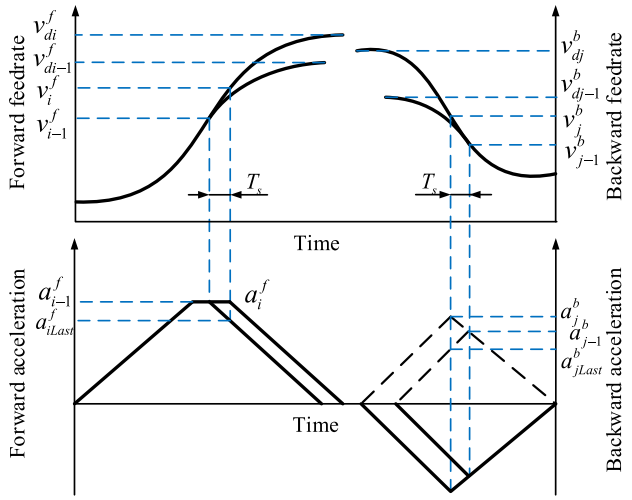


FIGURE 7. Distribution of forward and backward feedrate and acceleration profiles.

$$\Delta l''^{b}_{j} + \Delta l''^{b}_{dj} = v^{b}_{j-1}T_s + \frac{a^{b}_{j-1}}{3}T_s^2 + \frac{a''^{b}_{j}}{6}T_s^2 + v''^{b}_{j}t''^{b}_{dj} + \frac{a''^{b}_{j}}{2}(t''^{b}_{dj})^2 - \frac{J_{\max}}{6}(t''^{b}_{dj})^3 \quad (40)$$

where $v''^{f}_{i} = v^{f}_{i-1} + \frac{a^{f}_{i-1}+a''^{f}_{i}}{2}T_s$, $v''^{b}_{j} = v^{b}_{j-1} + \frac{a^{b}_{j-1}+a''^{b}_{j}}{2}T_s$, $t''^{f}_{di} = \frac{a''^{f}_{i}}{J_{\max}}$ and $t''^{b}_{dj} = \frac{a''^{b}_{j}}{J_{\max}}$.

Finally, the various lengths should satisfy the equation as follows:

$$\Delta l''^{f}_{i} + \Delta l''^{f}_{di} + \Delta l''^{b}_{j} + \Delta l''^{b}_{dj} + \Delta l''_{con} = l^{r}_{ij} \quad (41)$$

where $\Delta l''_{con} = v''^{f}_{di}T_s$. According to Eq. (37)-(41), the adjusted target acceleration of forward interpolation a''^{f}_{i} can be selected as the independent variable and Eq. (41) can be rewritten as follows:

$$f(a''^{f}_{i}) = 0 \quad (42)$$

Since Eq. (42) is a high order equation of a''^{f}_{i}, Newton-Raphson method is employed to solve a''^{f}_{i} with the following iterative format:

$$\begin{cases} do: a_m = a_{m-1} - \dfrac{f(a_{m-1})}{f'(a_{m-1})}, & m = 1, 2, \ldots \\ until: \left|\dfrac{a_m - a_{m-1}}{a_m}\right| \le \delta_{new}, & then: a''^{f}_{i} = a_m \end{cases} \quad (43)$$

where δ_{new} denotes the maximum relative root error, a_0 is the initial value.

From Eq. (27), it can be derived that v^{f}_{di} is monotonically increasing with a^{f}_{i}. Hence, based on the adaptive task scheduling method and Eq. (31), the maximum feedrate v^{f}_{di-1} calculated last period is smaller than v^{b}_{dj} as shown in Fig. 7. Therefore, there must exist a unique solution a''^{f}_{i} in $(a^{f}_{iLast}, a^{f}_{i})$ where $a^{f}_{iLast} = a^{f}_{i-1} - J_{\max}T_s$ and the solvability of the proposed meeting processing method is proved. Furthermore, based on the calculation results of Step 1, it can be derived that $a''^{f}_{i} \in (a^{f}_{iLast}, a'^{f}_{i})$. Hence, a_0 in formula (43) can be selected as $\frac{a'^{f}_{i}-a^{f}_{iLast}}{2}$. After a''^{f}_{i} is solved, it should go to Step 3.

- Step 3: output the forward interpolation results and store the backward interpolation results

In this step, the adjusted forward and backward interpolation results should be output and stored, respectively. Meanwhile, the Sub-stage 1 of the bidirectional interpolation is completed. In next period, the interpolation should enter Sub-stage 2 which consists of the phase 3, 4 and 5 of S-shaped ACC/DEC profile.

There is another situation that the interpolation should switch to Sub-stage 2. For long segment, the constant feedrate phase is existing. Hence the Eq. (32) is always satisfied. But both v^{f}_{di} and v^{b}_{dj} should not be larger than the command feedrate F. Therefore, when v^{f}_{di} and v^{b}_{dj} are both equal to F, the interpolation should enter Sub-stage 2.

3.2) Sub-stage 2: unidirectional feedrate scheduling and interpolation with round-off error compensation

In this sub-stage, the feedrate scheduling and interpolation of phase 3, 4, and 5 should be performed with the remaining length l^{r}_{ij}. After Sub-stage 1, the each phase time can be obtained and assumed as t_{p3}, t_{p4} and t_{p5}, respectively. However, it is hard to guarantee that both t_{p3}, t_{p4} and t_{p5} are integer multiple of the interpolation period T_s. Therefore, the existence of round-off error is inevitable.

To reduce the round-off error, the time rounding principle given in [25] is employed in this paper and tweaked according to the particularity of the bidirectional interpolation. The total interpolation period number in Sub-stage 2 can be calculated as follows:

$$N_{345} = [\frac{t_{p3} + t_{p4} + t_{p5}}{T_s}] \quad (44)$$

where the operator '[]' denotes rounding down. Therefore, the round-off error Δl_{err} is only the length corresponding to Δt which can be expressed as follows:

$$\Delta t = (t_{p3} + t_{p4} + t_{p5}) - N_{345}T_s < T_s \quad (45)$$

In order to avoid the discontinuity of acceleration profile, the compensated interpolation time Δt and length Δl_{err} should be taken from the constant feedrate phase. Hence, it should be ensured that the constant feedrate phase is existing and t_{p4} is bigger than Δt. In Sub-stage 1, Δl_{con} is introduced to meeting detection module and feedrate adjustment module, which can guarantee that the constant feedrate phase has at least one interpolation period whether the segment is long or short. Correspondingly, Δl_{err} can be calculated as follows:

$$\Delta l_{err} = v^{f}_{di}\Delta t \quad (46)$$

After Δl_{err} is obtained, the composite ACC/DEC algorithm [25], which consists of S-shaped ACC/DEC algorithm and the improved trapezoidal ACC/DEC algorithm,

is employed to complete the round-off error compensation. Therefore, the feed length Δl_i of each interpolation period can be calculated as follows:

$$\Delta l_i = \Delta l_{Si} + \Delta l_{Ti} \tag{47}$$

where Δl_{Si} is calculated by the S-shaped algorithm and Δl_{Ti} is the compensated length obtained by the improved trapezoidal algorithm.

When the interpolation of Sub-stage 2 is finished, Sub-stage 3 should begin in next period.

3.3) Sub-stage 3: extracting the backward interpolation data in reverse order

In this sub-stage, the feed length and the corresponding target point position do not need to be calculated because the remaining part has already interpolated by backward interpolation performed in Sub-stage 1. Therefore, the stored backward interpolation data $\{[u_j, C(u_j)]\}$ can be extracted in reverse order and sent to servo system in each period until all the stored data is taken out.

4) TARGET INTERPOLATION POINT CALCULATION

The target interpolation point calculation includes two sub-steps. The first step is to determine the parameter u_i of the target interpolation point based on the feed length Δl_i. Then in second step, the target interpolation point position $C(u_i)$ can be calculated by substituting u_i into the path description $C(u)$. The second step can be realized by de Boor-Cox algorithm [3] with high efficiency. A large number of previous researches on NURBS interpolation have focused on finding an efficient algorithm for the calculation of u_i including the first- and second-order approximation of Taylor series expansion [35], [36], the predictor-corrector interpolation (PCI) method [37] and re-mapping techniques [23], [33], [38]–[40]. The PCI method is easy to understand and implement with arbitrarily set accuracy and fast convergence rate. Therefore, it is employed in this paper. In the predictor stage, the initial value u_0 can be obtained by second-order expansion and expressed as follows:

$$\begin{aligned} u_0 &= u_{i-1} + T_s \frac{du}{dt}\Big|_{u=u_{i-1}} + \frac{T_s^2}{2}\frac{du^2}{d^2t}\Big|_{u=u_{i-1}} \\ &= u_{i-1} + \frac{\Delta l_i}{\sqrt{x'^2 + y'^2 + z'^2}} - \frac{\Delta l_i^2 (x'x'' + y'y'' + z'z'')}{2(x'^2 + y'^2 + z'^2)^2} \end{aligned} \tag{48}$$

Then in corrector stage, Newton-Raphson method shown in formula (43) is utilized with the specified tolerance of feedrate fluctuation δ_{flu} and the equation can be constructed as follows:

$$F(u_i) = \|C(u_i) - C(u_{i-1})\| - \Delta l_i = 0 \tag{49}$$

IV. SIMULATION AND EXPERIMENTAL RESULTS

In this section, simulations of two NURBS curves are conducted to evaluate the good performance of the proposed bidirectional adaptive feedrate scheduling (BAFS) method compared with feedrate scheduling (NIFS) method [23] and the bidirectional optimal interpolation (BOI) method [32] in IV.A. In addition, the experimental results of the test NURBS curves are shown in IV.B to evaluate the accuracy, applicability and real-time performance of BAFS method.

A. SIMULATION RESULTS AND ANALYSIS

1) TEST CASES AND SIMULATION ENVIRONMENT

The butterfly-shaped curve and the ∞-shaped curve [21] are selected as the case studies, which are wildly used as the test cases. Meanwhile, the simulations are conducted on a personal computer with similar configuration as given in [21].

Through the arc length and curvature calculation module given in III.A.2 with the parameter sampling length $\Delta u = 0.0001$, the test curves with the critical points and their curvature curves are shown in Fig. 8(a)-(d). The interpolation parameters used for simulations are listed in Table I. It can be seen that the curvature curve of butterfly-shaped curve changes more violently and has bigger maximum curvature than ∞-shaped curve. Meanwhile, the motion parameters for the butterfly-shaped curve are smaller than another group. Through these two kinds of curves with different features and motion parameters, the performance of the proposed BAFS method can be evaluated.

2) ANALYSIS AND COMPARISONS OF BUTTERFLY-SHAPED CURVE

The simulation results of butterfly-shaped curve obtained by the proposed BAFS method are shown in Fig. 9(a)-(e). As can be seen, the tangential acceleration profile is continuous and the feedrate profile is smooth. Meanwhile, the centripetal acceleration and chord error of each interpolation point can satisfy the specified tolerances. However, because of the round-off error compensation, the jerk of some interpolation points exceeds the tolerance but still keeps in a certain extent.

The simulation results obtained by NIFS method and BOI method are shown in Fig. 10(a)-(e) and Fig 11(a)-(e), respectively. However, the round-off error compensation methods for NIFS and BOI are not given. Therefore, the total machining times in Fig. 10 and Fig. 11 are not integer multiple of T_s and the jerk profile is limited in the given range. As can be seen, the tangential acceleration profile in Fig. 11(b) is discontinuous with abrupt changes in some interpolation points because of the unreasonable meeting feedrate adjustment method. In contrast, the tangential acceleration profile of NIFS method is continuous owing to the standard S-shaped ACC/DEC profile. In addition, the chord error and the centripetal acceleration of NIFS and BOI methods are within the allowable tolerance. The total machining times and maximum chord errors of NIFS, BOI and BAFS methods are shown in Table II. The machining time of BAFS is slightly longer than NIFS and BOI because the proposed round-off error compensation method adds a constant feedrate period for some short segments. Furthermore, κ_i is replaced by κ_{i-1}

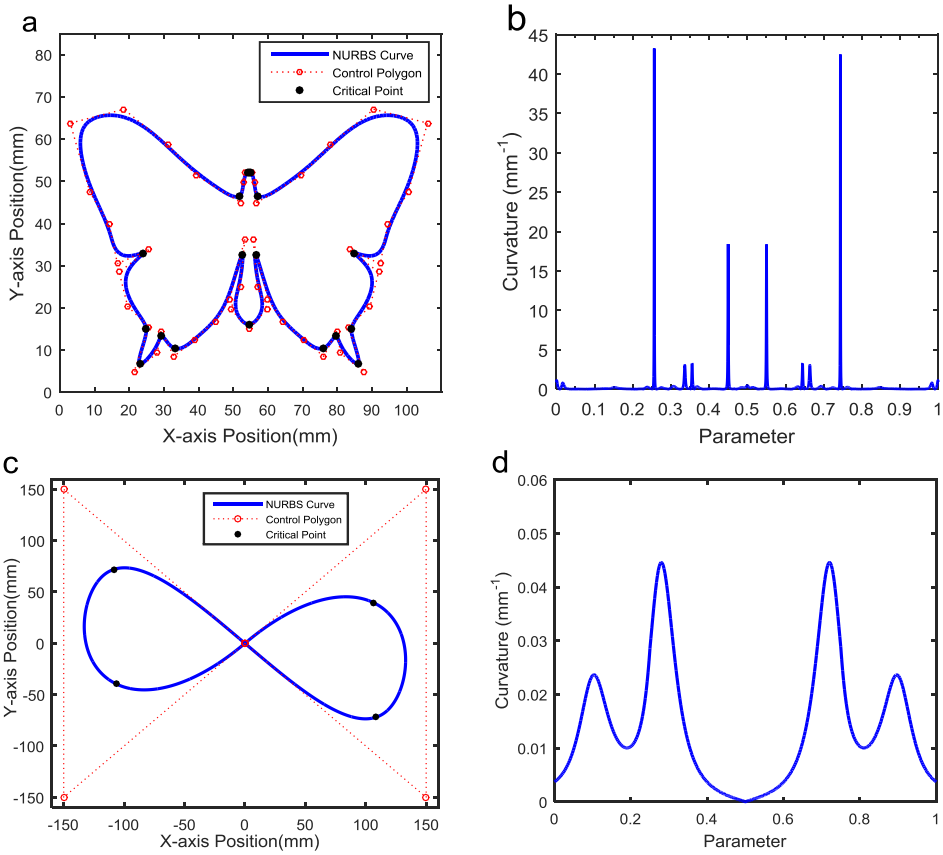


FIGURE 8. **Test curves and their curvature curves. (a) Butterfly-shaped curve. (b) Curvature of butterfly-shaped curve. (c) ∞-shaped curve. (d) Curvature of ∞-shaped curve.**

TABLE 1. **Interpolator parameters.**

Parameters	Butterfly-shaped curve	∞ -shaped curve
Interpolation period T_s	1 ms	1 ms
Arc length tolerance δ_{arc}	10^{-6} mm	10^{-6} mm
Maximum chord error δ_{cho}	0.5 μm	0.5 μm
Feedrate fluctuation tolerance δ_{flu}	10^{-8}	10^{-8}
Dichotomy method tolerance δ_{dic}	10^{-4} mm/s	10^{-4} mm/s
Newton-Raphson method tolerance δ_{new}	10^{-4} mm/s	10^{-4} mm/s
Command feedrate F	100 mm/s	650 mm/s
Maximum centripetal acceleration $a_{\max c}$	3000 mm/s^2	4200 mm/s^2
Maximum tangential acceleration $a_{\max t}$	3000 mm/s^2	4200 mm/s^2
Maximum jerk $J_{\max}$	60000 mm/s^3	70000 mm/s^3

to reduce the computational load in Eq. (26), which also increases the machining time slightly.

It can be seen from the simulation results of butterfly-shaped curve that NIFS method and the proposed BAFS method can generate similar results for the machining situations with smaller $J_{\max}$, $a_{\max t}$, F and NURBS curves whose curvature curve changes sharply especially in the areas near the critical points.

3) ANALYSIS AND COMPARISONS OF ∞-SHAPED CURVE

The simulation results for ∞-shaped curve by BAFS, NIFS and BOI methods are shown in Fig. 12(a)-(e), Fig. 13(a)-(e) and Fig. 14(a)-(e), respectively. As can be seen, the tangential acceleration profile shown in Fig. 14(b) obtained by BOI method still has abrupt changes for the same reason explained in IV.A.2. But the chord error and centripetal acceleration are always within the tolerances. For the results of NIFS method,

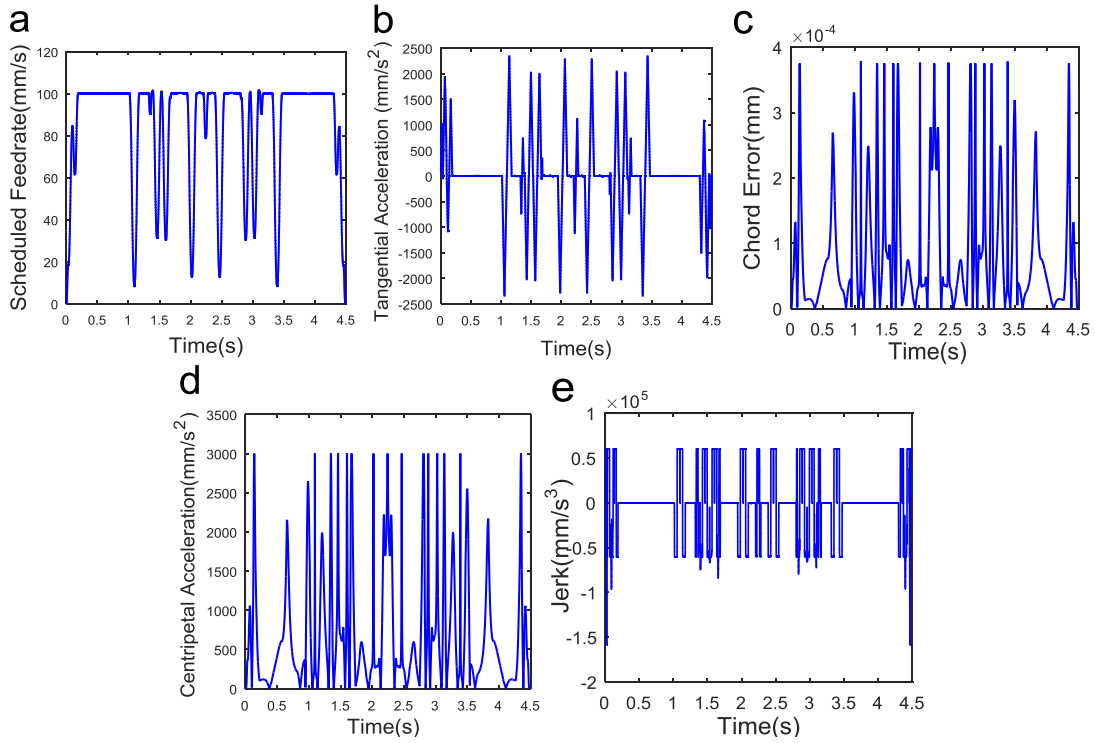


FIGURE 9. Simulation results of butterfly-shaped curve by BAFS method. (a) Scheduled feedrate. (b) Tangential acceleration. (c) Chord error. (d) Centripetal acceleration. (e) Jerk.

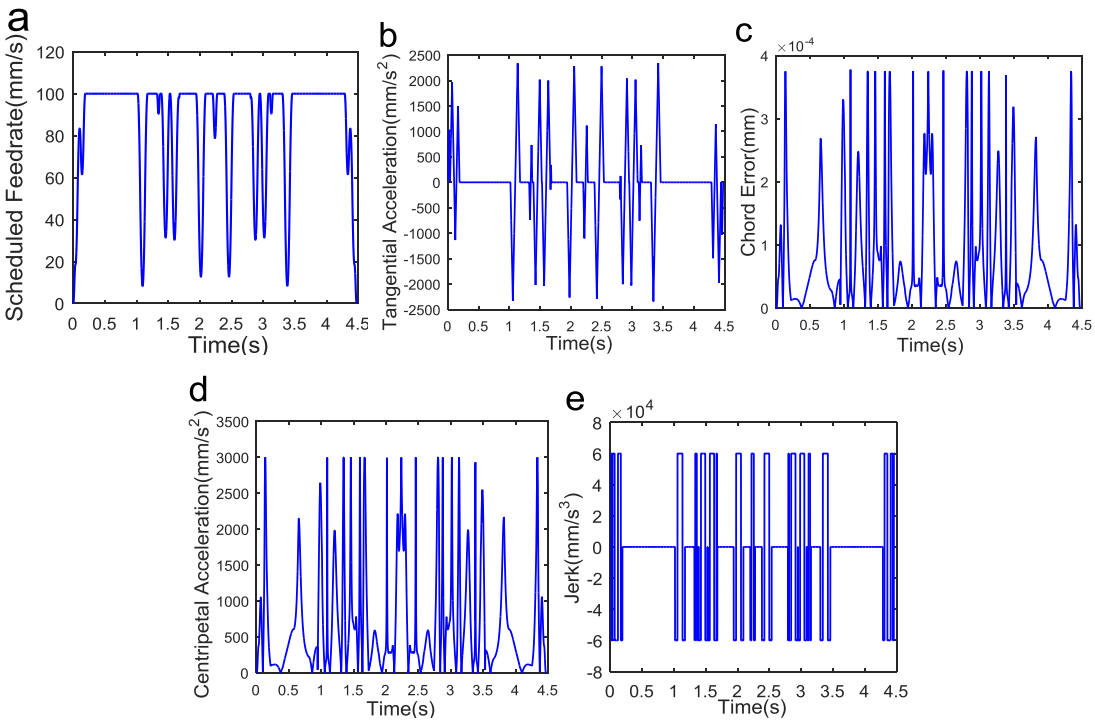


FIGURE 10. Simulation results of butterfly-shaped curve by NIFS method without round-off error compensation. (a) Scheduled feedrate. (b) Tangential acceleration. (c) Chord error. (d) Centripetal acceleration. (e) Jerk.

the chord error and centripetal acceleration in some interpolation points exceed the confined tolerances. Correspondingly, the feedrate profile shown in Fig. 13(a) cannot be within the restricted feedrate range in some areas because the constraints in the middle part of each segment are ignored. In contrast, the chord error and centripetal acceleration profiles shown

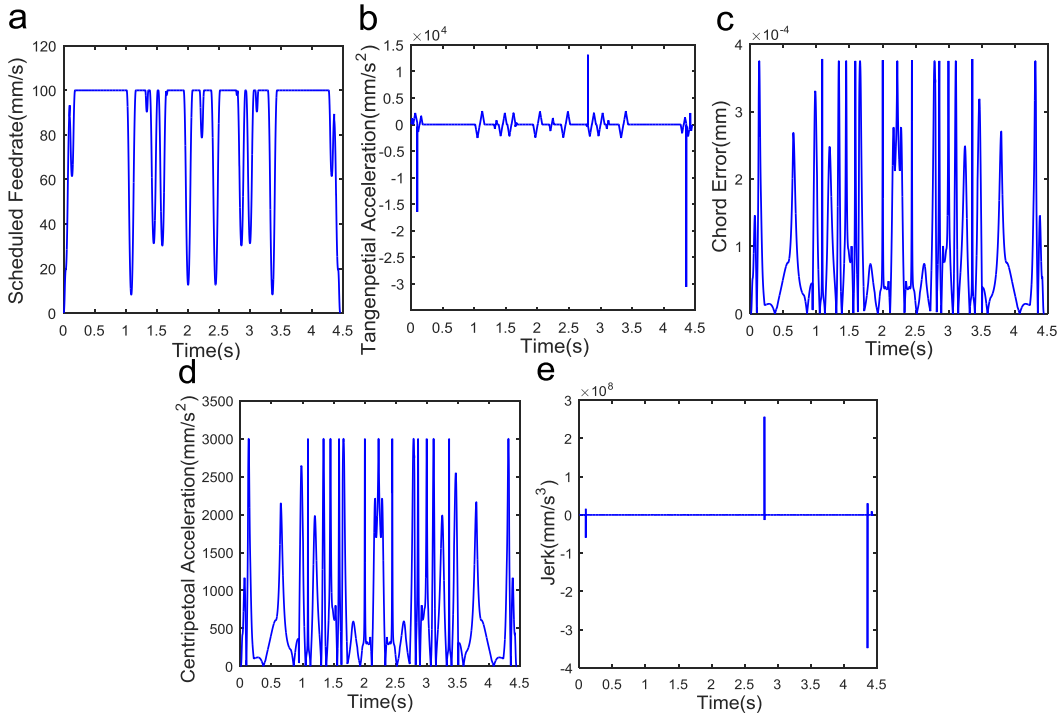


FIGURE 11. Simulation results of butterfly-shaped curve by BOI method without round-off error compensation. (a) Scheduled feedrate. (b) Tangential acceleration. (c) Chord error. (d) Centripetal acceleration. (e) Jerk.

TABLE 2. Static comparisons of maximum chord error and machining time of butterfly-shaped curve.

Test methods	Maximum chord error	Machining time
BAFS method	0.378 μm	4488 ms
NIFS method	0.378 μm	4475.1 ms
BOI method	0.378 μm	4442.8 ms

TABLE 3. Static comparisons of maximum chord error and machining time of ∞-shaped curve.

Test methods	Maximum chord error	Machining time
BAFS method	0.500 μm	1684 ms
NIFS method	0.637 μm	1664.9 ms
BOI method	0.500 μm	1682.7 ms

in Fig. 12(c)-(d) obtained by the proposed BAFS method are always within the confined tolerances. The jerk profile shown in Fig. 12(e) is oscillatory in some areas, which is different from the standard S-shaped ACC/DEC profile, because these parts on the ∞-shaped curve have stronger continuous constraints on feedrate than ACC/DEC algorithm and the jerk needs to be adjusted adaptively with some oscillations. Therefore, the jerk cannot be guaranteed to be a fixed value but the tangential acceleration profile shown in Fig. 12(b) is still continuous. Although the jerk profile has some oscillations and exceeds the maximum jerk limit caused by round-off error compensation, it still keeps in a certain extent and has little effect to interpolation accuracy and motion smoothness.

The machining times and maximum chord errors of NIFS, BOI and BAFS methods for ∞-shaped curve are shown in Table III. The BAFS method has smaller chord error and longer machining time because the adaptive scheduled feedrate under the continuous constraints is smaller than that obtained by standard S-shaped ACC/DEC algorithm. In addition, the reasons given in subsection IV.A.2 also extend the machining time.

It can be seen from the simulation results of ∞-shaped curve that the proposed BAFS method can improve the interpolation accuracy while maintaining the motion smoothness for the machining situations with larger $J_{\max}$, $a_{\max t}$, F and NURBS curves whose curvature curves change slowly especially in the areas near the critical points.

B. EXPERIMENT RESULTS

In this paper, the experiments are conducted on a two-axis motion platform with Panasonic MBDH series servo derives and MHMD series motors. The layout of the experimental system is shown in Fig. 15. The experiment is implemented in a PC-based motion controller which is self-developed by our team [21]. The Kithara real-time suite (KRTS) [41] is installed to the controller to guarantee the real-time performance. Based on KRTS, the OS of motion controller can be divided into two sub-systems: the non-real-time operating system (non-RTOS) which can conduct the tasks with

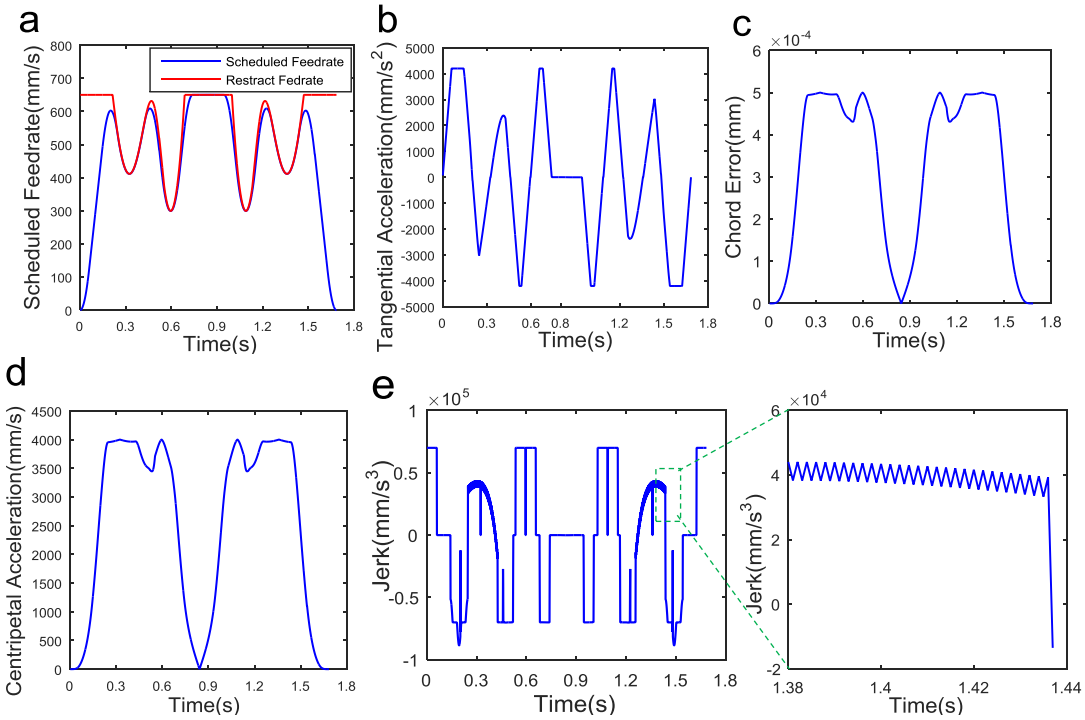


FIGURE 12. Simulation results of ∞-shaped curve by BAFS method. (a) Scheduled feedrate. (b) Tangential acceleration. (c) Chord error. (d) Centripetal acceleration. (e) Jerk.

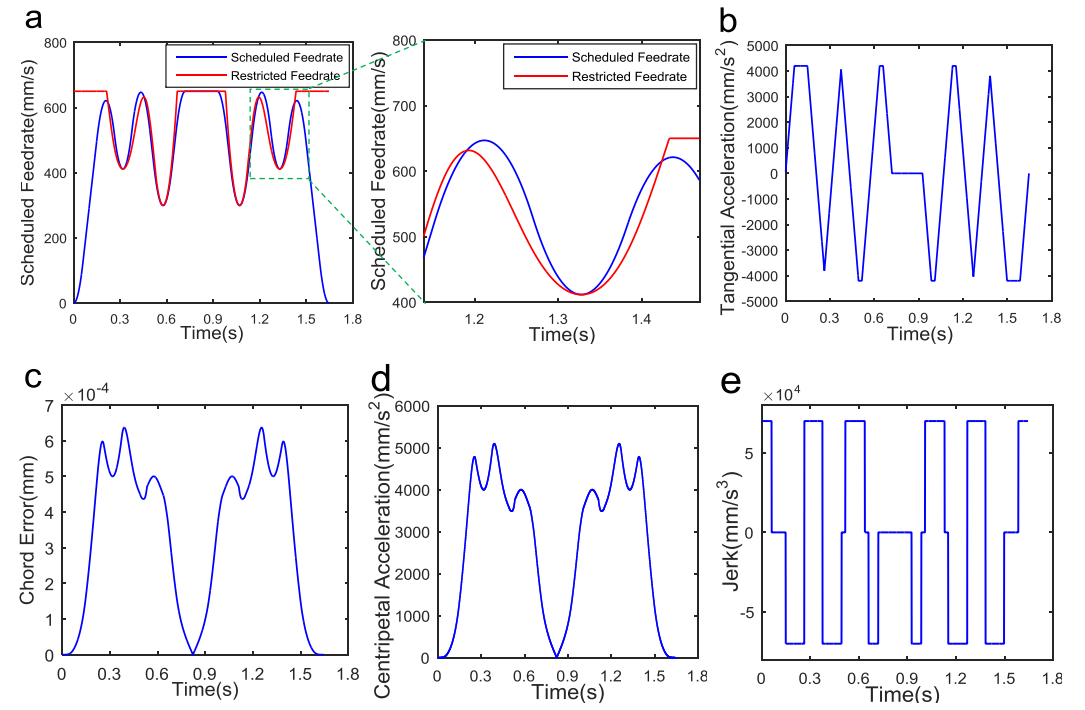


FIGURE 13. Simulation results of ∞-shaped curve by NIFS method without round-off error compensation. (a) Scheduled feedrate. (b) Tangential acceleration. (c) Chord error. (d) Centripetal acceleration. (e) Jerk.

no real-time requirements and the KRTS-Kernel which is a real-time system with excellent real-time performance. Therefore, the pre-processing stage illustrated in Fig. 1 can be implemented in non-RTOS while the real-time interpolation stage is conducted in KRTS-Kernel where the interpolation period is set to 1 ms. In addition, the axis control data can be sent to the corresponding servo derives by EtherCAT.

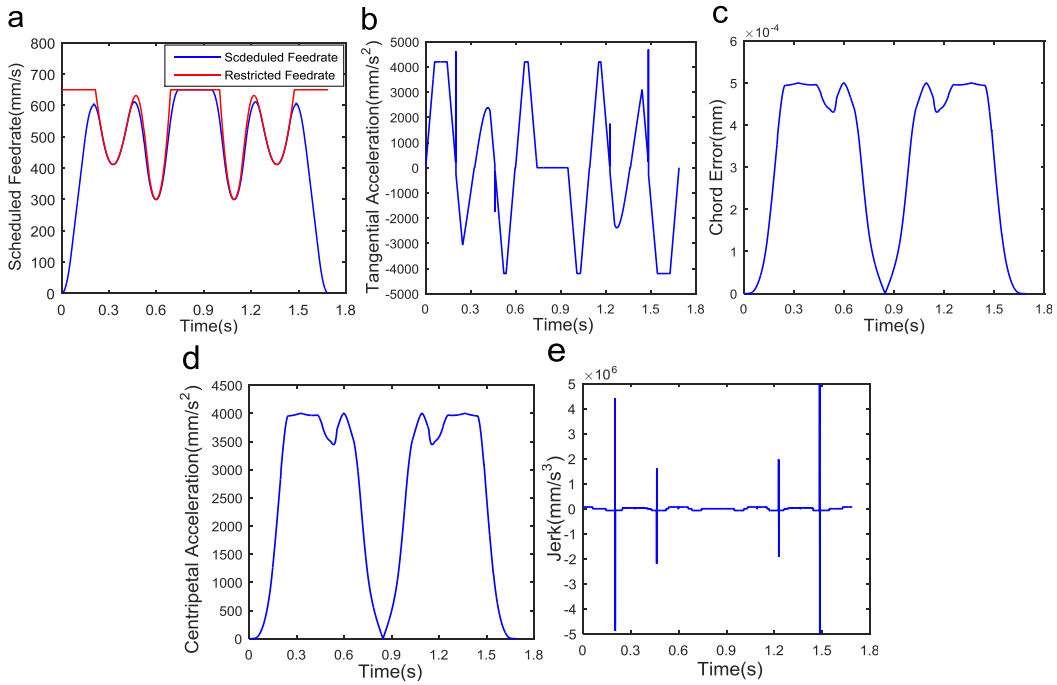


FIGURE 14. Simulation results of ∞-shaped curve by BOI method without round-off error compensation. (a) Scheduled feedrate. (b) Tangential acceleration. (c) Chord error. (d) Centripetal acceleration. (e) Jerk.

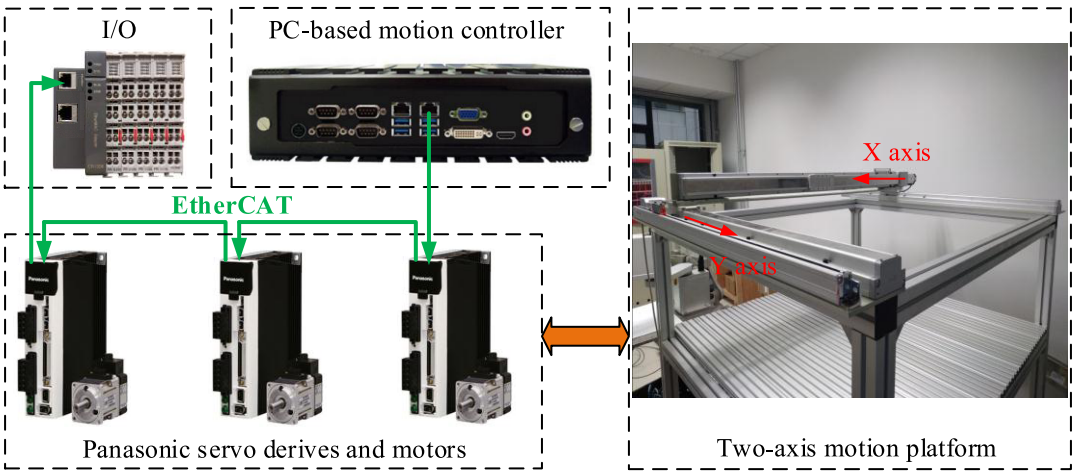


FIGURE 15. Layout of the experimental system.

TABLE 4. Static comparisons of the experiment results.

Test curves	Interpolation method	Maximum contour error (mm)	RMS of contour error (mm)
Butterfly-shaped curve	BAFS method	0.0256	0.00902
	NIFS method	0.0260	0.00903
∞ -shaped curve	BAFS method	0.0432	0.0280
	NIFS method	0.0535	0.0301

Because the acceleration profiles obtained by BOI method are discontinuous with huge impact to the experiment structure, only the experiments and comparative analysis of the BAFS method and the NIFS method are performed. The experiment results of the two test curves are shown in Fig. 16 and Fig. 17, respectively. And the statistical data

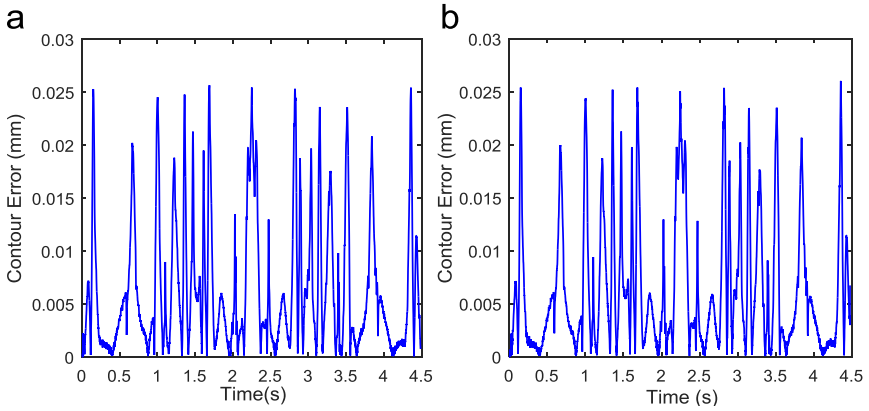


FIGURE 16. Contour error of the butterfly-shaped curve obtained by different method. (a) BAFS method. (b) NIFS method.

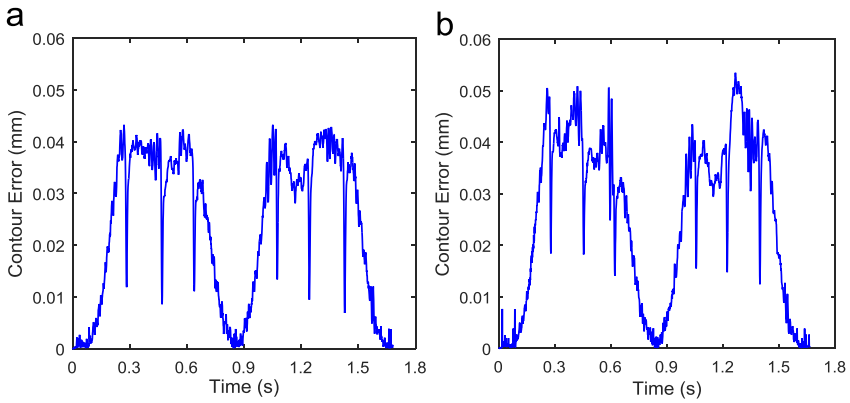


FIGURE 17. Contour error of the ∞-shaped curve obtained by different method. (a) BAFS method. (b) NIFS method.

TABLE 5. Statistical data of the computational time by BAFS method.

Test curves	Number of interpolation points	Maximum computational time	Average computational time
Butterfly-shaped curve	4488	38.809 μs	2.301 μs
∞ -shaped curve	1684	44.582 μs	2.671 μs

is summarized in Table IV. There are a lot of calculation methods of contour errors [42]–[44], the method in [44] is employed in this paper. As can be seen, the experiment results of butterfly-shaped curve obtained by NIFS method with 0.0260 mm maximum contour error and 0.00903 mm root-mean-square (RMS) of contour error are similar to the results obtained by the proposed BAFS method with 0.0256mm and 0.00902 mm respectively, which is consistent with the simulation results. But for the ∞-shaped curve, the maximum contour error obtained by BAFS method is 0.0432 mm which is 19.25% smaller than 0.0535 mm obtained by NIFS method. The corresponding RMS of contour error is 0.0280 mm which is smaller than 0.0301 mm obtained by NIFS method. Therefore, the proposed BAFS method has higher accuracy especially for the NURBS curves with slowly changing curvature curves and larger motion parameters.

The real-time performance of the proposed BAFS method is also tested. The computational time in each interpolation period is measured in real-time and illustrated in Table V. As can be seen, both the maximum and the average computational times of butterfly-shaped curve and ∞-shaped curve are smaller than 50 μs. Therefore, the real-time requirement can always be satisfied with 1 ms interpolation period.

V. CONCLUSION

This paper proposes a bidirectional adaptive feedrate scheduling method of NURBS interpolation based on S-shaped ACC/DEC algorithm. The main conclusions of the study are as follows:

- The continuous constraints of feedrate in the middle part of each segment can be considered by

the proposed bidirectional adaptive feedrate scheduling method. Therefore, the interpolation accuracy is improved especially for the NURBS curves with slowly changing curvature curves and large motion parameters.

- To realize the two directions interpolation orderly, an adaptive task scheduling method is designed. Meanwhile, a novel meeting processing method is proposed to adjust the forward and backward feedrates at the meeting point and generate smooth feedrate profile. Its solvability is also proved. Furthermore, the round-off error is also considered and compensated to maintain the motion smoothness.
- The feedrate look-ahead method is optimized with a back-trace correction module, which considers the continuous constraints of feedrate, to correct the look-ahead results obtained by traditional method.
- A serious of simulations and experiments are performed to validate the performance in accuracy, applicability and real-time of the proposed method.

REFERENCES

[1] J. Yang and A. Yuen, "An analytical local corner smoothing algorithm for five-axis CNC machining," *Int. J. Mach. Tools Manuf.*, vol. 123, pp. 22–35, Dec. 2017.

[2] M. Chen, X.-C. Xi, W.-S. Zhao, H. Chen, and H.-D. Liu, "A universal velocity limit curve generator considering abnormal tool path geometry for CNC machine tools," *J. Manuf. Syst.*, vol. 44, pp. 295–301, Jul. 2017.

[3] L. Piegl and W. Tiller, *The NURBS Book*. Berlin, Germany: Springer, 1997.

[4] S.-S. Yeh and P.-L. Hsu, "The speed-controlled interpolator for machining parametric curves," *Comput.-Aided Des.*, vol. 31, no. 5, pp. 349–357, 1999.

[5] F.-C. Wang and P. K. Wright, "Open architecture controllers for machine tools, part 2: A real time quintic spline interpolator," *J. Manuf. Sci. Eng.*, vol. 120, no. 2, pp. 425–432, 1998.

[6] M.-Y. Cheng, M.-C. Tsai, and J.-C. Kuo, "Real-time NURBS command generators for CNC servo controllers," *Int. J. Mach. Tools Manuf.*, vol. 42, no. 7, pp. 801–813, 2002.

[7] 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, 2002.

[8] M. Tikhon, T. J. Ko, S. H. Lee, and H. S. Kim, "NURBS interpolator for constant material removal rate in open NC machine tools," *Int. J. Mach. Tools Manuf.*, vol. 44, nos. 2–3, pp. 237–245, 2004.

[9] Z. Xu, J. Chen, and Z. Feng, "Performance evaluation of a real-time interpolation algorithm for NURBS curves," *Int. J. Adv. Manuf. Technol.*, vol. 20, no. 4, pp. 270–276, 2002.

[10] M. Annoni, A. Bardine, S. Campanelli, P. Foglia, and C. A. Prete, "A real-time configurable NURBS interpolator with bounded acceleration, jerk and chord error," *Comput.-Aided Des.*, vol. 44, no. 6, pp. 509–521, 2012.

[11] 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, 2010.

[12] J. Dong and J. A. Stori, "Optimal feed-rate scheduling for high-speed contouring," *J. Manuf. Sci. Eng.*, vol. 129, no. 1, pp. 497–513, 2003.

[13] X. Beudaert, S. Lavernhe, and C. Tournier, "Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path," *Int. J. Mach. Tools Manuf.*, vol. 57, no. 3, pp. 73–82, 2012.

[14] Q. Liu, H. Liu, and S. Yuan, "High accurate interpolation of NURBS tool path for CNC machine tools," *Chin. J. Mech. Eng.*, vol. 29, no. 5, pp. 911–920, 2016.

[15] H.-B. Leng, Y.-J. Wu, and X.-H. Pan, "Research on cubic polynomial acceleration and deceleration control model for high speed NC machining," *J. Zhejiang Univ.-Sci. A (Appl. Phys. Eng.)*, vol. 9, no. 3, pp. 358–365, 2008.

[16] F. Y. Luo, Y. F. Zhou, and J. Yin, "A universal velocity profile generation approach for high-speed machining of small line segments with look-ahead," *Int. J. Adv. Manuf. Technol.*, vol. 35, nos. 5–6, pp. 505–518, 2007.

[17] 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, no. 6, pp. 612–628, 2011.

[18] 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, 2016.

[19] W. Fan, X.-S. Gao, W. Yan, and C.-M. Yuan, "Interpolation of parametric CNC machining path under confined jounce," *Int. J. Adv. Manuf. Technol.*, vol. 62, nos. 5–8, pp. 719–739, 2012.

[20] X. Liu, J. Peng, L. Si, and Z. Wang, "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, 2016.

[21] 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*, to be published.

[22] 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, 2007.

[23] 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.

[24] X. Du, J. Huang, and L. M. Zhu, "A complete S-shape feed rate scheduling approach for NURBS interpolator," *J. Comput. Des. Eng.*, vol. 2, no. 4, pp. 206–217, 2015.

[25] H. Ni, T. Hu, C. Zhang, S. Ji, and Q. Chen, "An optimized feedrate scheduling method for CNC machining with round-off error compensation," *Int. J. Adv. Manuf. Technol.*, vol. 97, no. 5, pp. 2369–2381, 2018.

[26] Y. Li, J. Feng, Y. Wang, and J. Yang, "Variable-period feed interpolation algorithm for high-speed five-axis machining," *Int. J. Adv. Manuf. Technol.*, vol. 40, nos. 7–8, pp. 769–775, 2009.

[27] L. Wang, J. F. Cao, and Y. Q. Li, "Speed optimization control method of smooth motion for high-speed CNC machine tools," *Int. J. Adv. Manuf. Technol.*, vol. 49, nos. 1–4, pp. 313–325, 2010.

[28] R. Z. Xu, L. Xie, C. X. Li, and D. S. Du, "Adaptive parametric interpolation scheme with limited acceleration and jerk values for NC machining," *Int. J. Adv. Manuf. Technol.*, vol. 36, nos. 3–4, pp. 343–354, 2008.

[29] 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, 2010.

[30] H. Dong, B. Chen, Y. Chen, J. Xie, and Z. Zhou, "An accurate NURBS curve interpolation algorithm with short spline interpolation capacity," *Int. J. Adv. Manuf. Technol.*, vol. 63, nos. 9–12, pp. 1257–1270, 2012.

[31] 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, 2015.

[32] F. Luo, Y. You, and J. Yin, "Research on the algorithm of NURBS curve bidirectional optimization interpolation with S-type acceleration and deceleration control," *Chin. J. Mech. Eng.*, vol. 48, no. 5, pp. 147–156, 2012.

[33] 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, no. 10, pp. 1530–1541, 2007.

[34] J. H. Mathews and K. D. Fink, *Numerical Methods Using Matlab*. London, U.K.: Pearson, 2003.

[35] J.-T. Huang and D. C. H. Yang, "Precision command generation for computer controlled machines," *ASME Precis. Mach., Technol. Mach. Develop. Improvement*, vol. 58, pp. 89–104, 1992.

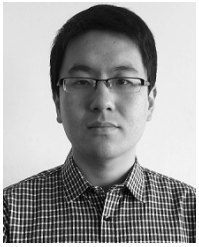
[36] Y. Koren, C. C. Lo, and M. Shpitalni, "CNC interpolators: Algorithms and analysis," ASME Prod. Eng. Div. Publ. Ped., Tech. Rep., 1993, vol. 64, pp. 83–92.

[37] M.-C. Tsai and C.-W. Cheng, "A real-time predictorcorrector interpolator for CNC machining," *J. Manuf. Sci. Eng.*, vol. 125, no. 3, pp. 449–460, 2003.

[38] B. Zhao and W.-G. Li, "Quintic spline interpolation with minimal feed fluctuation," *Mach. Electron.*, vol. 127, no. 2, pp. 523–531, 2007.

[39] J. Yang, W. Ai, Y. Liu, and B. Chen, "Kinematics model and trajectory interpolation algorithm for CNC turning of non-circular profiles," *Precis. Eng.*, vol. 54, pp. 212–221, Oct. 2018.

[40] J. Yang and Y. Altintas, "Generalized kinematics of five-axis serial machines with non-singular tool path generation," *Int. J. Mach. Tools Manuf.*, vol. 75, no. 12, pp. 119–132, 2013.
[41] *Kithara Real-Time Suite*. Accessed: Jun. 5, 2018. [Online]. Available: http://kithara.com/en/products/realtime-suite
[42] J. Yang and Y. Altintas, "A generalized on-line estimation and control of five-axis contouring errors of CNC machine tools," *Int. J. Mach. Tools Manuf.*, vol. 88, pp. 9–23, Jan. 2015.
[43] M. Yang, J. Yang, and H. Ding, "A high accuracy on-line estimation algorithm of five-axis contouring errors based on three-point arc approximation," *Int. J. Mach. Tools Manuf.*, vol. 130, pp. 73–84, Aug. 2018.
[44] J. Yang and Z. Li, "A novel contour error estimation for position loop-based cross-coupled control," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 643–655, Aug. 2011.



HEPENG NI received the B.S. degree in mechanical design, manufacturing and automation from Shandong University, Jinan, China, in 2014, where he is currently pursuing the Ph.D. degree in mechatronics engineering.

His research interests include motion control of CNC machine tools and robot, chatter analysis, and suppression of robot machining.

CHENGRUI ZHANG received the B.S. degree from the Shandong Institute of Mining Technology, Taian, China, in 1982, and the M.S. and Ph.D. degree in mechanical engineering from the Shandong University of Technology, Jinan, China, in 1988 and 1995, respectively, all in mechanical engineering.

He is currently a Professor with the School of Mechanical Engineering, Shandong University, Jinan. His research interests include intelligent manufacture, numerical control technology, robot control, and Ethernet-based distributed control systems.

SHUAI JI received the B.S. degree in mechanical design, manufacturing and automation and the Ph.D. degree in mechatronics engineering from Shandong University, Jinan, China, in 2008 and 2014, respectively.

Since 2014, he has been an Assistant Professor with Shandong Jianzhu University. His research interests include robotics computer numerical controller, real-time Ethernet, and networked control systems.

TIANLIANG HU received the B.S. degree in mechanical design, manufacturing and automation and the Ph.D. degree in mechatronics engineering from Shandong University, Jinan, China, in 2003 and 2008, respectively.

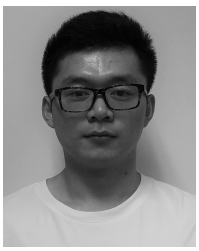
He is currently an Associate Professor with the School of Mechanical Engineering, Shandong University. His research interests include intelligent manufacture, numerical control technology, and robot.

QIZHI CHEN received the B.S. degree in mechanical design, manufacturing and automation from Shandong University, Jinan, China, in 2016, where he is currently pursuing the Ph.D. degree in mechanical manufacturing and automation.

His research interests include motion control of CNC machine tools and robot, chatter analysis, and suppression of robot machining.

YANAN LIU received the B.S. and M.S. degree in mechanical engineering from Shandong University, Jinan, China, in 2013 and 2016. He is currently pursuing the Ph.D. degree in robotics and autonomous system with the Faculty of Engineering, University of Bristol, Bristol, U.K.

His research interests include robotics, robotic vision, robot navigation, and visual surveying.



GONGCHENG WANG received the B.S. degree in mechanical electronic engineering from Shandong Agricultural University, Taian, China, in 2017. He is currently pursuing the M.S. degree in mechanical engineering with Shandong University, Jinan, China.

His research interest is motion control of CNC machine tools and robot.